Brazil Build or modify applications

Last updated: 10/07/2026

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Some examples and graphics depicted herein are provided for illustration only. No real association or connection to ServiceNow products or services is intended or should be inferred.

ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Please read the ServiceNow Website Terms of Use at www.servicenow.com/terms-of-use.html

Company Headquarters
2225 Lawson Lane
Santa Clara, CA 95054
United States
(408) 501-8550

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

servicenow

Table of Contents

Building applications **5**

AI Workflow Factory Prime 9

Discover 11

Build 11

Run 464

AI-led software development life cycle (SDLC) on the ServiceNow AI Platform 598

Learning about developing on the ServiceNow AI Platform 599

Find ServiceNow developer products quickly 599

Programming basics 617

Anatomy of an application 619

Personal developer instance guide 633

Early Availability guide 646

Low-code versus pro-code development 647

Modifying versus building an application 651

Understand the ServiceNow® UI experiences 651

Support for developers 653

Application collaboration 654

Contextual development environment 658

Licensing 696

Creation restrictions across application scopes 697

Planning your application 698

Submit your idea for app development 699

Specify data for your application 701

Delegated Development 702

Team Development 716

Developing your application 757

App Engine products and offerings 760

Creator Studio 760

App Engine Studio 997

ServiceNow CLI 1253

Building pro-code applications 1310

Builder library 1832

Application Runtime Policy (ARP) 2464

Testing and debugging applications 2482

Automated Test Framework (ATF) 2483

Performance Analyzer 2908

Deploying applications 2911

System update sets 2913

Administer your apps 2942

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Maintaining your application.....2972

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Building applications

Learn how to become an application developer using ServiceNow AI Platform tools. Start with what you know and use AI-powered tools, a library of reusable components, and published applications to modernize your legacy processes.

Get started

Learning about creating applications



Learn basic information about application development.

Phase 1: Planning your application



Plan how your application will work.

Phase 2: Agentically developing your application



Use AI tools such as Build Agent to agentically develop your application with natural language prompts.

Phase 3: Developing your application



Add components and content to your application.

Phase 4: Testing and debugging your application



Verify that the application meets your business requirements.

Phase 5: Deploying your application



Deploy your application to your production environment.

Phase 6: Maintaining your application



Review the status of your application and make changes as needed.

What's new

Build apps smarter and deliver them faster with the new ServiceNow Studio. ServiceNow Studio empowers platform developers with a modern, unified environment for building on the ServiceNow AI Platform. ServiceNow Studio features streamlined navigation to applications and

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

metadata, integrated low-code tools, efficient tracking and packaging of development work that accelerates development processes and enhances productivity.

App development phases



Learning about creating applications

Decide whether you want to build a new application or extend an existing application. Check the ServiceNow Store and the ServiceNow Community for existing solutions.

Before building your first application, you may want to learn some basic information about application development. This phase is optional, and you can complete it at any time while you work on other phases.

- How ServiceNow AI Platform is made up of tables and records. Learn how to convert a spreadsheet into record data.
- How to Get a development instance to practice creating applications.
- How to find out Licensing for which application features require a subscription.
- How to contact Support for developers to ask questions about application development.



Phase 1: Planning your application

The application development process starts with planning. Consider how the application will work, who will use it, and how it will improve your users' experience. Your application plan should answer the following questions:

- What are the goals, objectives, and outputs of your application?
- Who uses your application?
- Who has access to parts of the application?
- What tasks do people complete with your application?
- Where does the data come from?
- How do people interact with your application?
- What processes must the application support?
- What UI experience does the application use?
- Is there an existing application available on the ServiceNow Store or the ServiceNow Community that you can use or extend?
- What subscriptions does your application require?



Phase 2: Agentically developing your application

You can use AI tools to help you plan and agentically develop your application. Agentic development, or vibe coding, is an AI-driven approach to application

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

development. Use agentic development and ServiceNow agentic-assisted app building tools, such as Build Agent, to describe your goals in natural language, and have the ServiceNow AI Platform generate full-stack applications, workflows, and integrations. As successful agentic development depends on the quality of your prompts, consider the following guidelines when working with agentic development tools:

- Use clear, focused prompts with as much detail as possible to generate more accurate and relevant output.
- If you're using Build Agent, include as much context as possible in your first prompt to enable more robust development. Include roles, data requirements, and success criteria.
- When switching between two separate applications in Build Agent, specify the new application context.
- Save your successful prompts to repurpose and reuse. For more information, see Example prompts for agentic development.
- Save your session to quickly resume working where you stopped.
- Use the Build Agent chat panel for iterative development.
- Validate compliance using Vault Console to check audit trails and security settings before deployment.
- Develop in a sandbox using Developer Sandboxes for isolation and safety, or use a Personal Development Instance (PDI) or a non-production instance to avoid production deployment.



Phase 3: Developing your application

During the development phase, you add the components and content of your application. Most applications consist of the following:

Data

Information is stored in your application via tables that you configure. For example, employee phone numbers or office locations.

Experience

Experiences are graphical interfaces that your users interact with. For example, you can create a portal where users find information, submit requests, or complete business tasks.

Logic and automation

Automate all the work in your application by adding logic and automation. For example, you can build a flow that sends a notification to the admin when someone makes a request.

Security

Configure roles and access controls to limit who can use your application. For example, you can restrict access to application data to users who have a specific role.

Choose a builder that matches the type of user experience that your application provides.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- See ServiceNow Studio to learn about the unified developer experience on the ServiceNow AI Platform.
- See Build apps using App Engine Studio to learn about low-code development.
- See Creator Studio to learn about no-code development.
- See Build workflows to learn about creating automation with Workflow Studio.
- See Builder library to learn about specialized application resources.



Phase 4: Testing and debugging your application

Verify that the application meets your business requirements. Your testing should cover the following elements:

- Record operations, such as create, read, update, and delete.
- User interface elements, such as views and UI policies.
- Runtime operations, such as business rules and event script actions.



Phase 5: Deploying your application

After successfully testing an application, deploy it to your production environment with your builder tool.

- See ReleaseOps to learn about the improved and automated deployment process with ReleaseOps.
- See App Engine Management Center to learn about the unified app governance and deployment experience on the ServiceNow AI Platform.
- See ServiceNow application repository to learn about the Application Repository.
- See System update sets to learn about classic deployment using update sets.



Phase 6: Maintaining your application

Use your Phase 2 builder tool to update and modify your application. Use your Phase 3 testing tool to verify that your application still functions properly.

Applications and features

- Learning about developing on the ServiceNow AI Platform
- Planning your application
- Agentically developing your application
- Developing your application
- Testing and debugging applications

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

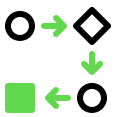
- Deploying applications
- Maintaining your application

AI Workflow Factory Prime

AI Workflow Factory Prime enables you to discover, build, run, and extend on the ServiceNow AI Platform®. From process and task mining, to agentic and sandbox development, use the AI Workflow Factory Prime tools to help your team move from idea to production faster.

Important: Check your entitlements to determine whether you have access to AI Workflow Factory Prime tools and features.

Get started

App Engine Management Center  Govern app development and deployment from a single, unified experience.	App Engine Studio  Build apps in a low-code, integrated builder environment.	ATF Test Generator and Cloud Runner  Generate regression tests for a specific instance with platform intelligence.
Autonomous Engineer  Generate a complete implementation plan from your requirements. Then Build Agent builds all work items in parallel, reducing the need to author each artifact manually.	Creator Studio  Create intelligent workflows in a guided, curated, no-code environment.	Developer Sandboxes  Enable parallel building and testing on top of a shared development instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ServiceNow Otto for App Engine  Add AI capabilities to custom applications.	ServiceNow Otto for Creator  Use AI capabilities such as Build Agent to make development on the ServiceNow AI Platform more efficient.	ServiceNow Studio for App Engine  ServiceNow Studio for App Engine includes premium plugins and capabilities for ServiceNow Studio.
Table Builder for App Engine  Create, design, and administer tables and forms visually from a single user interface.	Task Mining  Collect and analyze workstation activities to understand how tasks are performed, identify inefficiencies, and make data-driven decisions.	Workspace Builder  Create and tailor custom workspaces where agents work on open records.

App Engine Management Center

App Engine Management Center provides a unified experience for governing the development and deployment of custom applications for your organization. Oversee requests for new applications, collaboration requests, and insights into custom app usage and developer data. App Engine Management Center also includes ReleaseOps and Developer Sandboxes.

App Engine Studio

App Engine Studio is a low-code app building tool that enables you to build scoped ServiceNow applications using a guided wizard experience.

ATF Test Generator and Cloud Runner

ATF Test Generator and Cloud Runner generates tests depending on the instance behavior and workflows. You can also execute the generated tests using the Cloud Runner option.

Autonomous Engineer

Autonomous Engineer helps to reduce the time it takes to implement applications by taking your specifications as input and driving the full implementation lifecycle through Build Agent.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Creator Studio

Creator Studio is a tool that enables citizen developers to build request and fulfill applications in a no-code environment.

Developer Sandboxes

Developer Sandboxes provide isolated development environments that enable parallel building and testing on top of a shared development instance. Use sandboxes to reduce code conflicts, accelerate delivery, and safely test configurations without affecting other team members' work.

ServiceNow Otto for App Engine

ServiceNow Otto for App Engine gives developers the tools they need to add AI capabilities to custom applications. You can enhance an existing application with AI, or create an application that features AI from the start.

ServiceNow Otto for Creator

ServiceNow Otto for Creator includes generative and agentic AI features that make developing on the ServiceNow AI Platform® more efficient. The ServiceNow Otto for Creator AI capabilities are embedded across platform tools, so developers can generate apps, flows, code, and UI components without leaving the tools they use every day.

ServiceNow Studio for App Engine

ServiceNow Studio for App Engine includes premium plugins and capabilities for ServiceNow Studio. Install ServiceNow Studio for App Engine to get started building applications in ServiceNow Studio quickly.

Table Builder for App Engine

Table Builder for App Engine enables you to create, design, and administer tables and forms visually from a single user interface.

Task Mining

Task Mining helps process owners collect and analyze workstation activities to understand how tasks are performed, identify inefficiencies, and make data-driven decisions.

Workspace Builder

Workspace Builder enables you to quickly create and tailor custom workspaces where agents work on open records.

Discover

Discover how processes work in your ServiceNow ecosystem using process and task mining, AI-driven process insights, and Automation Center. Uncover bottlenecks, deviations, and inefficiencies that standard analytics don't show.

Build

Build and edit applications with agentic AI development tools available with AI Workflow Factory.

Agentic development on the ServiceNow AI Platform accelerates application development from weeks to minutes, enabling developers of all skill levels to build solutions for their

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

organization's unique workflows. The ServiceNow AI Platform has tools for each stage of application development, from generating flows, to agentically developing entire applications using Build Agent inside ServiceNow Studio or the ServiceNow IDE, to implementing work items autonomously and testing in dedicated sandbox environments.

Build Agent

Build Agent is an autonomous AI agent that generates full-stack ServiceNow® applications from natural language descriptions. You describe the application you need in the chat panel in ServiceNow Studio or the ServiceNow IDE, and Build Agent generates the necessary code, organizes files, and manages both the core logic and user interface components. You can also ask it general questions about developing on the ServiceNow AI Platform.

Build Agent can enhance existing custom applications with agentic capabilities. It evaluates existing app context, such as tables, roles, business rules, and metadata, and generates in-app agents, skills, and agentic workflows scoped to your application's data model. Generated agents are automatically registered in AI Control Tower for centralized governance, and all artifacts deploy through standard update sets.

For more information, see Build Agent.

Agentic development and vibe coding

Agentic development, or vibe coding, is an AI-driven approach to app development in which you build applications through conversations with AI. On the ServiceNow AI Platform, Build Agent makes agentic development possible.

For more information, see Agentic development on the ServiceNow AI Platform.

Autonomous Engineer

Autonomous Engineer is an agentic worker that accepts requirements, generates a structured implementation plan, and builds all work items in parallel using background agents. Autonomous Engineer uses Build Agent, a separate AI agent, as its execution layer. Autonomous Engineer helps teams move from a defined story to a working application in less time.

For more information, see Autonomous Engineer.

ServiceNow Otto for Creator

ServiceNow Otto for Creator includes generative and agentic AI capabilities that can make developing on the ServiceNow AI Platform more efficient. You can use the generative AI skills available with ServiceNow Otto for Creator, app summary generation, flow generation, and UI generation. ServiceNow Otto for Creator also includes Build Agent, an autonomous AI agent that can develop full-stack applications ready for deployment through conversations using natural language.

For more information, see ServiceNow Otto for Creator.

ServiceNow Otto for App Engine

ServiceNow Otto for App Engine enables you to enhance custom applications with AI agents and skills that application users can access at application runtime. With ServiceNow Otto for App Engine, you have access to Platform skills, AI agents, and agentic workflows, and the tools for developing custom AI assets using AI Skill Kit and AI Agent Studio.

For more information, see ServiceNow Otto for App Engine.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Developer Sandboxes

Developer Sandboxes provide admins and delegated developers the ability to request, access, and manage individual sandboxes on top of the same underlying development instance. Delegated developers can write and merge code and configuration changes without the risk of their changes getting overwritten on the instance mid-development.

For more information, see Developer Sandboxes.

Autonomous Engineer

Autonomous Engineer is an agentic worker that augments product development capacity across the software development lifecycle (SDLC). It accepts requirements, generates a plan with structured work items, and builds all work items in parallel using background agents.

Autonomous Engineer reduces time to value by taking your specifications as input and driving the full implementation lifecycle end-to-end. It helps to ensure quality by applying ServiceNow AI Platform best practices during planning and implementation, and automatically generating and running Automated Test Framework (ATF) tests for each work item.

Autonomous Engineer is available in ServiceNow Studio and supports custom applications as well as agent pack implementations.

Get started

Explore  Learn how Autonomous Engineer works and when to use it.	Configure  Set up Autonomous Engineer on your instance.
Use  Run Autonomous Engineer to build applications from requirements.	Reference  Reference information for Autonomous Engineer, including work item states and agent packs.

Helpful resources

The following resources provide additional information:

Build Agent docs

Build Agent

Agentic development overview

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Agentic development on the ServiceNow AI Platform

Support

Contact Customer Service and Support

Exploring Autonomous Engineer

Autonomous Engineer is an agentic worker that accepts requirements, generates a structured implementation plan, and builds all work items in parallel using background agents.

Autonomous Engineer overview

Implementing applications on the ServiceNow AI Platform can require significant manual effort between defining requirements and delivering value to users. Autonomous Engineer reduces that time to value by taking your specifications as input and driving the full implementation lifecycle end-to-end.

Autonomous Engineer follows two sequential phases: planning and execution. You remain in control throughout both phases. Autonomous Engineer requires your review and approval before building.

Autonomous Engineer is available in ServiceNow Studio and supports custom applications. Autonomous Engineer uses agent packs with product-specific domain knowledge to run implementations.

How Autonomous Engineer works

Planning phase

Provide your requirements in any of the following forms:

- A natural language prompt or paragraph.
- A file upload, such as a CSV of user stories.

Autonomous Engineer reads your requirements and conducts an interview to resolve ambiguous or incomplete details. It queries your instance to identify existing artifacts and metadata such as tables, roles, fields, and catalog items, so existing artifacts aren't duplicated.

The output of planning is a plan with work items. Work items are structured in an Agile user story format and include a description and acceptance criteria. They are grouped into logical sections and sequenced using a dependency graph so that each work item executes only after its dependencies are complete.

Review every work item before execution begins. You can read the details, edit the content, and make changes to align the plan with your requirements. Nothing is built until you approve the plan and its work items.

Execution phase

After you approve the plan, Autonomous Engineer generates a background agent for each work item and builds all work items in parallel. Each agent runs in isolation. The plan dashboard displays the status of each work item as it progresses.

For each work item, Autonomous Engineer builds the required metadata, runs Test Agent to generate Automated Test Framework (ATF) tests, and runs those tests. If tests fail, Autonomous Engineer identifies the root cause and attempts to resolve the failure through several retry cycles. Work items that can't be resolved automatically appear in the plan dashboard and the chat panel for your review.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

When all work items are complete, Autonomous Engineer generates an update set for the plan. Export the update set and deploy it for user acceptance testing or to a production environment.

To view and manage all your plans across sessions, use the **Plans** view in the Build Agent navigator panel. For more information, see Plans in Autonomous Engineer.

Manual steps

Some implementation tasks fall outside what Autonomous Engineer can perform autonomously, such as activating plugins. When a plan includes steps that require manual action, Autonomous Engineer lists those steps in a manual verification section before you run the plan. Completing manual steps before execution reduces the risk of work items failing because of missing dependencies.

Autonomous Engineer benefits

Autonomous Engineer accelerates application development on the ServiceNow AI Platform by enabling:

- Parallel execution of work items, reducing total implementation time.
- Automatic ATF test generation and self-healing retry cycles.
- Structured planning with dependency sequencing so artifacts are built in the correct order.
- A single update set output, ready for deployment to test or production environments.
- Consolidation of requirements across multiple sources and systems.
- Clarification and validation of complex requirements.

What to explore next

For more information about configuring and using Autonomous Engineer, see:

- Configure Autonomous Engineer
- Using Autonomous Engineer
- Autonomous Engineer reference

Autonomous Engineer and Build Agent

Autonomous Engineer and Build Agent are separate products that work together. Autonomous Engineer uses Build Agent as its execution layer.

Separate products

Autonomous Engineer and Build Agent are separately entitled products. Check your entitlements to determine your access to each. You can use Build Agent without Autonomous Engineer, but Autonomous Engineer requires Build Agent to run.

Build Agent

An autonomous AI agent that creates and updates ServiceNow applications through a conversational chat interface in ServiceNow Studio and the ServiceNow IDE. You interact with Build Agent directly through natural language to build, edit, and deploy applications to update sets.

Autonomous Engineer

An agentic worker that accepts requirements, generates a structured implementation plan, and builds all work items in parallel using background agents. Autonomous Engineer handles large or complex implementations where

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

building each artifact manually is not practical. Autonomous Engineer is available in ServiceNow Studio.

Note: Autonomous Engineer is a separate application available in the ServiceNow Store. Installing Autonomous Engineer also installs Build Agent as a dependency. Installing Build Agent alone does not install Autonomous Engineer.

How they work together

Autonomous Engineer uses Build Agent as its execution layer. When Autonomous Engineer builds work items in the execution phase, it does so through Build Agent. The Build Agent chat panel displays status updates and surfaces work items that require your review during execution.

Both products are available in ServiceNow Studio. You can use the Build Agent chat panel to interact with either product depending on your development needs. Use Build Agent for conversational, incremental development. Use Autonomous Engineer when you need to go from a full set of requirements to a complete implementation without building each artifact manually.

What is shared

The following are shared between Autonomous Engineer and Build Agent:

ServiceNow Studio environment

Both products run in ServiceNow Studio. The Explorer panel, editor tabs, and open files are in context for both products.

Build Agent chat panel

The chat panel in ServiceNow Studio is the interface for both products. During an Autonomous Engineer run, the chat panel surfaces work items that need your attention and displays progress from background agents.

Test Agent integration

Both products generate and run Test Agent tests as part of the build process to validate artifacts before deployment. For more information, see Test what you built with Autonomous Engineer.

Tools

Both products use the same set of tools to support application development tasks such as semantic search, schema inspection, code search, UI validation, database querying, and script execution. For more information, see Supported tools for Autonomous Engineer.

Models

Both products draw from the same set of supported AI models. You can change the active model from within either product. For more information, see Supported models for Autonomous Engineer.

Settings

Configuration settings for Autonomous Engineer and Build Agent are managed in the same location. Settings that apply to Build Agent also apply when Autonomous Engineer runs. For more information, see Configure Autonomous Engineer.

Deployment

Both products use the same deployment methods to move applications from development to production environments, including update sets and the application repository. For more information, see Deploying what you built with Autonomous Engineer.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Agent packs for Autonomous Engineer

Agent packs bundle the domain knowledge, tools, and skills that Autonomous Engineer uses to implement ServiceNow products.

When you use Autonomous Engineer to implement a ServiceNow product, Autonomous Engineer loads an agent pack to understand that product. An agent pack bundles the domain knowledge, tools, and skills that background agents require to generate accurate, product-specific artifacts during plan execution.

Agent packs are active during Autonomous Engineer execution.

Note: You must install the Autonomous Engineer app from the ServiceNow Store. Check your entitlements to see whether you can use Autonomous Engineer.

How agent packs work

All agent packs share a common base that provides the core capabilities available to every background agent during Autonomous Engineer execution. Product-specific packs extend this base with product-specific knowledge. When you run a plan that involves a ServiceNow product, Autonomous Engineer loads the corresponding agent pack. That agent pack is available to every background agent working on a work item for that product.

Each agent pack also includes Automated Test Framework (ATF) Test Agent capabilities for the product it covers. When a background agent completes a work item, it uses Test Agent in the agent pack to generate tests. The tests validate the work item against the acceptance criteria in your plan.

For details on Test Agent use in Autonomous Engineer, see Test what you built with Build Agent.

For details on using agent packs to implement ServiceNow products and applications, see Use Autonomous Engineer to implement an application.

Agent packs and custom applications

Agent packs are available in the Brazil EA (Patch 0) release:

- The Planning pack, which parses requirements and interviews you for clarity, applies ServiceNow best practices, and generates a structured plan of work items.
- The Platform development pack, which provides domain knowledge for implementing custom applications on your instance. The Platform development pack gives Autonomous Engineer awareness of platform tables, roles, and configuration patterns specific to custom app development. The awareness helps generated work items and artifacts align with standard platform conventions.

Agent packs are specific to ServiceNow products. Autonomous Engineer applies instance awareness and your requirements to generate the plan without product-specific domain knowledge.

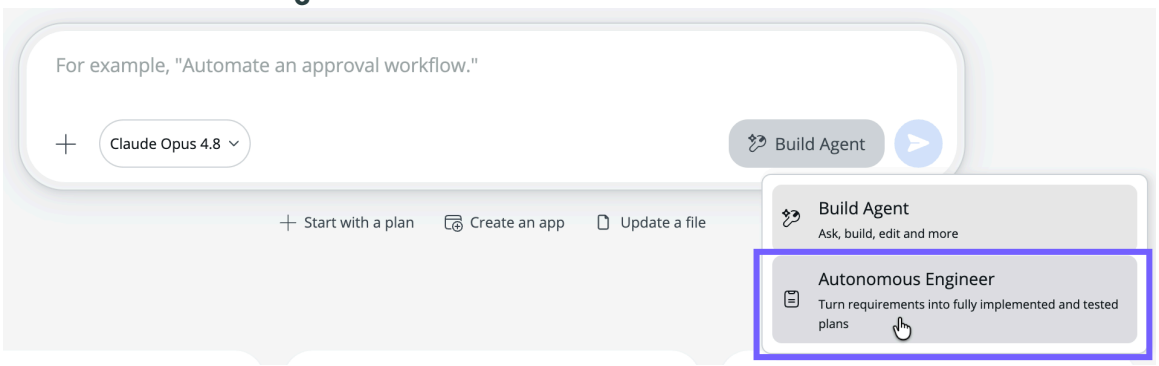
If your implementation spans a ServiceNow product and custom application development, Autonomous Engineer will apply the appropriate agent packs. The agent pack applies only to work items that target the product. Work items for custom artifacts are handled by the base implementation capabilities.

Autonomous Engineer chat panel

The Autonomous Engineer chat panel within Build Agent is where you interact with the AI agent during planning and development. Use it to submit requests, review responses, run tests, and apply generated code.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Select Autonomous Engineer



For details on using the chat panel, see Build Agent and Autonomous Engineer chat panel.

Using the dashboard

The Autonomous Engineer dashboard displays the status of every work item in your plan and provides navigation to individual work items, milestones, and pre-flight verification items.

After you approve a plan, the dashboard opens in Autonomous Engineer. The dashboard updates as background agents complete work items, so you can track progress without leaving the editor. The dashboard has two tabs: **Overview** and **Work items**. To return to a previous plan or find plans created by other users, use the **Plans** view in the Build Agent navigator panel. For more information, see Plans in Autonomous Engineer.

Overview tab

The **Overview** tab shows the plan header and two sections: **Pre-flight verification** and **Milestones**.

The plan header displays the plan title, a status badge, and a last-updated timestamp. The status badge reflects the overall state of the plan as work items progress.

Pre-flight verification

The pre-flight verification section appears when one or more work items have blockers that require action before or during execution. The section has two subsections:

Install required plugins

Lists any plugins that work items depend on, with the installation state of each. Use the inline link to navigate to the plugin installation page for each plugin listed.

Complete manual work items

Lists implementation steps that Autonomous Engineer can't perform autonomously. Each item includes instructions for completing the step manually. Complete all manual steps before execution begins to help reduce the risk that dependent work items fail.

When all pre-flight blockers are resolved, the section collapses.

Note: The section does not appear for plans that have no pre-flight requirements.

Milestones

The **Milestones** section lists the subplans that make up the current plan. Each row shows the milestone name and the number of work items it contains. Select the

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

drill-down arrow on a milestone row to open that subplan's own **Overview** tab. Use the breadcrumb navigation at the top of the dashboard to return to a parent plan at any depth.

The dashboard updates the work item counts and statuses in the **Milestones** section as execution progresses.

Update sets

When the plan completes, a batch update set is available and includes all the update sets for the work items for the plan.

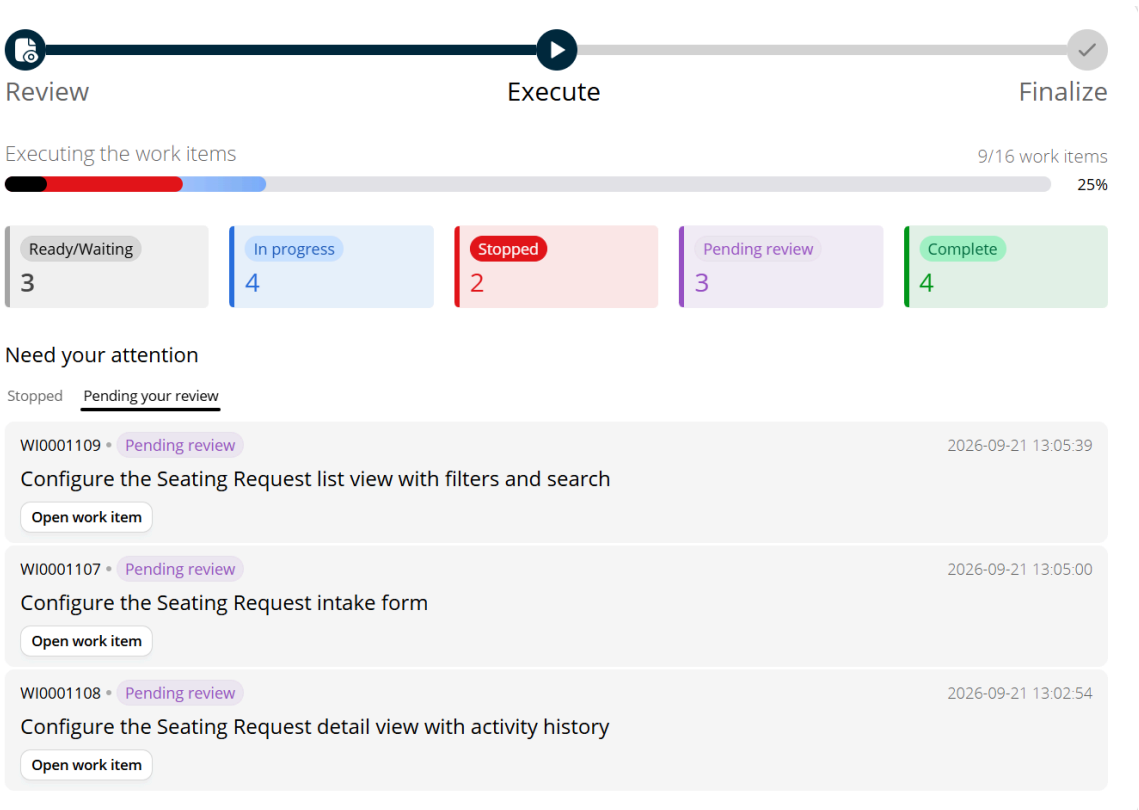
Work items tab

The **Work items** tab lists every work item in the current plan scope. Each row shows the work item name, its current state, and any relevant status indicators. The list updates incrementally as background agents progress through execution, so you don't have to refresh the dashboard.

Select a work item row to open the work item record. From there you can:

- Read the generated artifacts and Test Agent (ATF) test results. For more information, see Test Agent.
- Edit the work item content and ask Autonomous Engineer to retry.
- Roll back the changes for that work item if the output doesn't meet your requirements. For more information, see Checkpoints and conversation change log.
- When the work item completes, an update set is available and includes all the changes for the work item.
- Mark the work item complete when you're satisfied with the results.

Work items dashboard



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Work items that require your attention, for example because they failed automated retries or surfaced a question during execution, are highlighted in the list and chat panel.

Chat panel context

While the Autonomous Engineer dashboard is open, the chat panel displays the context of the active plan. Use the chat panel to ask questions about plan progress, check the status of specific work items, and respond to questions that background agents surfaced during execution.

Resilience in Autonomous Engineer

Autonomous Engineer monitors background agents during execution and automatically detects and retries work items that get stuck or unresponsive.

Autonomous Engineer runs work items in parallel using background agents. During execution, individual agents or jobs can get interrupted or unresponsive due to platform conditions, long-running operations, or transient failures. The resiliency framework monitors for these conditions and takes action automatically so that execution continues without requiring manual intervention in most cases.

Stuck agent detection

A scheduled job monitors all active work items during execution. When a background agent has not responded within a configurable time threshold, Autonomous Engineer treats that work item as stuck. The resiliency monitor identifies the agent handling the work item and cancels it, then retries the work item automatically.

Retry behavior

When a work item is retried, Autonomous Engineer generates a new background agent for it and attempts execution again. The plan dashboard reflects the updated state of the work item as the retry progresses.

Autonomous Engineer attempts three retry cycles for each work item. Work items that can't be resolved after all of the retry attempts appear in the plan dashboard and the chat panel for your review. You can edit the work item content and ask Autonomous Engineer to retry manually, or you can roll back the work item.

When a work item reaches the stopped state, Autonomous Engineer retries it automatically before surfacing it for manual intervention. If the work item can't be resolved within the automatic retry limit, it appears in the plan dashboard and the chat panel for your review.

Interrupted and long-running jobs

In addition to stuck agents, the resiliency monitor handles jobs that are interrupted by platform events such as instance restarts. Interrupted jobs are detected by the same scheduled monitoring job and are resubmitted automatically when the instance recovers.

Long-running jobs, such as jobs that are making progress but have been active longer than typical, are monitored separately. The resiliency framework does not cancel long-running jobs that show ongoing activity; it cancels only jobs that have stopped responding within the threshold period.

Supported models for Autonomous Engineer

Learn which AI models and versions Autonomous Engineer supports and how to change them. Use this information to verify compatibility and select the right model for your task.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Supported models

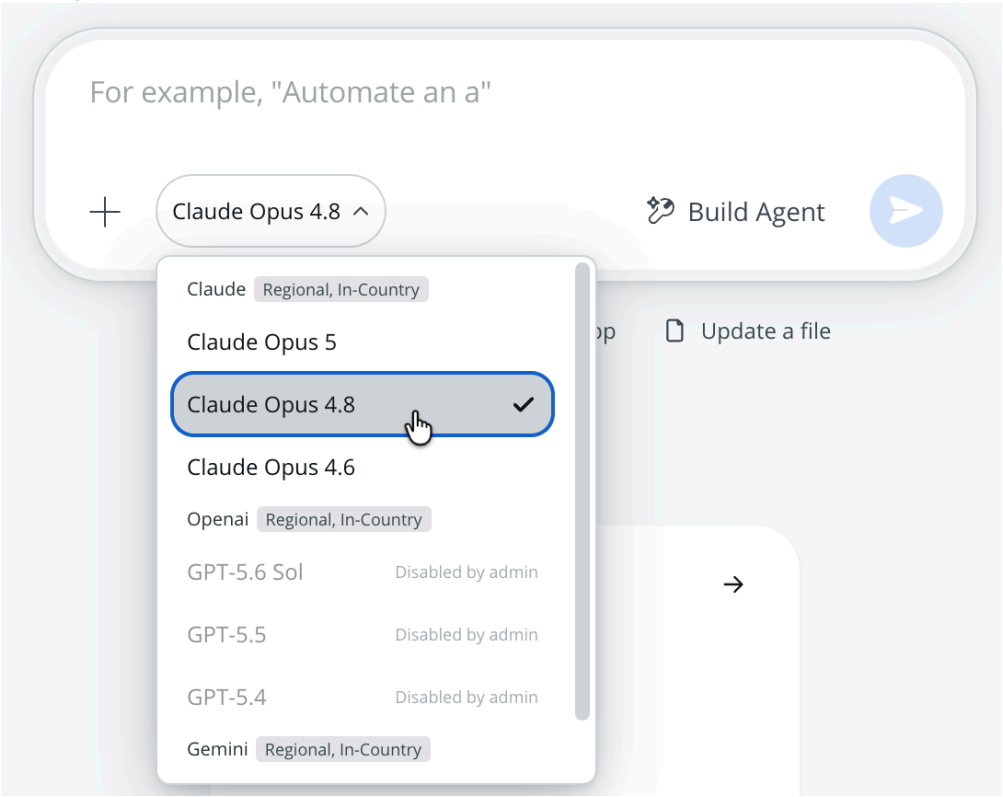
Autonomous Engineer currently supports the following models:

- Azure OpenAI 5.4
- Azure OpenAI 5.5
- Azure OpenAI GPT 5.6 Sol
- Google Gemini 3.7 Flash
- Google Gemini 3.5 Flash
- Gemini 2.5 Pro
- Claude Opus 4.6
- Claude Opus 4.8
- Claude Opus 5.0

Changing the model and version

You can change the version by selecting it directly in Build Agent or Autonomous Engineer.

Change model version



To change the model in use, navigate to the AI Admin Hub. For information on changing the model that Build Agent or Autonomous Engineer uses, see Manage model providers.

Supported tools for Autonomous Engineer

Autonomous Engineer tools support application development tasks such as semantic search, schema inspection, code search, planning, UI validation, database querying, app navigation, and script execution. Each tool extends what Autonomous Engineer can do during a build session.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Code search tool

The code search tool searches workspace files by exact text or regex patterns. Use the code search tool to find references to a field, script, or function across your app, and to understand the impact of a change before making it.

Open app tool

For applications that were not developed using the ServiceNow IDE, ServiceNow Studio, or the ServiceNow SDK, you must convert them into Fluent format to enable development within the Build Agent. You can prompt the Build Agent to use the open app tool to locate the application you want. Alternatively, you can search for an application directly within the Build Agent, and it will automatically use the open app tool. The open app tool can find an application, convert it to Fluent format, and then add the converted app to your workspace.

Planning tool

Build Agent and Autonomous Engineer include a planning tool that creates a detailed, step-by-step plan for your application development. You can refine the plan iteratively by prompting for changes and providing feedback until you reach a final version.

Rollback script tool

The rollback script tool reverts a previous run script tool operation on demand. After Build Agent or Autonomous Engineer runs a script, it returns a rollback context you can use to undo that run. Use the rollback script tool when a script run produces an unexpected result and you want to restore your instance to its previous state.

Only scripts run by Build Agent or Autonomous Engineer in the current session can be reverted using this tool. For details on security and governance around the rollback script tool, see Build Agent governance.

Run query tool

Build Agent and Autonomous Engineer can use the run query tool to query a specific table within your instance and return the top five records or derive specific insights.

Run script tool

The run script tool generates and runs a server-side script on your instance during a build session. Use the run script tool when you need Build Agent or Autonomous Engineer to perform an operation that goes beyond querying a table. For example, to retrieve data that requires custom logic, or to create something dynamically when no other tool covers the task.

Before the script runs, Build Agent or Autonomous Engineer shows you the generated script, its stated intent, and the application scope it will run in. You must approve the operation before it executes. The script runs in the scope of the app you're building by default. For details on security and governance around the run script tool, see Build Agent governance.

Semantic metadata search tool

The semantic metadata search tool is part of the Build Agent and Autonomous Engineer tool registry. Semantic search enables you to describe what you're looking for without providing the exact file name or keyword. For example, you can search for `files related to incident process management`, and the tool finds relevant results based on semantic understanding, not just keyword matching. Indexing keeps results current as your instance evolves.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Build Agent uses semantic search in two ways:

Implicit search

Build Agent and Autonomous Engineer determine when to use the semantic search tool based on your development task. When you describe what you want to accomplish, Build Agent or Autonomous Engineer searches for existing files and applications that match your intent. It then presents the results in a plan for your approval, which helps prevent duplicate content creation.

Explicit search

You can directly request a search by asking Build Agent or Autonomous Engineer to find something on your instance. Use explicit search when you want to locate specific files, applications, or knowledge without making changes.

Table schema inspector tool

The table schema inspector tool returns every field on a table, including field names, types, mandatory flags, and reference targets. Use the table schema inspector tool to understand a table's structure before adding fields or writing logic against it, and to verify the schema of base system tables.

UI validation tool

Build Agent and Autonomous Engineer detect UI components created during the build process, then analyzes and validates the UI using the UI validation tool. The tool automatically uses skills available in the Playwright MCP and executes them on the Automated Test Framework (ATF) Cloud Runner infrastructure.

You can specifically prompt UI validation by asking Build Agent or Autonomous Engineer to `validate the UI built using the ATF cloud runner tooling`. The UI validation tool is available for Build Agent in both ServiceNow Studio and the ServiceNow IDE.

Note: The ATF test generator and Cloud Runner app must be installed on the instance to use the UI validation tool.

Web search tool

Two Build Agent and Autonomous Engineer tools support external search.

- Use the web search tool to find information on the public web.
- Use the web fetch tool to retrieve content from a URL that you supply.

Note: The web search tool must be enabled in the Build Agent or Autonomous Engineer settings. For more information, see Configure web search.

MCP connections and Autonomous Engineer

MCP connections enable Autonomous Engineer to access external tools and resources through standardized communication, extending the planning and implementation workflow with data and context from third-party systems.

MCP is an open protocol that defines how AI agents communicate with external systems to discover and invoke tools, retrieve resources, and share context. By standardizing this communication layer, MCP enables AI agents to interact with a wide range of third-party applications without requiring custom integration logic for each one.

In Connect Hub, an MCP connector represents a configured connection between ServiceNow and an external system that exposes a server compatible with MCP. After an administrator sets

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

up an MCP connector and you authenticate your connection, Autonomous Engineer can use it to access tools and capabilities from the external system. This enables Autonomous Engineer to pull in external context during planning and to create records in connected systems as part of implementation.

How Autonomous Engineer uses MCP connections

Autonomous Engineer and Build Agent share the same MCP connection registry. Connections that are approved, authenticated, and enabled for Build Agent are available to Autonomous Engineer in the same session. You don't have to configure connections separately for each product.

Autonomous Engineer can use MCP connections at any stage of the workflow. For example, during planning, Autonomous Engineer can read requirements from a connected project management tool. During implementation, it can create or update records in connected systems as part of executing a plan.

Approve and activate MCP servers

Before you can enable an MCP server in Build Agent, an administrator must approve it as an AI asset in AI Control Tower. Each MCP server requires this approval, regardless of whether the server is enabled by default.

The end-to-end flow for making an MCP server available is:

1. The administrator adds the MCP server as a Workflow Data Fabric (WDF) connection.
2. The administrator approves the server as an AI asset in AI Control Tower.
3. You authenticate the connection in Personal Integrations.
4. You enable the MCP server in Build Agent settings.

For details on enabling MCP connections, see Client registration using custom connector.

Individual MCP servers are enabled by default, but the complete flow must be completed before any server is available for use.

For details on adding a new MCP connection in Workflow Data Fabric, see Model Context Protocol connectors.

Supported MCP servers

The following MCP connections are currently supported for Build Agent:

- Atlassian Rovo
- AWS DevOps
- Box
- Docusign
- Figma
- Linear
- Miro
- Prisma Postgres
- Zoom
- Zoom Chat

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Zoom Docs
- Zoom Whiteboard

Note: Because Build Agent doesn't support multiple servers connecting to one app, each Zoom integration requires its own app in Zoom.

View available tools per MCP server

You can view the tools available for each connected MCP server in the **MCP servers** tab in the Build Agent settings panel. Expand a server by selecting the chevron next to the server name to see its tools and descriptions. Because Autonomous Engineer runs within the Build Agent interface, the same pane applies when Autonomous Engineer is the active mode.

How to configure MCP connections

For details on configuring MCP connections, see Connect Build Agent to a supported MCP server. Because Autonomous Engineer shares the Build Agent settings panel, the configuration steps are the same for both products. For more information on MCP connections on the platform, see Additional connector configurations.

Playbooks in Autonomous Engineer

Use Autonomous Engineer to author and manage Playbook Designer artifacts through a conversation. You can generate playbook structures, configure activities, set runtime permissions, and define launcher configurations without manually navigating the Playbook Designer UI.

For details on using playbooks in Autonomous Engineer, see Playbook authoring with Build Agent and Autonomous Engineer.

Autonomous Engineer governance

Governance controls for Autonomous Engineer are managed through Build Agent.

Autonomous Engineer uses Build Agent as its execution layer. Governance controls that apply to Build Agent also apply when Autonomous Engineer runs. For full governance information, including AI Control Tower integration, cross-scope privileges, background script containment, and custom skills and rules, see Build Agent governance.

Autonomous Engineer limitations

Limitations that apply to Build Agent also apply to Autonomous Engineer.

Autonomous Engineer depends on Build Agent as its execution layer. Limitations that apply to Build Agent, including constraints on regulated environments, deployment capabilities, and performance, also apply when Autonomous Engineer runs. For full limitations information, see Build Agent limitations.

Configure Autonomous Engineer

Configure Autonomous Engineer by installing it from the ServiceNow Store and enabling the features required for planning and parallel execution on your instance.

Autonomous Engineer is a separate application available in the ServiceNow Store. Installing Autonomous Engineer also installs Build Agent as a dependency. Configuring Autonomous Engineer involves installing the application, confirming the required roles and permissions, and optionally installing agent packs to extend Autonomous Engineer with product-specific domain knowledge.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Prerequisites

Before configuring Autonomous Engineer, verify the following:

- Autonomous Engineer is installed on your instance. For installation instructions, see Install Autonomous Engineer.
- You have the admin role.

Installation

For installation steps, see Install Autonomous Engineer.

Agent packs

Autonomous Engineer uses agent packs with product-specific domain knowledge to run implementations. For more information, see Agent packs for Autonomous Engineer.

Install Autonomous Engineer

Install Autonomous Engineer from the ServiceNow Store to make it available on your instance.

Before you begin

Autonomous Engineer is a separate application available in the ServiceNow Store. Installing Autonomous Engineer also installs Build Agent as a dependency. Installing Build Agent alone does not install Autonomous Engineer.

- Check your entitlements to confirm your access to Autonomous Engineer before installing.
- Review the Autonomous Engineer application listing in the ServiceNow Store for information on dependencies, licensing or subscription requirements, and release compatibility.
- You can use Autonomous Engineer on a Personal Development Instance (PDI) or in Developer Sandboxes. For more information, see Accessing Build Agent in ServiceNow Studio.

Role required: admin

About this task

For instructions on installing ServiceNow products from the ServiceNow Store, see Install an application or plugin.

Procedure

1. From the Autonomous Engineer application page on the ServiceNow Store, select **Buy**.
2. After approval has been granted, on your instance, navigate to **All** > **System Applications** > **All Available Applications** > **All**.
3. Using the search bar, search for the Autonomous Engineer application.
4. Select **Install**.
   Autonomous Engineer and Build Agent are installed on your instance.
5. Enable the Autonomous Engineer skills:
   a. Navigate to **All** > *AI Admin Hub***Skills**.
   b. Select the **Creator** tab.
   c. Locate Autonomous Engineer prompt, Autonomous Engineer work item creation, and Autonomous Engineer work item execution in the skills list and select **Turn on** for each of them.

   The skills are enabled for all admin users.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

What to do next

After installing Autonomous Engineer, install the agent pack for the product domain you plan to implement. For more information, see Agent packs for Autonomous Engineer.

Configuring Autonomous Engineer settings in Build Agent

Autonomous Engineer and Build Agent share the same configuration settings. Settings you configure for Build Agent also apply when Autonomous Engineer runs.

Because Autonomous Engineer is powered by Build Agent as its execution layer, both products use the same settings. For information on configuring these settings, see Build Agent configuration.

Using Autonomous Engineer

Use Autonomous Engineer to provide requirements, review an implementation plan, and build all work items in parallel using background agents in Build Agent.

Use Autonomous Engineer in ServiceNow Studio to go from requirements to a working implementation without building each piece of metadata manually. Autonomous Engineer guides you through two phases: planning and execution. You review and approve the plan before any building begins.

Provide requirements and review the plan

Start Autonomous Engineer from the chat panel in ServiceNow Studio and provide your requirements in any of the following forms:

- A natural language prompt or paragraph.
- A file upload, such as a CSV of user stories.

Autonomous Engineer conducts an interview to resolve ambiguous or incomplete details, then generates a plan with structured work items. Review each work item before approving the plan. You can open individual work items to read their details and make edits. Nothing is built until you approve.

If the plan includes manual steps, such as activating plugins, complete those steps before starting execution to reduce the risk of work item failures.

Monitor and review execution

After you approve the plan, Autonomous Engineer generates a background agent for each work item and builds all work items in parallel. Monitor progress from the plan dashboard, which displays the status of each work item as it progresses.

As work items reach a ready-for-review state, open each one to review the generated artifacts and Test Agent (ATF) test results. Mark the item complete when it meets your requirements. If your requirements change during execution, edit a work item and ask Autonomous Engineer to retry. You can also roll back changes for an individual work item if needed.

Work items that can't be resolved automatically through retry cycles appear in the plan dashboard and the chat panel for your review.

For more information on using Test Agent, see Test Agent.

Using the chat panel during a plan

While you navigate the plan and its work items, the chat panel remains in context of the plan. Use the chat panel to ask questions about the plan, check on the status of specific work items, and review any questions that background agents surfaced during execution.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Accessing Autonomous Engineer

Autonomous Engineer is available in ServiceNow Studio. You access it through the Build Agent chat panel by switching the active mode to Autonomous Engineer.

Before you begin

Autonomous Engineer must be installed on your instance before you can access it. For installation instructions, see Install Autonomous Engineer.

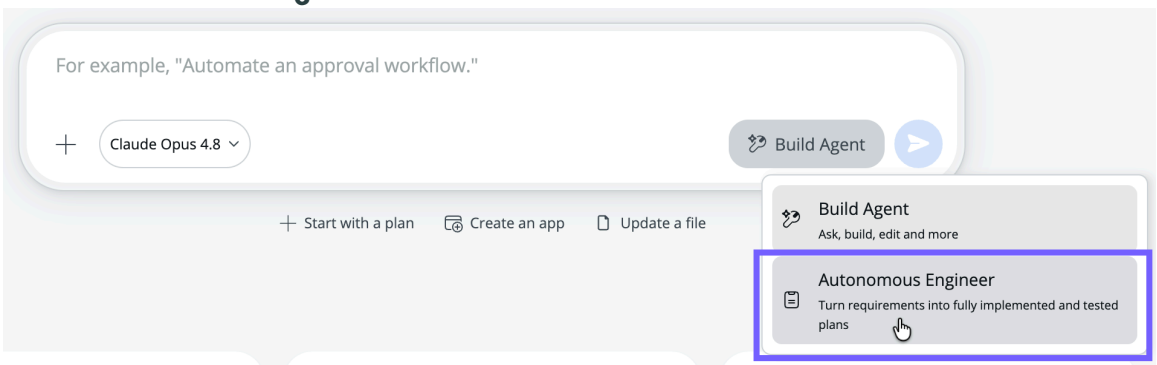
Note:

- Currently, only admins have permissions to use Build Agent.
- Autonomous Engineer is a separate application available in the ServiceNow Store. Installing Autonomous Engineer also installs Build Agent as a dependency. Installing Build Agent alone does not install Autonomous Engineer.

Opening Autonomous Engineer

When you open ServiceNow Studio, the Build Agent chat panel runs in Build Agent mode by default. To switch to Autonomous Engineer, select **Build Agent** from the mode selector in the chat panel and choose Autonomous Engineer.

Select Autonomous Engineer



After you select Autonomous Engineer, the chat panel updates to reflect the planning interface. Enter your requirements to begin the planning phase. For information on what to do next, see Using Autonomous Engineer.

Use Autonomous Engineer to implement an application

Use Autonomous Engineer (powered by Build Agent) to generate a plan from your requirements and build all work items in parallel, without creating each artifact individually. When implementing a ServiceNow product or application, Autonomous Engineer uses agent packs to give background agents product-specific domain knowledge.

Before you begin

Note: You must install the Autonomous Engineer app from the ServiceNow Store. Check your entitlements to see whether you can use Autonomous Engineer.

Role required: admin

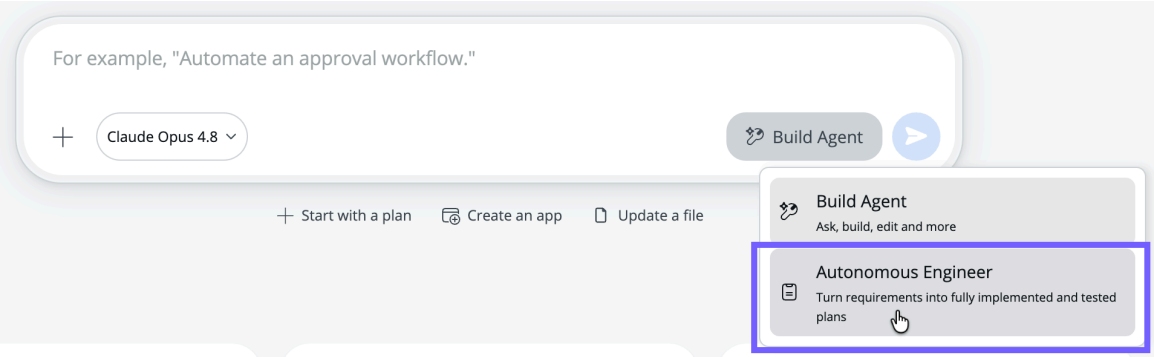
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

About this task

Procedure

1. Navigate to **All** > **App Development** > **ServiceNow Studio**.
2. In the chat panel, open the mode selector and select **Autonomous Engineer**.

Enable Autonomous Engineer



Note: Select **Autonomous Engineer** specifically to access the planning and parallel execution workflow described in this topic.

3. Provide your requirements using one of the following methods.

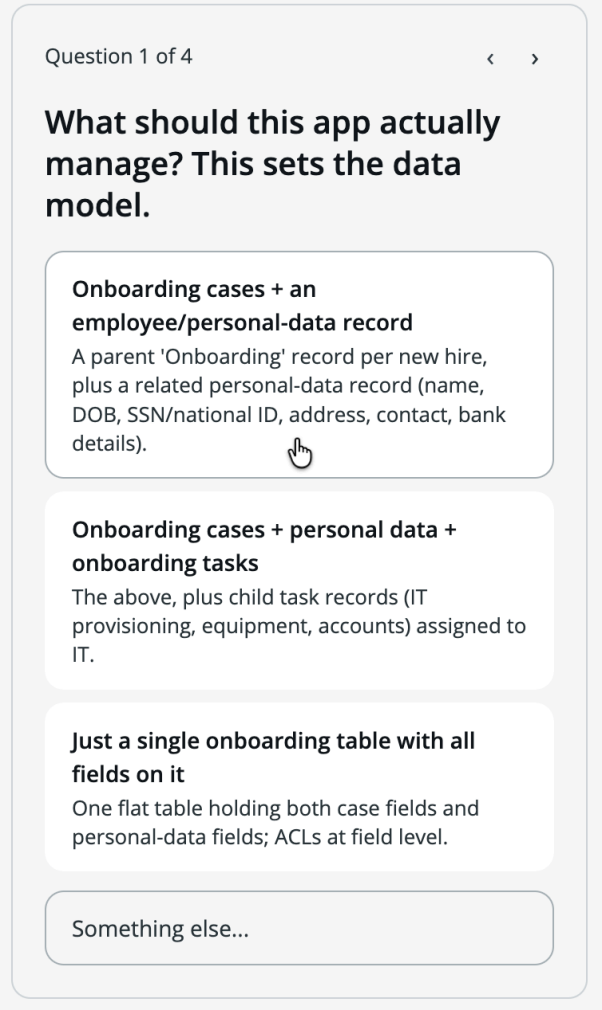
Ways to provide requirements

Method	Steps
Describe your requirements in plain language	Enter a description of what you plan to implement in the chat panel. Include as much detail as available. See Example prompts for some ideas.
Upload a file of requirements	Attach a file, such as a CSV of user stories, to the chat panel.

4. Answer any interview questions that Autonomous Engineer asks to clarify your requirements.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Autonomous Engineer asks clarifying questions



Autonomous Engineer processes your requirements and searches your instance to identify existing artifacts such as tables, roles, fields, and catalog items. If anything is ambiguous or incomplete, Autonomous Engineer asks follow-up questions. For each question, select an answer option or type a response. Autonomous Engineer then generates a brief from which to draft a plan.

5. Tell Autonomous Engineer whether you approve of the brief or want to make any changes. Review and update the acceptance criteria and test criteria for each work item as needed.

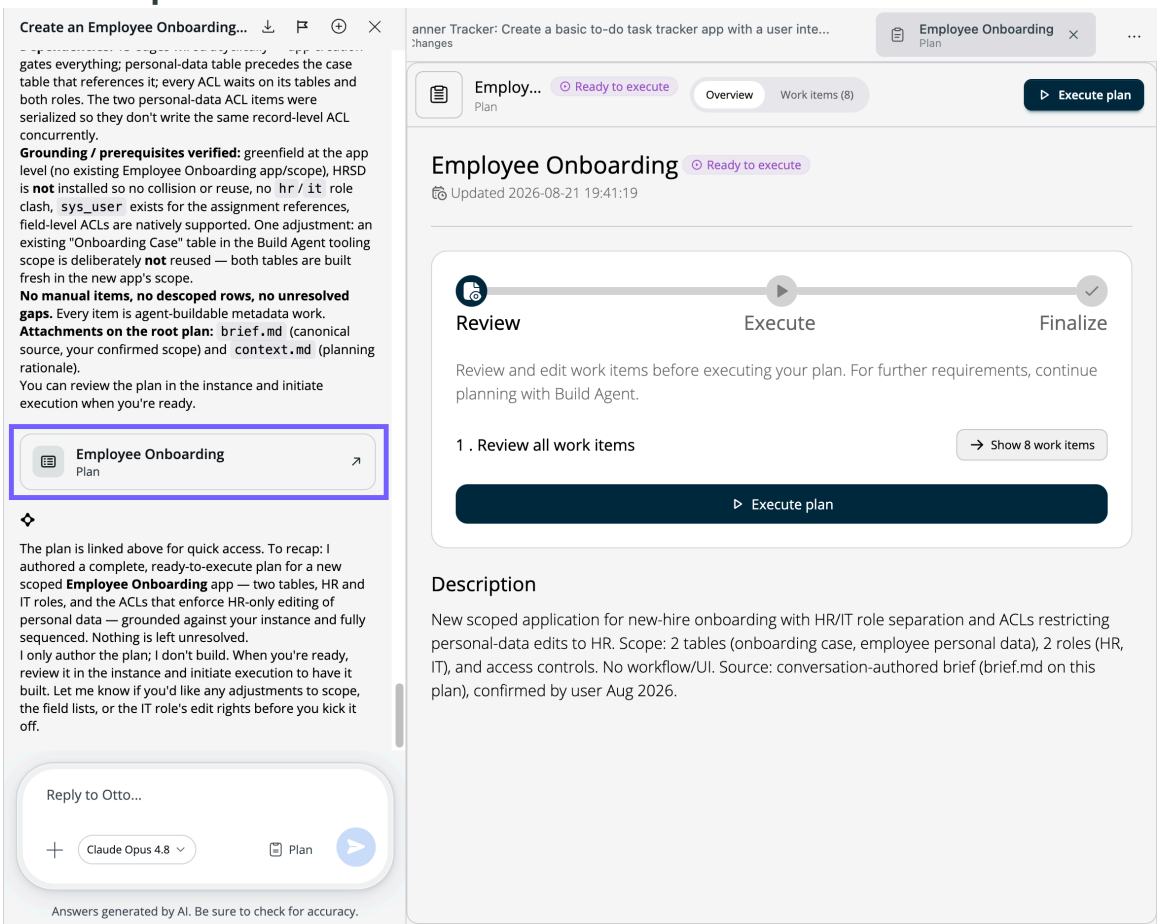
   Accurate acceptance criteria and test criteria improve execution quality and help Autonomous Engineer build and validate each work item correctly.

   Autonomous Engineer generates a plan with work items, and approves the work items. Work items are structured in an Agile user story format and include acceptance criteria and test criteria. They are grouped into logical sections and sequenced using a dependency graph so that each work item executes only after any items it depends on are complete.

6. Select the plan in the chat panel to review its contents and work items in a new window.

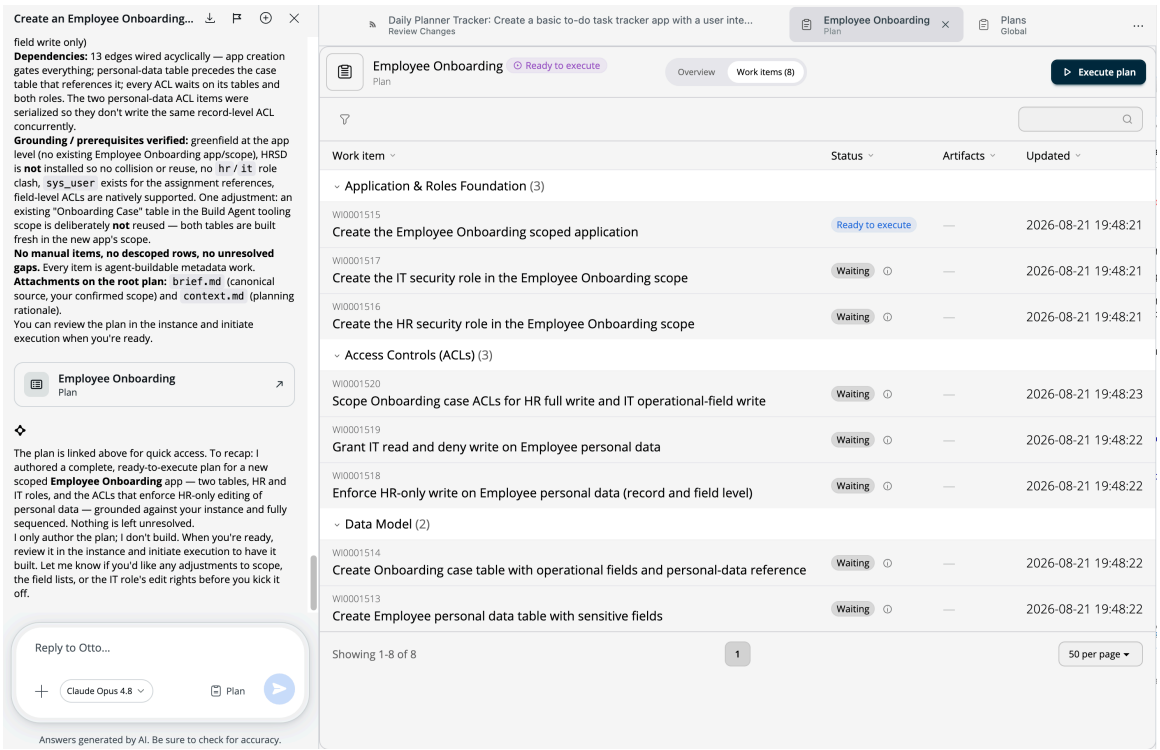
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Review the plan



7. Select the **Show work items** button to review the work items in the plan.

Work items



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

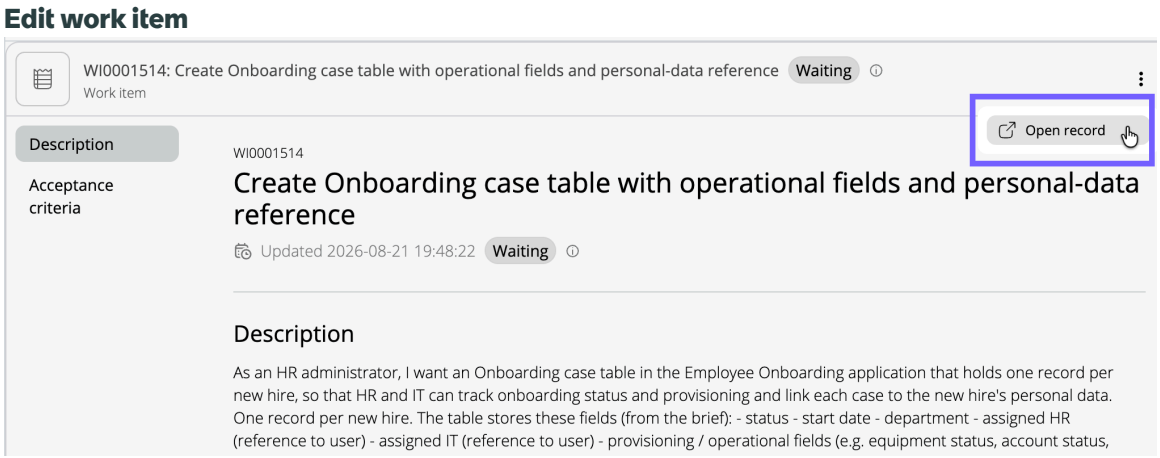
8. Review each work item.
9. **Optional:** Edit the content in the work item directly.
   You can edit work items using the chat panel.

   a. Select a work item to open it.

   b. Select the more options icon ⋮.

   c. Select **Open record**.
   The record for the work item opens in a new browser tab.

   d. Make any changes to the work item record and select **Update.**



10. Complete any required manual steps, if the plan includes them, before you run the plan.

    Work items that depend on an incomplete manual step may fail.

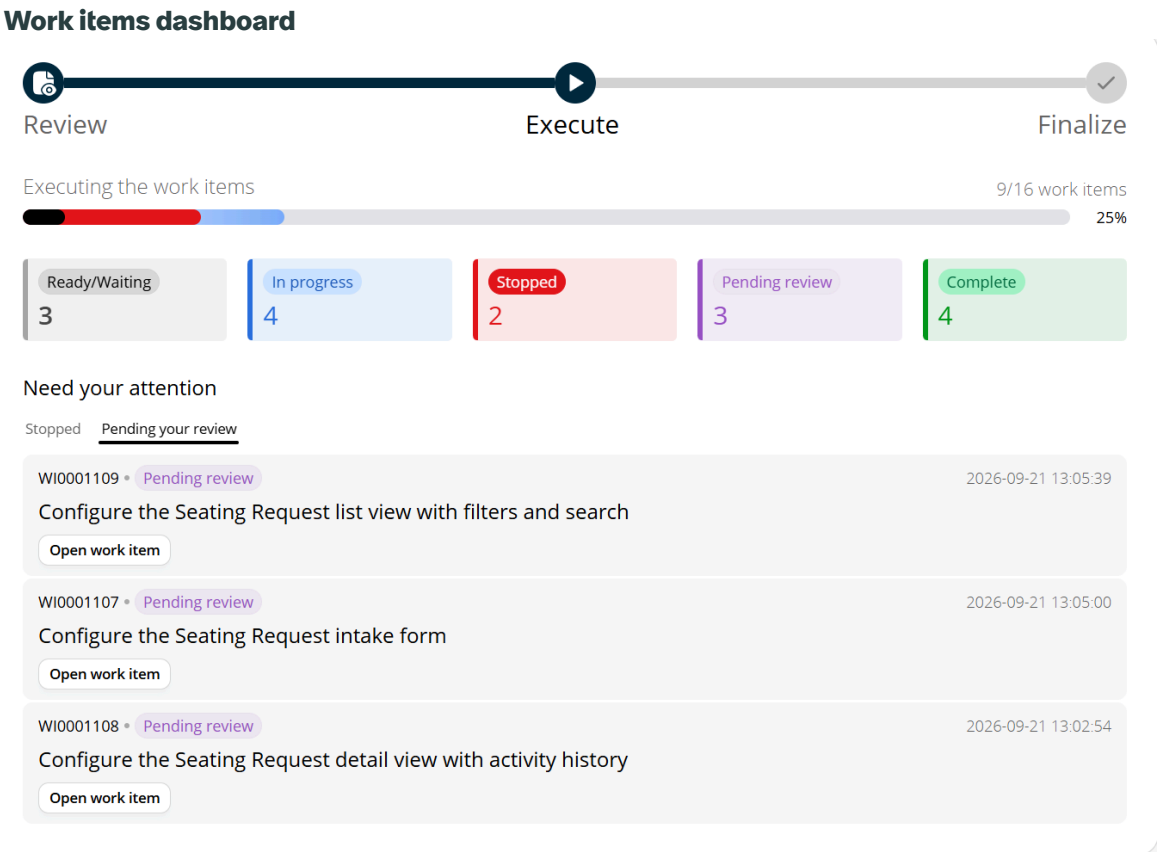
11. Select the **Execute plan** button on the **Plan** tab to approve the plan and start execution.
    The **Plan** tab view becomes a dashboard where you can track the plan progress. Autonomous Engineer generates a background agent for each work item and builds all work items in parallel.

    The dashboard displays the status of each work item as it progresses. For a description of each status, see Autonomous Engineer plan and work item states.

12. Monitor progress and respond to items that require your attention.
    The plan dashboard shows a count of work items by status. If a work item reaches a blocked or error state, it appears in the dashboard and a notification appears in the chat panel.

    To see more detail on a work item, open it from the dashboard or ask Autonomous Engineer in the chat panel. To retry a blocked work item after reviewing or editing it, ask Autonomous Engineer to retry that item. If you must undo changes for a specific work item, ask Autonomous Engineer to roll back that item.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



13. When a work item reaches the Needs validation status, open it to verify the results and mark it complete.
    A work item in Needs validation status shows a validation section with Test Agent (ATF) test results and a list of the artifacts that were created. If you used an agent pack, the tests are product-specific.

    If the results look correct, select **Mark as complete**. To make changes, edit the work item content and ask Autonomous Engineer to retry.

Result
When all work items are complete, Autonomous Engineer generates an update set for the plan. Export the update set and deploy it to your UAT or production environment.

For information on deploying update sets, see Update sets and Build Agent.

Plans in Autonomous Engineer

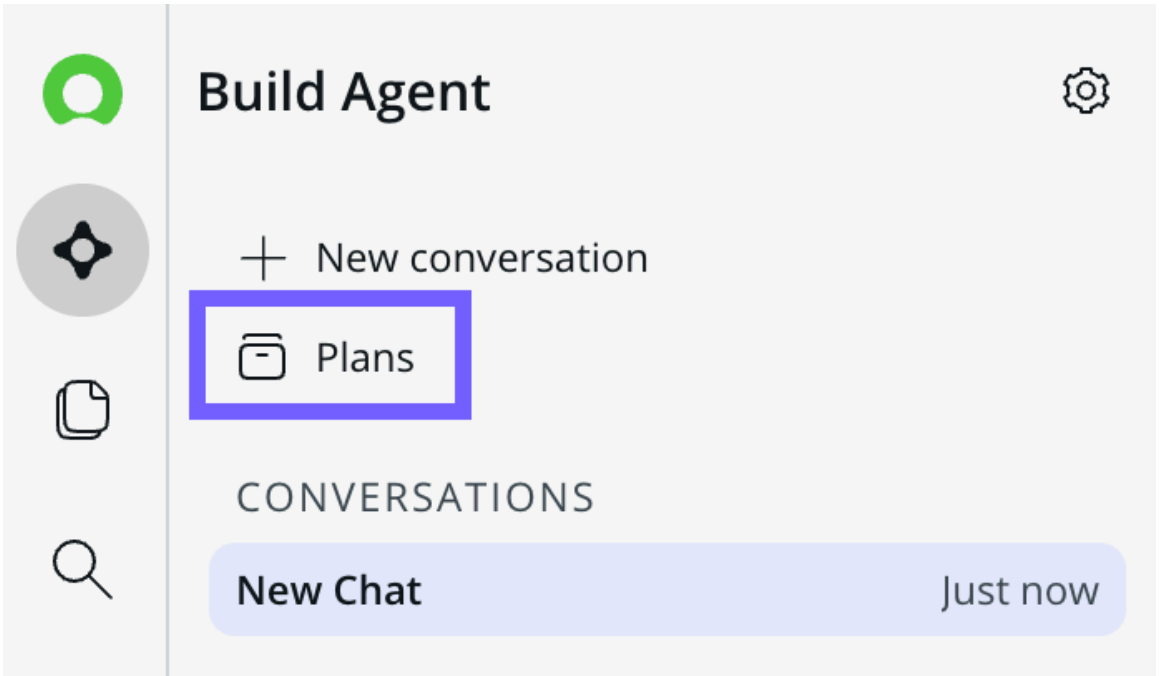
Access all Autonomous Engineer plans from the **Plans** view in the Build Agent panel. The **Plans** view shows the plans you created and all plans available on the instance.

Access the Plans view

To open the **Plans** view, select the Conversations icon in the Navigator panel, and then select the Plans icon in the Build Agent panel that appears.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Access the Plans view



The **Plans** view lists all Autonomous Engineer plans in the current application scope, and has two tabs.

Created by me

Plans that you created in the current application scope.

All plans

All plans available in the current application scope, including plans created by other users.

Start a plan

To create a plan, select **Create a new plan** from the **Plans** view. This opens a new conversation in the chat panel where you can provide your requirements. To create and manage plans using Autonomous Engineer, see Use Autonomous Engineer to implement an application.

Manage plans

From the **Plans** view, you can take the following actions on an existing plan:

Open a plan

Select a plan from the list to open the plan dashboard and review work item status, artifacts, and execution progress.

Application scope and plans

Plans are scoped to the application scope active when the plan was created. The current scope is shown in the breadcrumb at the top of the **Plans** view. To view plans from a different scope, change the application scope in the Build Agent panel before opening the **Plans** view.

Manage work items during execution

Monitor Autonomous Engineer work items as they build. You can review generated artifacts and test results, resolve work items that require attention, and export the update set when all work items are complete.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

An Autonomous Engineer plan must be in execution. To start a plan, see Use Autonomous Engineer to implement an application.

Role required: admin

About this task

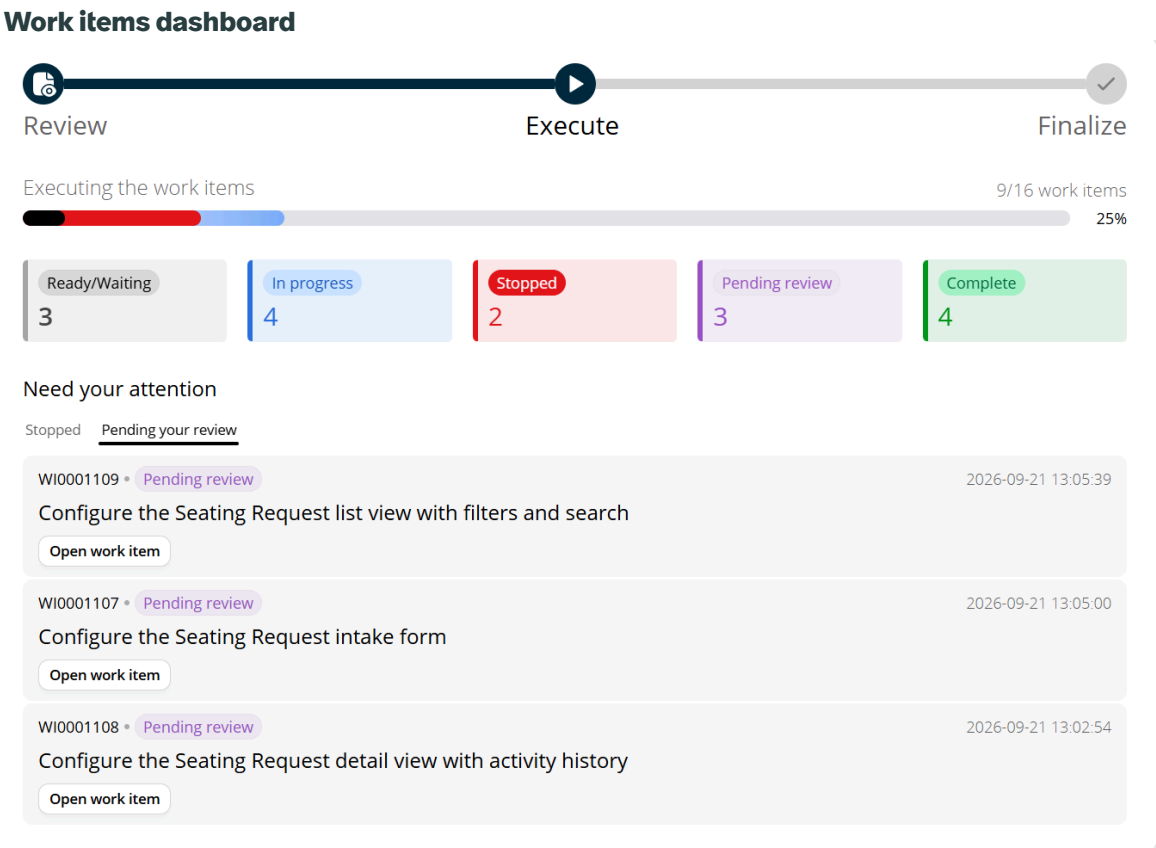
During execution, background agents build work items in parallel. You can review work items as they reach a ready-for-review state without waiting for all work items to complete. Work items that require your attention, for example, if a background agent generated a question, are highlighted in the plan dashboard and the chat panel.

Procedure

1. From the plan dashboard, open the **Work items** tab to monitor execution progress.

   The Work items tab lists every work item in the plan. Each row shows the work item name and its current state. The list updates as background agents progress. For a description of each state, see Autonomous Engineer plan and work item states.

2. When a work item reaches a ready-for-review state, select it to open the work item record.
3. Review the generated artifacts and Automated Test Framework (ATF) test results for the work item.
   Each work item record shows the artifacts that the background agent generated and the ATF test results. Verify that the artifacts meet the acceptance criteria listed in the work item before marking it complete.



4. Take one of the following actions based on your review.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Manage work items actions

Action	Steps
Mark complete	If the work item meets your requirements, mark it complete. The dashboard updates the work item state.
Edit and retry	If your requirements changed or the output does not match what you intended, edit the work item content directly in the work item record. Then ask Autonomous Engineer to retry. A new background agent runs for the work item.
Roll back	If you must undo the changes for a work item, roll back that work item. The rollback applies only to the selected work item and does not affect other work items in the plan.

5. Repeat steps 2 through 4 for each work item as it becomes ready for review.
6. When all work items are complete, export the update set that Autonomous Engineer generated for the plan.

   Autonomous Engineer generates a single update set for the plan when all work items reach a complete state. Export the update set from the plan dashboard and deploy it to a test environment for user acceptance testing, or promote it to production.

Result

Your implementation is captured in an update set that is ready for deployment.

What to do next

For information about work item states and their definitions, see Autonomous Engineer plan and work item states. For information about how Autonomous Engineer handles stuck or unresponsive agents automatically during execution, see Resilience in Autonomous Engineer.

View changes in the change log

When Autonomous Engineer completes a turn, it updates the change log.

To review what a background agent worker did during execution, open the work item and select the **Log** tab. The **Log** tab shows the chat history of the background agent, including the messages it sent and received during the work item build.

For information on how the Build Agent change log works, see Checkpoints and conversation change log.

Update sets and Autonomous Engineer

When you work with Autonomous Engineer, your changes are automatically tracked in update sets so you can review, revert, and deploy them without leaving ServiceNow Studio.

Autonomous Engineer tracks every change it makes to your application in update sets. Changes from each checkpoint in a conversation are captured together in a single update set. You can access, review, and open the update sets directly from the Autonomous Engineer chat panel. You can also open the update sets from the Current Changes List (CCL) page in ServiceNow Studio, without navigating to the platform.

Update sets use descriptive names to help identify what each update set contains. Names follow this pattern: *`application-name`*`build agent install 1`, `build agent install 2`, and so on.

After Autonomous Engineer creates a checkpoint, it automatically opens a manual edit checkpoint that captures any changes you make directly to your application outside of

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Autonomous Engineer. These manual edits are tracked in a separate update set named `manual edit 1`, `manual edit 2`, and so on.

- When a plan completes, a batch update set is available and includes all the update sets for the work items for the plan.
- When a work item completes, an update set is available and includes all the changes for the work item.

For general information about update sets on the ServiceNow AI Platform, see System update sets.

How Autonomous Engineer tracks changes

Autonomous Engineer automatically captures changes in a unified update set as you work.

Note: Manual changes aren't included in the automatically generated update set, and must be deployed separately.

The consolidated update set merges all the update sets created for previous agentic and manual development into a single update set for ease of deployment. The consolidation occurs after you tell Autonomous Engineer that you're done working on a task and no longer require rollback.

Autonomous Engineer automatically captures changes to the app and metadata that you're working on.

A checkpoint is created automatically after you approve each task plan. When Autonomous Engineer reaches a checkpoint, the changes associated with that checkpoint are captured in the update set. You can view and open the relevant update set directly from each checkpoint in the chat panel. For more information on checkpoints, see View changes in the change log.

When Autonomous Engineer prepares to create the next checkpoint, it checks whether any changes exist in the manual edit update set. If no changes are present, Autonomous Engineer removes the update set. If changes exist, Autonomous Engineer completes the update set and renders it with the other update sets from your conversation. You can access manual edit update sets directly from the checkpoints panel in your Autonomous Engineer conversation.

Deploy update sets

After your changes are ready, you can find the update sets from your Autonomous Engineer session on the **Deployment** tab on the ServiceNow Studio home page.

Test what you built with Autonomous Engineer

Test Agent generates test coverage for each work item that Autonomous Engineer builds, executes the tests, and performs root cause analysis (RCA) on failures.

To enable testing, see .

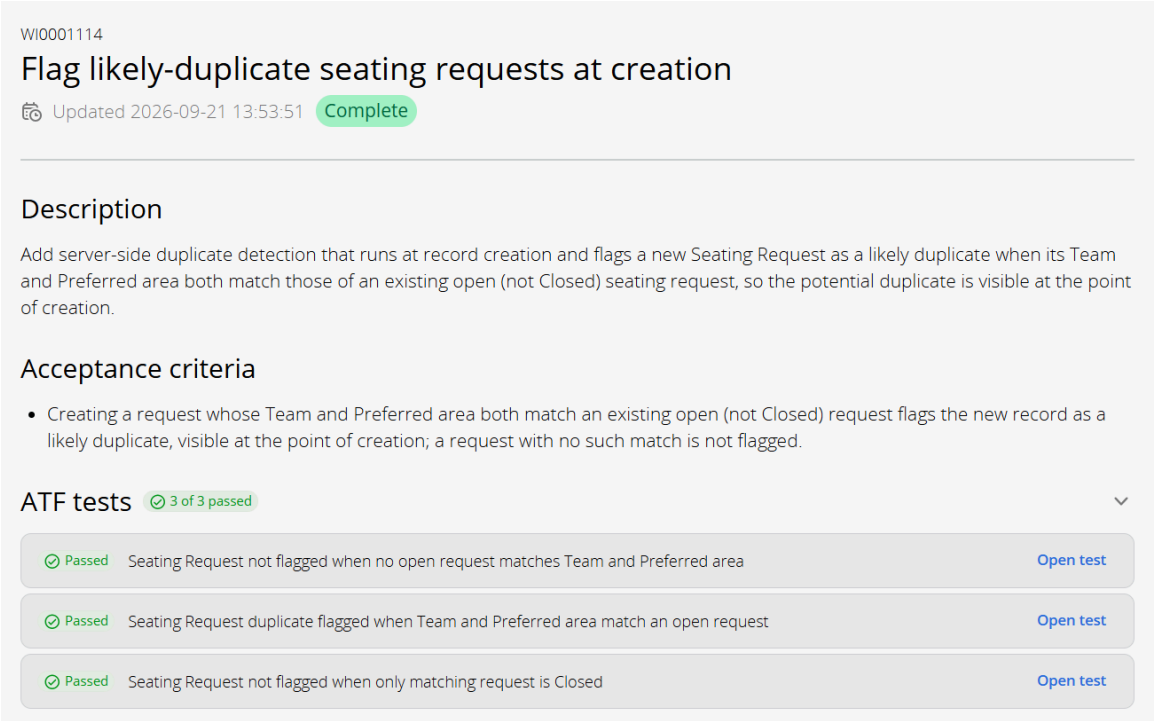
Autonomous Engineer runs Test Agent on each work item as part of plan execution. Build Agent provides the underlying execution layer for this testing. Tests are generated and run for each work item, and the results appear directly under the work item in the plan dashboard.

To enable testing in Autonomous Engineer, see Configure auto test prompting and UI tests.

For complete documentation on Test Agent, see Test Agent.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Tests under a work item in Autonomous Engineer



If a test fails, Test Agent performs an RCA. It then either applies safe fixes autonomously or surfaces actionable guidance in the chat panel so you can resolve the issue without leaving ServiceNow Studio.

Customer outcomes

Test Agent delivers the following outcomes when used with Autonomous Engineer:

- Build and test in one session: Test generation happens within the same plan execution, without a separate prompt or context switch.
- Faster failure triage: Automated RCA and proposed fixes reduce the time spent reviewing logs after a test run.
- More release confidence: Automated test execution provides verifiable evidence of code health before promotion to production instances.
- Generated ATF tests are stored in the sys_atf_tests [sys_atf_tests] table under the app scope for which they were created. You can schedule regression test runs using the generated tests.

Test generation

If automatic test generation is enabled in Build Agent settings, Autonomous Engineer will generate tests for each work item acceptance criteria.

Automatic prompting for tests reduces the chance of skipping test coverage during execution by keeping test generation part of the standard work item loop.

To manually prompt for tests, ask Autonomous Engineer to `Generate ATF tests for all the feature permutations on the app we built`. Then tell Autonomous Engineer to `Execute all ATF tests.`

Test Agent workflow

The end-to-end workflow is:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

1. Autonomous Engineer builds a work item in the execution phase.
2. Test Agent uses the work item context to generate contextually relevant functional, UI, and ATF tests.
3. Autonomous Engineer asks whether you want to run the tests.
4. Failures are automatically triaged. Test Agent produces an RCA and either applies safe fixes autonomously or proposes them through the chat panel.
5. Autonomous Engineer ingests the RCA and re-executes tests until a passing status is achieved, completing the auto-heal loop. Stale tests are automatically updated to reflect the newest functionality.
6. View tests in the **Work item** dashboard.
7. Optionally, ask Autonomous Engineer to create or run a test suite to execute multiple tests as a group and review consolidated results in the chat panel.

Scope and availability

Test Agent is available for Autonomous Engineer in the following environments and scopes:

Test Agent scope and availability

Dimension	Supported values
Authoring environment	ServiceNow Studio
Application scope	Global, custom, store
Test types	ATF functional tests, UI tests, and test suites
Execution target	Cloud Runner lanes

Note: Test execution requires the ATF Test Generator and Cloud Runner app to be installed and a cloud user set up. For more information, see ATF Test Generator and Cloud Runner.

Deploying what you built with Autonomous Engineer

When all work items are complete, Autonomous Engineer generates an update set that you export and move through the standard ServiceNow deployment process.

Workflow for deployment

Autonomous Engineer generates a single batch update set for the plan when all work items reach a complete state. First, export the update set from the plan dashboard. Then deploy it to a test environment for user acceptance testing, or promote it to production using your standard ServiceNow deployment process.

Important: Changes are isolated until you explicitly deploy and install the application. In ServiceNow Studio, changes are tracked in update sets and promoted when you move the update set between instances.

Deployment methods for Autonomous Engineer

Autonomous Engineer supports update sets and application packaging for moving apps from development to production.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Pack update sets into scoped applications for easier transport and installation across instances, for example using Application Repository (AppRepo). For more information, see ServiceNow application repository.
- For information on update sets for Autonomous Engineer, see Update sets and Autonomous Engineer.
- For more information on System Update Sets, see System update sets.

Autonomous Engineer reference

Reference information for Autonomous Engineer, including work item states, agent packs, and supported capabilities in Build Agent.

This section contains reference information for Autonomous Engineer. Use it to look up work item states, agent pack details, and supported capabilities.

Work item states

For a list of work item states and their definitions, see Autonomous Engineer plan and work item states.

Agent packs

Autonomous Engineer uses agent packs with product-specific domain knowledge to run implementations. For details on available agent packs, see Agent packs for Autonomous Engineer.

Limitations

For information on limitations that apply to Autonomous Engineer and Build Agent, see Build Agent limitations.

Related reference

Additional reference topics for Build Agent:

- Build Agent reference
- Agent packs for Autonomous Engineer
- Build Agent limitations

Domain separation support in Autonomous Engineer

Autonomous Engineer supports domain separation, allowing you to set the domain (`sys_domain`) and use `sys_override` on APIs and records that support it.

Overview of domain separation

Autonomous Engineer supports domain separation on APIs and records that support it on the ServiceNow AI Platform. You can set the domain (`sys_domain`) on those APIs and records, and use `sys_override` where applicable.

Domain separation support is available through the ServiceNow Fluent SDK. When you author application code using ServiceNow Fluent and deploy it to an instance that uses domain separation, the generated artifacts respect the domain configuration of the target instance.

Considerations for domain support

Review the following when developing applications for instances that use domain separation:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Domain separation applies to APIs and record types that support it. Not all APIs or records support domain assignment.
- When you set a domain on a record, it applies to that record within the defined domain scope. Records set to the Global domain are accessible across all domains.
- Use `sys_override` to override domain-controlled values on records where overrides are permitted.

For more general information about domain separation, see Exploring domain separation.

Supported metadata in Autonomous Engineer

Metadata and app file types that Autonomous Engineer can create and manage.

For metadata types not listed in this table, query Autonomous Engineer directly.

Important: Build Agent and Autonomous Engineer only create metadata supported by ServiceNow® Fluent. For more information, see ServiceNow Fluent. For the latest API reference, see https://servicenow.github.io/sdk/

Supported metadata

Supported metadata types

Metadata type	Description
Automated Test Framework (ATF) tests	Automated test cases for validating application behavior.
Application menus and modules	Application menus are the top-level categories in the application navigator sidebar. Modules are the individual links within those categories and can link to tables, URLs, lists, or other platform resources. When Build Agent or Autonomous Engineer creates a new application with tables, it generates the corresponding navigator structure so users can access the application from the sidebar.
Assessments	Assessment instances, metric types, and related configurations for evaluating records against defined criteria. For more information, see Assessments.
Assignment rules	Rules that automatically assign records to users or groups based on field conditions.
Business rules	Server-side scripts that run when records are displayed, inserted, updated, or deleted.
Choice lists	Field value options for select fields on forms and tables.
Client scripts	Scripts that run in the browser to control form behavior.
Client-side logic	Browser-based logic for controlling UI interactions and form behavior.
Condition builder and query conditions	Filter conditions used in business rules, workflows, and list views.
Configuration	System and application configuration settings.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Supported metadata types (continued)

Metadata type	Description
Connection and credential alias	External system connections and stored credentials for authenticating with third-party services and platforms.
Custom AI agents	AI agents scoped to application data models, roles, and ACLs. For more information, see Create agentic workflows, agents, and skills.
Database views	Database view definitions (`sys_db_view`) and database view table configurations (`sys_db_view_table`) that combine fields from multiple tables into a single queryable view.
Data policies	Rules that enforce field-level data requirements, such as mandatory or read-only constraints, across all interfaces including the API.
Data import pipeline	Build Agent creates the full data import pipeline: data sources, staging tables, import sets, and transform maps. Supported source types include CSV, Excel, JSON, XML, JDBC, LDAP, and REST. Specify the external data source, target table, and field mapping, and Build Agent or Autonomous Engineer generates the pipeline artifacts. The capability covers both one-time bulk loads and recurring scheduled imports.
Data lookup	Metadata definitions for data lookup fields that retrieve values from related tables or external sources.
Dictionary overrides	Field-level customizations that override base system dictionary entries without modifying the original.
Event registry configurations	Event definitions that trigger notifications, scripts, or other platform actions.
Field-level overrides	Attribute overrides applied at the field level on extended tables.
Flows	Automated processes built in Workflow Studio using a no-code or low-code interface.
Forms	Record views that define which fields are displayed and how they are arranged.
Inbound email actions	Rules that process incoming email and create or update records based on message content.
JavaScript modules	Build Agent or Autonomous Engineer creates JavaScript modules for organizing reusable server-side code within an application. Modules support standard import and export patterns and can reference server-side APIs. Use this pattern when multiple business rules, script includes, or other server-side scripts must share utility functions, constants, or logic without duplicating code.
Knowledge base access	Configurations that control which users and groups can read, contribute to, or manage knowledge bases and knowledge articles.
LDAP server configurations	Build Agent or Autonomous Engineer creates LDAP server configurations and LDAP server URLs for connecting to external directory services such as Active Directory and OpenLDAP. LDAP server configurations include failover and load-balancing URL configurations and SSL settings. To import records from the directory into ServiceNow tables, use a data import pipeline instead.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Supported metadata types (continued)

Metadata type	Description
List controls	Settings that control list view behavior, including column display and row counts.
Playbooks	Representations of cross-enterprise business processes that organize tasks and activities into logical stages to guide users through a record lifecycle. Playbooks combine triggers that specify when to start, stages that group sequences of activities, and activities that define the automation and user-facing experience. Build Agent or Autonomous Engineer can create playbook configurations and activity definitions to help organizations digitize and standardize their business processes. To generate a playbook in Build Agent or Autonomous Engineer, attach a file, such as an image, an XML file, or a text description. • For details on playbooks support in Build Agent and Autonomous Engineer, see Playbook authoring with Build Agent and Autonomous Engineer. • For more information on playbooks, see Workflow Studio playbooks.
Record insertion	Build Agent or Autonomous Engineer inserts data into any table, including tables that don't have a dedicated build skill. The functionality covers seed data, demo data, reference data, and configuration records. The same approach applies to creating metadata records on platform tables that Build Agent or Autonomous Engineer don't have a specialized skill for yet.
REST message and HTTP method	REST message definitions and HTTP method configurations for building API integrations and outbound integrations.
Scheduled jobs	Automated tasks that run on a defined schedule.
Script includes	Reusable server-side JavaScript libraries callable from other scripts.
Scripted REST APIs	Custom REST endpoints built with server-side scripting.
Security	Access control lists (ACLs), roles, and related security configurations.
Server-side logic	Scripts and business rules that execute on the server.
Service Catalog items and configurations	Catalog items, variables, and fulfillment flows for the Service Catalog. Includes `dependentQuestion` support for variables, enabling dynamic default values that auto-populate based on a referenced variable's record. For example, when a user changes the **Requested for** field, related fields such as **Department** and **Location** can be populated automatically.
Service Portal configuration, pages, and widgets	Portal structure, page definitions, and custom widgets.
Skills	ServiceNow Otto skills for AI-powered responses and actions.
Sys wizard answers	Answer options for platform interceptors stored in the `sys_wizard_answer` table. Interceptors are the choice dialogs that route users to different

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Supported metadata types (continued)

Metadata type	Description
	destinations when they navigate to a form or module. Each answer record defines a label and the target destination for one choice.
Tables	Database tables that store application data, including fields, relationships, and access controls.
Transition conditions	Conditions that determine when a record can move from one state to another in a workflow or flow, enforcing stage-gate logic and data requirements.
UI actions	Buttons, context menu items, and links that trigger scripts or navigation on forms and lists.
UI components	Reusable interface elements built with UI Builder.
UI pages	Custom pages built outside of standard form and list views.
UI styles	Style definitions stored in the `sys_ui_style` table that control the visual appearance of forms and fields.
UI policies	Client-side rules that show, hide, require, or lock fields based on conditions.
UI views	Named configurations of form layouts for different user contexts.
User criteria	Rules and criteria definitions for filtering and selecting users based on field conditions and organizational structures.
Workflows	Automated process sequences built in the legacy workflow editor.
Workspaces	Agent-facing interfaces built with configurable workspace components.

Supported file types for Autonomous Engineer

Autonomous Engineer accepts images, documents, code files, and file types specific to ServiceNow as uploads. Supported formats, size limits, and general guidelines apply when generating apps and metadata.

File types

Supported file types

Category	Extensions
Code and scripts	.js, .ts, .tsx, .jsx, .py, .java, .css, .html, .xml, .json
Documents	.txt, .md, .csv, .log
Images	.png, .jpg, .jpeg, .gif, .svg, .webp
Specific to ServiceNow	.now.ts (Fluent DSL), .xml (update sets)

You can also use MCP server connections, such as for Figma, to provide context for Build Agent. For more information, see MCP connections and Autonomous Engineer.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Size limits

Size limits

Category	Description
Individual file	Up to 10 MB per attachment.
Images	View and analyze screenshots and diagrams directly in the chat.

File upload guidelines

- Upload text-based files to let Build Agent read and analyze the full content, then incorporate it into your app.
- Upload images or screenshots to provide UI mockups, error screenshots, or architecture diagrams as context for development.
- If an attachment is not visible after a window reload, re-upload the file.

File uploads and your project

Uploaded files are available only within the current conversation and aren't added to your ServiceNow project or application automatically.

Upload scenarios

Upload	What Build Agent can do
A script file (.js, .ts)	Read the file, then write the content into your app using the appropriate source file tool.
A screenshot of an error	Analyze the screenshot and help resolve the issue.
A CSV or data file	Use the data to inform table design or generate test data.
A mockup or diagram	Use the file as a reference to build a UI page.
An XML update set	Analyze the file and recreate the metadata using Fluent APIs.

Note:

- Uploading a file does not add it to your app. To incorporate file content into your project, ask Build Agent to do so explicitly, for example, "Add this script to my app as a script include."
- Uploaded files don't persist across sessions. If the window reloads, re-upload the file for Build Agent to access the content again.

Autonomous Engineer plan and work item states

Plan and work item states appear during planning and execution. Identify where a plan or work item is in the execution process and what action is required.

Plan states

The Autonomous Engineer dashboard displays the current state of the plan.

Standby

The plan is not in progress. All work items are in Draft state.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

In progress

Autonomous Engineer is actively working on the plan. Work items are in progress or waiting.

In review

Autonomous Engineer has completed all work. Work items are pending your review.

Completed

The plan has been reviewed and completed and is ready for deployment.

Cancelled

The plan was cancelled and will not be executed.

Work item states

Each work item moves through states during plan execution. The plan dashboard displays the current state of every work item.

Draft

The work item was generated during planning and has not yet been queued for execution.

Ready

The work item has been reviewed and approved and is waiting for its dependencies to complete before it can be queued.

Queued

The work item is in the execution queue and waits to be picked up by a background agent.

Waiting

The work item has unmet dependencies and can't begin execution until those dependencies are complete.

In progress

A background agent is actively building the work item.

Needs validation

The work item is built and Automated Test Framework (ATF) tests have run. The work item requires your review before it can be marked complete.

Complete

You have reviewed and approved the work item. Its artifacts are included in the plan update set.

Blocked (user input)

The background agent requires information from you before it can continue. A notification appears in the chat panel.

Blocked (error)

The background agent encountered an error it can't self-heal. Open the work item or ask Build Agent in the chat panel for details, then ask Build Agent to retry.

Stopped

The background agent encountered an error and stopped execution. Autonomous Engineer retries the work item automatically before surfacing it for manual intervention. If the work item can't be resolved after the automatic retry limit is reached, it appears in the plan dashboard and the chat panel for your review.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Cancelled

The work item was cancelled and will not be executed. Cancelled work items aren't included in the plan update set.

Build Agent

Use Build Agent, an autonomous AI agent, for creating and updating ServiceNow® applications. You can also ask Build Agent general ServiceNow development questions.

Get started

Explore  Learn about Build Agent.	Configure  Install and configure Build Agent to use its features.
Use  Use Build Agent to autonomously create and update ServiceNow applications.	Reference  Build Agent reference material, such as example prompts.

Important:

- Not all model providers are available for customers with in-country SKUs, and some AI products/features are currently unavailable for in-country customers. For more information, see the KB1584492 article in the Now Support Knowledge Base. Be sure to check for model provider availability updates in future releases.
- Some AI products/features are currently unavailable for customers in the FedRAMP, NSC DOD IL5, or Australia IRAP-Protected data centers, self-hosted customers, or in other restricted environments. For more information, see the KB0743854 article in the Now Support Knowledge Base. Be sure to check for availability updates in future releases.
- Some AI products/features are currently available only for customers in some regions. Be sure to check for availability updates in future releases.
- Some AI products and skills are not available in Regulated Markets. For more information, see KB2593939: Regulated Markets AI Products/Skills Not Available. Be sure to check for availability updates in future releases.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Key benefits of Build Agent

- Built-in governance to develop securely
- Faster enhancement of existing applications, such as adding new features
- Reduced development backlogs and streamlined deployment of new applications
- Better documentation of existing applications through automated documentation
- Increased developer productivity across skill levels
- Reduction in overall development cost
- Automated testing with Automated Test Framework (ATF) test suite generation and execution

Note: Currently, only admins have permissions to use Build Agent.

Data collection

ServiceNow collects and uses the inputs, outputs, and edits to outputs of this application to develop and improve ServiceNow technologies including ServiceNow models and AI products. Customers can opt out of future data collection at any time, as described in the AI Opt-Out page.

Exploring Build Agent

Build Agent enables developers to create, edit, and deploy full-stack ServiceNow® applications to update sets that encompass both user interface and back-end components.

ServiceNow Otto is the new AI experience brand. This change is reflected in the name of ServiceNow products, including Build Agent. Your product entitlements remain unchanged. Check your entitlements to determine your access to specific features.

Build Agent overview

Build Agent is an AI tool designed for developers within ServiceNow Studio and the ServiceNow Integrated Development Environment (ServiceNow IDE). Build Agent acts as an autonomous AI agent capable of independently generating a complete ServiceNow® scoped or global application. For example, a global scope app that uses tables such as incident, problem, and change.

Migrating to Build Agent from App Engine Studio

If you currently build applications in App Engine Studio (AES), you can migrate to Build Agent when your development needs grow beyond what AES supports. For example, when you need custom scripting, global scope applications, or the ability to modify base system applications. Existing app artifacts like tables, flows, workspaces, and ACLs are already accessible in ServiceNow Studio, so migration is primarily a workflow adjustment rather than a conversion process. For more information, see Migrating to Build Agent.

Conversational interface

Using the chat panel within ServiceNow Studio or the ServiceNow IDE, you can interact with Build Agent through an easy-to-use multi-turn conversation interface. You can also ask it general questions about developing on the ServiceNow AI Platform.

All you have to do is describe an application in natural language, and the agent can then automatically create it. Build Agent generates the necessary code, organizes files clearly, and manages both the core logic and user interface components of the application.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Build Agent can understand natural language prompts, autonomously generate full-stack applications, oversee the entire build process, respond to feedback, deploy applications to update sets, and more.

Build Agent is enabled by default to create apps with AI, for example in ServiceNow Studio. To use other ServiceNow Otto products, such as the app generation skill, disable Build Agent. For example, using the setting in your ServiceNow Studio preferences.

Note: Depending on your license, you will have access to certain application features, generative AI skills, agentic workflows, and AI agents. For more information, see ServiceNow product tiers.

Where to use Build Agent

You can use Build Agent in ServiceNow Studio, including the ServiceNow IDE. For detailed instructions on using Build Agent in ServiceNow Studio, see Accessing Build Agent in ServiceNow Studio and Build Agent in ServiceNow Studio.

You can also use Build Agent in Developer Sandboxes, which provides a controlled baseline configuration for isolated, parallel development. For more information, see Exploring Developer Sandboxes.

Providing context with file uploads

You can upload supported file types, such as images, code, and documents, to provide more context about application design and functionality. For more information, see Supported file types for Build Agent.

Extended capabilities

While creating and updating applications is the primary use case for Build Agent, its capabilities extend beyond that. It can perform various code-related tasks, such as rewriting tables, explaining code, validating and enhancing existing applications, fixing application errors, and more. For instance, Build Agent can use the Run Query tool to query a specific table within your instance and return the top five records or derive specific insights.

To use Build Agent as part of an end-to-end development practice that includes Git, the ServiceNow SDK, ReleaseOps, and CI/CD pipelines, see the SDLC on ServiceNow guide in the ServiceNow SDK documentation.

Autonomous Engineer

For large or complex implementations, you can use Autonomous Engineer to go from requirements to a working implementation without building each artifact manually. Autonomous Engineer conducts an interview, generates a plan with work items, and builds all work items in parallel in the background. For more information, see Autonomous Engineer.

Playbook Designer support

You can use Build Agent to author Playbook Designer artifacts, including runtime permissions, activity definitions, and Agentic activity field configuration. Playbook records are consolidated into a single XML update set file for consistent deployment across instances. For more information, see Exploring Playbook.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ServiceNow Fluent and modern web frameworks

Build Agent generates code in ServiceNow Fluent, the domain-specific language for developing on the ServiceNow AI Platform. Build Agent can also incorporate modern web frameworks, such as React, when building custom user experiences.

Important: Build Agent and Autonomous Engineer only create metadata supported by ServiceNow® Fluent. For more information, see ServiceNow Fluent. For the latest API reference, see https://servicenow.github.io/sdk/

Build Agent (Trial) app overview

Build Agent is available as a trial app on a freemium model, and is automatically installed on all instances.

After you install the Build Agent (Trial) app, your instance receives 100 free user interactions for 30 days at no additional charge. The free interactions enable you to explore Build Agent features at no additional cost.

If you exceed the free interaction limit, you must wait 30 days for a reset, or install the paid version of Build Agent.

Automatic upgrades

Qualifying instances receive automatic upgrades when a new version of Build Agent is published to the ServiceNow Store. For more information, see Install Build Agent.

Localization and Build Agent

Build Agent incorporates the ServiceNow AI Platform localization, so you can use it in any supported language. For more information, see Localization Framework.

How prompts are counted

Prompts are counted each time you submit a message to Build Agent. If Build Agent asks a clarifying question and you respond, that response counts as a prompt. Approving a plan that Build Agent presents does not count as a prompt. To get the most value from each prompt, draft your message in a text editor before you submit it.

For more information on prompting, see Example prompts.

What apps are in ServiceNow

A ServiceNow app is a package that performs a specific task for a specified group of users. Think of an app as a container with a set of rules around who can access and edit it. For example, ServiceNow apps can include an API, a table, a workspace, a form, a flow, or any combination of those things.

Enhance applications with agentic experiences

You can enhance your applications with agentic experiences by using Build Agent.

Build Agent enables you to create agentic workflows, agents, and skills for your custom applications. Additionally, Build Agent recommends in-app agents tailored to specific use cases.

For more information, see Create agentic workflows, agents, and skills.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Build Agent benefits

Build Agent accelerates application development on the ServiceNow AI Platform. It helps automate many repetitive and time-consuming tasks that developers previously had to do manually, enabling:

- Increased developer productivity
- Reduced development backlogs
- Quicker deployment of new business applications
- Reduction in overall development cost

What to explore next

To learn more about configuring and using Build Agent, see:

- Build Agent configuration
- Use Build Agent
- Build Agent reference
- Support for regulated markets

To learn more about prompting, see this Community article on The fastest way to learn Build Agent prompting? Ask Build Agent.

Build Agent use cases

Use Build Agent for a wide range of development scenarios beyond application creation, including app analysis, modernization, documentation, governance, and learning assistance.

Documentation creation

Use Build Agent to create customized knowledge articles, so users can access information and learn the application more easily.

Build Agent can automate the creation of readme files for source code repositories. Generated readme files provide clear, structured documentation that outlines installation procedures, usage guidelines, and example snippets, which reduces onboarding time for new developers.

Learning assistance

Use Build Agent as a resource for learning ServiceNow application development practices. You can ask about many topics, including development techniques, available APIs, and the overall functionality of the ServiceNow AI Platform.

Build Agent supports continuous learning by summarizing relevant documentation, highlighting key points, and providing concrete examples that illustrate complex concepts. These examples help developers apply general guidelines in their work.

For guidance on structuring a complete development lifecycle with Build Agent, including source control, testing, and release management, see the SDLC on ServiceNow guide in the ServiceNow SDK documentation.

Brainstorming sessions

Use Build Agent to refine application development ideas and capture structured requirements for later build steps during brainstorming sessions. Requirements are captured during these sessions, supporting a structured process for developing application features. Ideas are

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

documented so your team can revisit previous discussions and integrate feedback during the development lifecycle.

Additional use cases

Use the following scenarios to identify specific ways to apply Build Agent in your development workflow.

Specification-driven implementation with Autonomous Engineer

Provide requirements as a prompt or a file upload. Build Agent interviews you to resolve ambiguity, queries your instance for existing artifacts, and generates a plan with work items in an Agile user story format. After you approve the plan, Build Agent builds all work items in parallel in the background, generates and runs Automated Test Framework (ATF) tests, and produces an update set on completion. For more information, see Autonomous Engineer in Build Agent.

Playbook authoring

Use Build Agent to author Playbook Designer artifacts through a conversation. You can have Build Agent do the following for playbooks that it generates:

- Generate runtime permissions at the playbook and stage levels
- Define on-demand launcher configurations
- Create optional activities scoped globally or to a specific stage
- Configure Set Playbook Outputs activity definitions for nested playbooks
- Set Agentic activity fields on form-based and record-based activities

For details on using playbooks with Build Agent, see Supported metadata in Build Agent. For more information on using playbooks, see Exploring Playbook.

App gap analysis

Summarize an existing app, identify gaps in coverage, and propose targeted enhancements. For example, Build Agent identifies tables without ACLs, business rules with no test coverage, and manual steps that could be automated with flows.

Security posture review

Open an existing app and audit its ACLs, roles, and data filters. Build Agent identifies gaps such as tables with no ACLs, overly permissive rules, and missing role hierarchy, then generates fixes.

Test coverage bootstrapping

Analyze an app's metadata, identify what has zero test coverage, generate Automated Test Framework (ATF) tests for the gaps, run them, and triage failures.

Architecture diagrams

Generate Mermaid.js diagrams by querying and collating data from the platform.

Common workflow patterns

Describe a common business process, such as intake and approval, escalation routing, or SLA monitoring with dashboards. Build Agent then scaffolds the full workflow, including tables, flows, catalog items, notifications, and reporting.

Migration acceleration

Provide your source system's table definitions, business logic, or data model pasted as SQL, text, CSV, or screenshots. Build Agent then recreates the equivalent tables, fields, relationships, and automation on the ServiceNow AI Platform.

App modernization

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Analyze legacy customizations such as classic workflows, Jelly UI, and script-heavy logic, then rebuild them using modern patterns: flows instead of workflows, React UI pages instead of Jelly, and structured security instead of script-based checks.

Data-model scaffolding

Paste a CSV file, drop a spreadsheet screenshot into the chat, or describe your data model in plain text. Build Agent generates the matching ServiceNow tables, columns, and relationships. For Excel or PDF input, convert to CSV or paste a screenshot first.

Cross-scope workflows

Build custom apps that write to global tables such as Incident. For example, auto-generate incidents when assets become overdue.

Build from user stories

Point Build Agent at a user story from the rm_story [rm_story] table or paste one from another tool. The acceptance criteria drive development, creating the tables, logic, and tests that satisfy each criterion.

> **Note:** Querying the rm_story [rm_story] table directly requires the Agile Development 2.0 plugin.

SQL to table

Paste a SQL `CREATE TABLE` definition into the chat and Build Agent translates it into a ServiceNow table with matching column types and relationships.

Post-build summaries

Generate release notes and summary documents after the build foundation is complete.

Instance governance setup

Create instance scan checks that enforce your organization's standards, including naming conventions, security requirements, and code quality. Checks run as part of your existing instance scan infrastructure.

Use cases by development goal

Use the following table to identify Build Agent use cases based on your development stage and goal.

Build Agent use cases by development stage and goal

Use case	Immediate productivity	Process and quality improvement	Strategic transformation
Explore (zero risk, read-only)	• Explain any concept, API, or metadata type in plain language	• Document an existing app end-to-end: architecture, security model, and data flows	• Run existing Automated Test Framework (ATF) tests and triage failures to assess application health before a release

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Build Agent use cases by development stage and goal (continued)

Use case	Immediate productivity	Process and quality improvement	Strategic transformation
	• Look up how a specific business rule, flow, or ACL works	• Generate Mermaid diagrams of table relationships and integration points	• Audit security posture: review ACLs, identify missing role controls, and surface compliance gaps
Enhance (modifications, reviewable before promotion)	• Add a field, client script, or UI policy to a form • Add a catalog variable or update a record producer	• Add email notifications, scheduled jobs, and flows to automate a manual process • Build a scripted REST API for an external integration	• Generate ATF test suites for untested apps, then run and triage them to establish a quality baseline • Add instance scan checks to enforce governance standards across the platform
Implement (specification-driven, parallel execution)	Provide requirements as a prompt or file and let Autonomous Engineer generate and build a full plan	• Review, edit, and approve work items before execution to align the plan with your standards • Use agent packs to implement custom applications on your instance.	• Compress multi-month implementation timelines to hours by executing all work items in parallel with automated ATF test generation and self-healing • Export the completed update set and deploy to UAT or production
Build (new development on non-production)	• Prototype an app from a one-line description • Create a catalog item with variables and a fulfillment flow	• Build a full app with tables, roles, ACLs, business rules, flows, catalog items, and a UI page • Build a Service Portal or workspace for self-service	• Build AI agents and ServiceNow Otto skills to bring intelligent automation into your processes • Build from design mockups or a detailed PRD, complete with automated test coverage from day one

Autonomous Engineer in Build Agent

Autonomous Engineer is an agentic worker that augments product development capacity across the SDLC. It intakes requirements, applies platform best practices through agent packs, and generates an implementation plan before building work items in parallel through Build Agent.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Implementing applications on the ServiceNow AI Platform can require significant manual effort between defining requirements and delivering value to users. Autonomous Engineer reduces that time to value by taking your specifications as input and driving the full implementation lifecycle end to end.

Note: You must install the Autonomous Engineer app from the ServiceNow Store. Check your entitlements to see whether you can use Autonomous Engineer.

Autonomous Engineer uses agent packs to give background agents product-specific domain knowledge for application and product implementation. For more information, see Agent packs for Autonomous Engineer.

Autonomous Engineer is available in ServiceNow Studio and covers custom applications.

For full documentation on Autonomous Engineer, including how it works, how to configure it, and step-by-step instructions for starting and managing a plan, see Autonomous Engineer.

To start using Autonomous Engineer in Build Agent, see Using Autonomous Engineer.

Build Agent workflow

The Build Agent workflow automates building applications, testing, and deploying update sets on the ServiceNow AI Platform. Build Agent streamlines development by handling code compilation, quality checks, and deployment steps without manual intervention.

Note: For large or complex implementations, you can use Autonomous Engineer instead of the conversational workflow that follows. Autonomous Engineer accepts your requirements up front, interviews you to resolve ambiguity, generates a plan with work items, and builds all work items in parallel in the background. For more information, see Autonomous Engineer in Build Agent.

A general workflow for using Build Agent in either ServiceNow Studio or the ServiceNow IDE is the following:

1. Make sure that everything you need is properly configured in the settings, such as supported MCP server connections. For more information, see Build Agent configuration.
2. Open ServiceNow Studio. Use the central chat area on the home page to start a new conversation, or select the Conversations icon in the Navigator panel to open an existing conversation.
3. Describe what to create or change in natural language.
4. Let Build Agent parse requirements and propose the application and files to create or modify.
5. Build Agent edits code or metadata or scaffolds a new application.
6. Review proposed edits, diffs, and summaries, and approve or adjust before applying changes. Review checkpoints and manual edit update sets. In ServiceNow Studio, view generated app details from the **Apps** tab, and inspect the source code from the **Explorer** tab.
7. Iterate until the desired metadata changes are complete. For more information, see Supported metadata in Build Agent.
8. Prompt Build Agent to create and run Automated Test Framework (ATF) tests to verify that the tests execute as expected. Depending on your configuration, Build Agent may ask you if you want to run ATF tests. If there are failures, auto troubleshooting triages the tests and produces a regression test suite that you can use to monitor app health.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

9. Instruct Build Agent to build the application; verify results in the File Navigator or Metadata Explorer.
10. Deploy the application. If you're using source control, you can push to Git. For more information, see Deploying what you built with Build Agent.

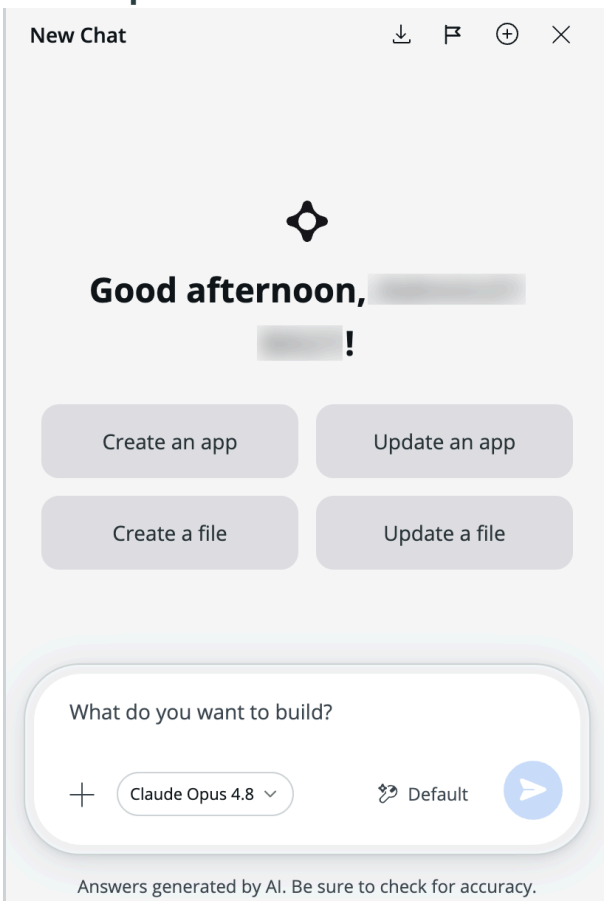
When your app is connected to Git-based source control, you can also manage source control operations from the Build Agent chat panel without switching to the VS Code UI. For more information, see #unique_89.

Build Agent and Autonomous Engineer chat panel

The Build Agent chat panel is where you interact with the AI agent during development. Use it to submit requests, review responses, and apply generated code.

Use the Build Agent or Autonomous Engineer chat panel to create or update an app or app file. Make a selection to begin the chat, or enter a prompt. If you have a longer prompt, the chat box expands as you extend your input. The chat panel runs in Build Agent mode by default. To use Autonomous Engineer, select **Build Agent** from the selector in the chat panel and choose Autonomous Engineer.

New chat panel



Continue your conversation in the chat panel until you're happy with the results.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Navigating the Build Agent chat panel

Function	Description
New chat icon	Open a new chat in the Build Agent chat panel. Begin a new chat when you want to start working on a new application or need a fresh start for updates.
Chats icon	See a list of all your chats with Build Agent.
Checkpoints icon	See a list of all the checkpoints within your current chat with Build Agent. Checkpoints show all the progress points in your application. You can revert to any of these checkpoints during the course of developing your app.

Supported models and versions

Learn which AI models and versions Build Agent supports and how to change them. Use this information to verify compatibility and select the right model for your task.

Supported models

Build Agent currently supports the following models:

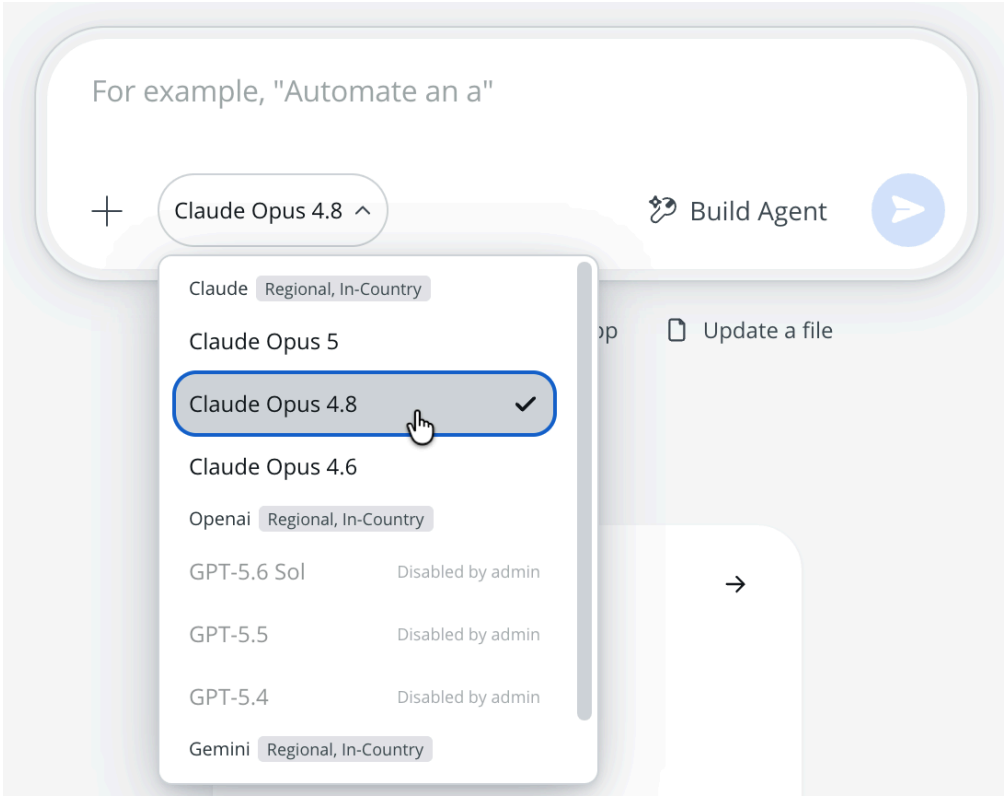
- Azure OpenAI 5.4
- Azure OpenAI 5.5
- Azure OpenAI GPT 5.6 Sol
- Google Gemini 3.7 Flash
- Google Gemini 3.5 Flash
- Gemini 2.5 Pro
- Claude Opus 4.6
- Claude Opus 4.8
- Claude Opus 5.0

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Changing the model and version

You can change the version by selecting it directly in Build Agent or Autonomous Engineer.

Change model version



To change the model in use, navigate to the AI Admin Hub. For information on changing the model that Build Agent or Autonomous Engineer uses, see Manage model providers.

Tutorial for Build Agent in ServiceNow Studio

Learn to develop reusable server-side logic and build a ServiceNow® application in Build Agent, from data modeling through testing, using agentic development.

Tutorial information and link

The tutorial covers the following items:

- Application creation, custom data modeling, and table design
- Server-side business logic with Script Includes
- Validation and testing using background scripts
- Reusable utility patterns for real-world use cases

The full lab is available on GitBook (you may be required to log in).

Get started with prompting

To get started with Build Agent, try one of the following approaches.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Prototype a simple app: Describe a straightforward application in one or two sentences. Review the plan that Build Agent presents, then approve it to start the build. This approach lets you learn the workflow with low risk.
- Document an existing app: Open a custom scoped app and ask Build Agent to summarize its architecture, data model, roles, and key components. This produces useful documentation with no risk to the app.
- Extend an existing app or base system table: Add a field, a business rule, or an ACL to an app you're familiar with. Build Agent supports global-scope tables such as Incident, as well as custom and ServiceNow Store apps.

General guidelines for Build Agent

Use these guidelines to get the most out of Build Agent in your development workflow.

Development guidelines

To maximize Build Agent effectiveness, use the following practices:

- Design before coding: Think through and document the requirements for your application across the data and UI layers, for example using Workflow Studio or Figma.
- Start with a clear plan: Collaborate with Build Agent to define scope, required tables, and metadata types.
- Instruct with context: Write instructions for what you want to achieve with your application using Markdown in your file system. You can then ask Build Agent to use the file as context for its work.
- Use specific terminology: Treat Build Agent as your development partner. Provide specific, clear instructions using ServiceNow platform terminology such as table names, field names, roles, and artifact types.
- Test early and often: Add sample records, test on the instance, and run ATF tests throughout development.
- Use version control: Use Git for tracking changes and maintaining a clean workspace structure.
- Provide visual context: Give screenshots to Build Agent to troubleshoot UI issues and request changes.
- Extend with third-party libraries: Integrate third-party Node Package Manager (NPM) libraries, such as React JS, for enhanced interfaces.
- Maintain documentation: Keep clear rules and documentation in project folders; use Build Agent to create knowledge base articles.
- Iterate and experiment: Prototype freely, then ask Build Agent for an ideal prompt to recreate the result once your vision is complete.

Prompting guidelines

Use the following guidelines to get better results from Build Agent.

Use Autonomous Engineer for large implementations

For complex apps or ServiceNow product implementations, use Autonomous Engineer. Autonomous Engineer interviews you, generates a plan with work items, and builds all work items in parallel in the background. Review and approve the plan before Build Agent starts building. For more information, see Autonomous Engineer in Build Agent.

Ask Build Agent to prompt for you

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Ask Build Agent to suggest an ideal prompt for your goal or coach you on common pitfalls for a given metadata type. To learn more, see this Community article on The fastest way to learn Build Agent prompting? Ask Build Agent.

Be comprehensive

One detailed prompt can provide better results than several vague ones. Include tables, fields, roles, and automation in a single prompt.

Let Build Agent interview you

For broad ideas, let Build Agent ask clarifying questions before planning. In Autonomous Engineer, this interview is built into the workflow. In standard chat, you can prompt Build Agent to ask questions before it starts building. Both approaches produce better results than trying to anticipate every detail yourself.

Use specific ServiceNow terminology

Say `reference to sys_user` instead of `add a field for who ordered`. Naming the table, field, scope, role, or artifact reduces interview cycles and produces more accurate output on the first attempt. Conversational prompts work well for early exploration, while precision works better as the build progresses.

Draft offline

Write prompts in a text editor, refine them, then paste into the chat, which saves prompts and can help to produce better results.

Develop incrementally

Develop each page or feature in separate chats. Implement and test, then start a new conversation for the next feature.

Use markdown

Place a markdown file (for example, named `GUIDE_*.md` or `BUILD_AGENT_RULES.md`) in your project directory. Build Agent reads these grounding files and follows your conventions throughout the session.

Note: Markdown for Build Agent is currently available only in the ServiceNow IDE. Teams that get the most value from Build Agent markdown files document their organizational standards in these files, including the following:

- Naming conventions: table names, field names, business rule naming patterns, and scope prefixes
- Technical standards: coding practices, preferred patterns, and anti-patterns to avoid
- Development cadence: processes, update set management, and code review requirements
- Last-resort actions that require explicit approval: creating new tables, changing ACLs or roles, modifying system properties, and altering base system business rules

Use custom skills and rules

Define instance-level, application-level, and user-level instructions that Build Agent follows in every session. Use rules to enforce consistent behavior automatically, such as naming conventions or required fields. Use skills to provide on-demand guidance for specific tasks. Unlike markdown grounding files, custom skills and rules require no project files and apply across all sessions at the configured scope. For more information, see Configure custom skills and rules.

Use ESLint

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Use ESLint on the ServiceNow AI Platform to define your preferred coding style, and ask Build Agent to manage and enforce the ESLint configuration, for example using snake case for variables.

Build Agent tools

Build Agent tools support application development tasks such as semantic search, schema inspection, code search, planning, UI validation, database querying, app navigation, and script execution. Each tool extends what Build Agent can do during a build session.

Code search tool

The code search tool searches workspace files by exact text or regex patterns. Use the code search tool to find references to a field, script, or function across your app, and to understand the impact of a change before making it.

Open app tool

For applications that were not developed using the ServiceNow IDE, ServiceNow Studio, or the ServiceNow SDK, you must convert them into Fluent format to enable development within the Build Agent. You can prompt the Build Agent to use the open app tool to locate the application you want. Alternatively, you can search for an application directly within the Build Agent, and it will automatically use the open app tool. The open app tool can find an application, convert it to Fluent format, and then add the converted app to your workspace.

Planning tool

Build Agent and Autonomous Engineer include a planning tool that creates a detailed, step-by-step plan for your application development. You can refine the plan iteratively by prompting for changes and providing feedback until you reach a final version.

Rollback script tool

The rollback script tool reverts a previous run script tool operation on demand. After Build Agent or Autonomous Engineer runs a script, it returns a rollback context you can use to undo that run. Use the rollback script tool when a script run produces an unexpected result and you want to restore your instance to its previous state.

Only scripts run by Build Agent or Autonomous Engineer in the current session can be reverted using this tool. For details on security and governance around the rollback script tool, see Build Agent governance.

Run query tool

Build Agent and Autonomous Engineer can use the run query tool to query a specific table within your instance and return the top five records or derive specific insights.

Run script tool

The run script tool generates and runs a server-side script on your instance during a build session. Use the run script tool when you need Build Agent or Autonomous Engineer to perform an operation that goes beyond querying a table. For example, to retrieve data that requires custom logic, or to create something dynamically when no other tool covers the task.

Before the script runs, Build Agent or Autonomous Engineer shows you the generated script, its stated intent, and the application scope it will run in. You must approve the operation before it executes. The script runs in the scope of the app you're building by default. For details on security and governance around the run script tool, see Build Agent governance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Semantic metadata search tool

The semantic metadata search tool is part of the Build Agent and Autonomous Engineer tool registry. Semantic search enables you to describe what you're looking for without providing the exact file name or keyword. For example, you can search for `files related to incident process management`, and the tool finds relevant results based on semantic understanding, not just keyword matching. Indexing keeps results current as your instance evolves.

Build Agent uses semantic search in two ways:

Implicit search

Build Agent and Autonomous Engineer determine when to use the semantic search tool based on your development task. When you describe what you want to accomplish, Build Agent or Autonomous Engineer searches for existing files and applications that match your intent. It then presents the results in a plan for your approval, which helps prevent duplicate content creation.

Explicit search

You can directly request a search by asking Build Agent or Autonomous Engineer to find something on your instance. Use explicit search when you want to locate specific files, applications, or knowledge without making changes.

Table schema inspector tool

The table schema inspector tool returns every field on a table, including field names, types, mandatory flags, and reference targets. Use the table schema inspector tool to understand a table's structure before adding fields or writing logic against it, and to verify the schema of base system tables.

UI validation tool

Build Agent and Autonomous Engineer detect UI components created during the build process, then analyzes and validates the UI using the UI validation tool. The tool automatically uses skills available in the Playwright MCP and executes them on the Automated Test Framework (ATF) Cloud Runner infrastructure.

You can specifically prompt UI validation by asking Build Agent or Autonomous Engineer to `validate the UI built using the ATF cloud runner tooling`. The UI validation tool is available for Build Agent in both ServiceNow Studio and the ServiceNow IDE.

Note: The ATF test generator and Cloud Runner app must be installed on the instance to use the UI validation tool.

Web search tool

Two Build Agent and Autonomous Engineer tools support external search.

- Use the web search tool to find information on the public web.
- Use the web fetch tool to retrieve content from a URL that you supply.

Note: The web search tool must be enabled in the Build Agent or Autonomous Engineer settings. For more information, see Configure web search.

Playbook authoring with Build Agent and Autonomous Engineer

Use Build Agent and Autonomous Engineer to author and manage Playbook Designer artifacts through a conversation. You can generate playbook structures, configure activities, set runtime

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

permissions, and define launcher configurations without manually navigating the Playbook Designer UI.

Build Agent and Autonomous Engineer can generate and configure Playbook Designer artifacts from a conversation. Describe the playbook structure, activity logic, or configuration you need, and Build Agent or Autonomous Engineer produces the corresponding records and settings on your instance.

For more information on playbooks, see Exploring Playbook.

Fluent-based playbook authoring

You can build, edit, and manage Platform Playbooks using the Fluent TypeScript DSL directly in the ServiceNow IDE or a local SDK. Playbook components such as stages, triggers, and roles are defined as modular `.ts` files that support Git-based version control and team collaboration.

Existing playbooks can be converted to Fluent format. During conversion, metadata is transformed into editable and deployable `.ts` files. Use `now-sdk fetch` and `now-sdk deploy` commands to sync playbook changes bidirectionally between a local IDE and a ServiceNow instance.

All playbook records related to a playbook definition are consolidated into a single XML update set file, which simplifies version control and deployment.

What you can do from chat

From the Build Agent or Autonomous Engineer chat panel, you can:

- Generate playbook structures including stages, activities, and conditions
- Generate runtime permissions at the playbook level and at the stage level
- Generate and configure the Set Playbook Outputs activity for nested playbooks
- Configure agentic fields on form-based and record-based activities when the AI Agent plugin is active
- Generate on-demand playbook launcher configurations
- Configure golden path settings and define the ideal path through decision nodes.
- Author public playbooks and playbook variants
- Configure Go back to activity definitions
- Configure golden path settings
- Use Automation plan pills in playbook activities
- Generate playbook scaffolding from an uploaded image

Considerations for playbooks

- Agentic activity field configuration requires the AI Agent plugin to be active on your instance.
- Fluent-based playbook authoring requires ServiceNow IDE or the ServiceNow SDK.
- For supported playbook metadata types, see Supported metadata in Build Agent.

MCP connections and Build Agent

MCP connections enable Build Agent to access external tools and resources through standardized communication. Use these connections to integrate third-party applications like Figma for accelerated design-to-development workflows.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The following MCP connections are currently supported for Build Agent:

- Atlassian Rovo
- AWS DevOps
- Box
- Docusign
- Figma
- Linear
- Miro
- Prisma Postgres
- Zoom
- Zoom Chat
- Zoom Docs
- Zoom Whiteboard

Note: Because Build Agent doesn't support multiple servers connecting to one app, each Zoom integration requires its own app in Zoom.

MCP is an open protocol that defines how AI agents communicate with external systems to discover and invoke tools, retrieve resources, and share context. By standardizing this communication layer, MCP enables AI agents to interact with a wide range of third-party applications without requiring custom integration logic for each one.

In Connect Hub, an MCP connector represents a configured connection between ServiceNow and an external system that exposes a server compatible with MCP. After you set up an MCP connector, AI agents can use it to access tools and capabilities from the external system, enabling coordinated workflows across models and systems.

Configuration details

For details on configuring MCP connections in Build Agent, see Connect Build Agent to a supported MCP server. For more information on MCP connections, see Additional connector configurations.

Connect to supported MCP servers

Build Agent can use any MCP connection that's available in an instance's registry, though you must manually authenticate the connection the first time you use it with Build Agent.

For details on adding a new MCP connection in Workflow Data Fabric, see Model Context Protocol connectors.

An example prompt: `Let's create a new project to track my baby's sleep and create issues in Linear based on inputs from the tracker.`

Note: MCP connections are available in both ServiceNow Studio and the ServiceNow IDE.

Approve and activate MCP servers

Before you can enable an MCP server in Build Agent, an administrator must approve it as an AI asset in AI Control Tower. Each MCP server requires this approval, regardless of whether the server is enabled by default.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The end-to-end flow for making an MCP server available is:

1. The administrator adds the MCP server as a Workflow Data Fabric (WDF) connection.
2. The administrator approves the server as an AI asset in AI Control Tower.
3. You authenticate the connection in Personal Integrations.
4. You enable the MCP server in Build Agent settings.

For details on enabling MCP connections, see Client registration using custom connector.

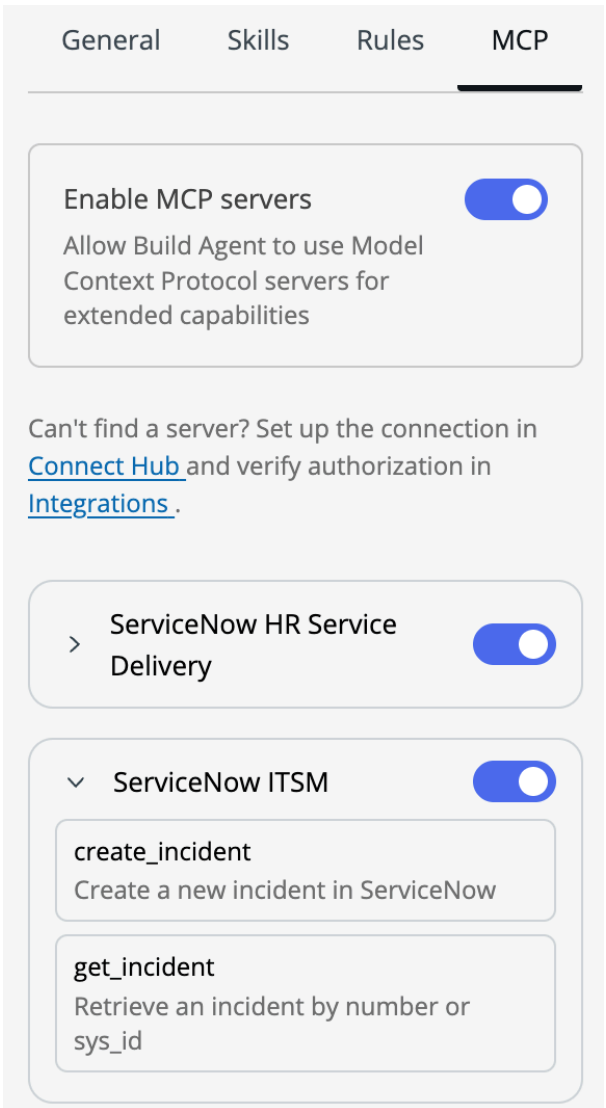
Individual MCP servers are enabled by default, but the complete flow must be completed before any server is available for use.

For details on adding a new MCP connection in Workflow Data Fabric, see Model Context Protocol connectors.

View available tools per MCP server

You can view the list of tools available for each connected MCP server in the **MCP servers** pane in the Build Agent settings panel. Expand a server by selecting the chevron next to the server name to see its tools and descriptions.

MCP tools list



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Application development with Figma MCP server

Connect the Figma MCP server to the Build Agent to accelerate the transition of your Figma designs to enterprise-grade applications on the ServiceNow AI Platform®.

Note: You can upload additional types of files to accelerate development, such as images and code files. For more information, see Supported file types for Build Agent.

The Figma MCP (Model Context Protocol) server establishes a secure connection between the Build Agent and Figma using OAuth for login authentication. This server enables the Build Agent to access structured design data directly from Figma files, rather than relying on static screen captures. With the available data, the Build Agent can effectively create applications.

The Figma MCP server translates Figma data, such as components, styles, and variables, into a machine-readable format. The Build Agent uses the translated data to generate code, gather design context, or develop design systems directly from Figma. Developers can create more accurate and consistent application UIs faster than with traditional approaches.

The process of converting Figma designs to applications is as follows:

- Designers use Figma to create user interface (UI) components, styles, and variables.
- The Figma MCP server uses OAuth to securely access design data from Figma.
- The Figma MCP server acts as an intermediary, converting Figma design data into the Build Agent context.
- The Build Agent, which is installed on your ServiceNow® instance, receives the translated data and uses it to generate applications.
- The Build Agent creates applications based on the design data provided by Figma.

Note: An allowlist process is required to connect the Build Agent MCP client to the Figma MCP server. Contact Now Support to initiate the process.

Build Agent governance

Governance controls in Build Agent help with code quality, security, and compliance when generating applications. The Build Agent automated safeguards prevent common development issues and enforce organizational standards.

Every app and AI agent generated with Build Agent inherits the governance of AI Control Tower and App Engine Management Center, plus the same identity framework as the rest of the ServiceNow AI Platform®. Use the governance tools to monitor agents, manage app lifecycle, and enforce policy.

Governance addresses the following:

- Risk and compliance: AI-generated apps meet enterprise security standards and regulatory requirements.
- Quality assurance: Automated code is validated through testing and review.
- Visibility and control: Prevents shadow IT and enforces lifecycle transparency.

Build Agent automatically generates Access Control Lists (ACLs) that enforce role-based access, validates scripts for security vulnerabilities, and applies code optimization during generation. Every app that's developed agentically on the ServiceNow AI Platform includes audit trails, security controls, and compliance checks without requiring explicit prompts for these features.

Build Agent supports the full set of native ServiceNow security controls, not only ACLs and roles. For example, Build Agent can create Security Attributes for attribute-based access control and Security Data Filters for row-level data restriction.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Security Attributes control access based on properties assigned to users and resources, instead of role membership alone. For more information on Security Attributes, see Security Attributes.
- Security Data Filters restrict which rows a user can see on a table, for example, so that managers see only their team's records.

All four security metadata types can be combined in a single application for layered access control.

Some ways that Build Agent enforces governance include:

- Enforce ACLs and role-based access for generated apps, which Build Agent can do.
- Validate AI-generated scripts for security vulnerabilities.
- Apply code optimization and review before publishing.
- Create Cross-Scope Privileges to control which tables, scripts, and resources one scoped application can access from another. Use Cross-Scope Privileges to diagnose and resolve `operation not allowed` errors between scoped apps.
- While creating agents and skills, Build Agent asks which users and roles it should operate as, as well as which users are allowed to access the agents or skills.

AI Control Tower

Agents generated by Build Agent are registered as AI assets in AI Control Tower, where AI stewards can track lifecycle progression, monitor security posture, and assess compliance. From the AI asset record, stewards can review governance health, evaluation scores, and risk classification for each agent without leaving the workspace.

AI Control Tower identifies specific security considerations for generated agents, including the following:

- Agents with elevated permissions
- Agents that experience access-related errors
- Agents that have been inactive for more than 90 days but still retain active permissions

The access map visualizes relationships between agents, agentic workflows, and the tools they use, which helps stewards assess dependencies and potential impact before making changes.

If your organization uses AI Risk and Compliance, generated agents can be evaluated against organizational policies and regulatory frameworks. For example, the NIST AI Risk Management Framework and the EU Artificial Intelligence Act. Risk classification and compliance posture are surfaced on the **Risk and Compliance** tab of each AI asset record.

For more information, see AI Control Tower (legacy).

For more information on governance, agentic development and other ServiceNow development tools, see Governance for agentic development.

Cross-scope privileges

Build Agent can create Cross-Scope Privilege records to control which tables, scripts, and resources one scoped application can access from another. If a scoped app needs to read a table or call a script include from a different scope, Build Agent generates the appropriate privilege records. Cross-Scope Privileges are also useful for diagnosing and resolving `operation not allowed` errors between scoped apps.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Background script containment

When Build Agent runs a background script using one of the script tools, several containment controls limit the impact of server-side execution.

Approval before execution

Build Agent requires you to approve every script run before it executes. The approval prompt shows the generated script, its stated intent, and the target application scope so you can review the operation before it runs.

Scope restriction

Scripts run in the scope of the app being built by default. You can specify a different scope by name. If the scope name can't be resolved, the run fails rather than executing in the global scope.

On-demand rollback

After a script runs, Build Agent returns a rollback context you can use to revert the operation. Rollback requires the same approval step as the original run. Only scripts run by Build Agent can be reverted using this mechanism.

For more information on using the script tools, see Build Agent tools.

Custom skills and rules

Admins can define custom skills and rules to enforce organizational standards and guide Build Agent behavior across sessions. Rules apply automatically to every session at the configured scope. Skills provide on-demand guidance that Build Agent can draw on when needed.

For more information, see Configure custom skills and rules.

Build Agent limitations

Plan deployments and troubleshoot issues by learning about Build Agent constraints that affect deployment capabilities and performance.

Troubleshooting

If you encounter something that Build Agent can't currently do, complete that step directly on the ServiceNow AI Platform® and keep working with Build Agent for the rest of your development. Build Agent is designed to complement your workflow, not to replace it.

For details on troubleshooting, see Issues and solutions in Build Agent.

General limitations of Build Agent

Be aware of the following limitations when using Build Agent:

- AI-generated code requires human review for ACL correctness, security, platform conventions, and governance compliance. Put peer-review gates, static checks, and validation steps in place.
- You still must work with builders in ServiceNow Studio for app files and metadata types not yet supported in Build Agent, such as certain tables, UIs, or advanced features.

Note: Build Agent can work on existing applications. You can use Build Agent to enhance base workflows, such as creating business rules on existing tables.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Security in apps developed with Build Agent is at the application and API level, not at granular record and field level by default. However, you can use agentic development to build security onto the app by making requests. For examples, see Example prompts.
- Limited support for cross-product AI integration.

Build Agent can generate data models and business rules. You should understand where data from your AI-generated apps is stored on the ServiceNow AI Platform.

Build Agent and ServiceNow Fluent

To understand the upper limit of what Build Agent can do, review the ServiceNow Fluent documentation. If you're not familiar with the ServiceNow AI Platform, the ServiceNow Fluent documentation can help you determine what Build Agent can do. For more information, see the following topics:

- ServiceNow Fluent API reference
- ServiceNow Fluent

If ServiceNow Fluent does not support a metadata type, Build Agent cannot update it unless you provide precise platform-specific instructions.

Regulated environments

Build Agent, Autonomous Engineer, and Test Agent depend on off-instance services that have not completed the security compliance review required for regulated hosting environments. As a result, Build Agent v2 (Australia Patch 0 and Zurich 8 and higher) is not available in GCC, NSC, or FedRAMP environments.

Note: As of Brazil Patch 1, Build Agent, Autonomous Engineer, and Test Agent support ServiceNow Protected Platform (SPP) for Australia, the EU, and Singapore.

Customers in regulated environments must remain on Build Agent v1, which runs on-platform and is certified for regulated use. Autonomous Engineer, which depends on Build Agent v2 services, is also unavailable in regulated environments.

In regulated environments where Build Agent v2 is unavailable, you can continue to use the following:

- Core update set and Git workflows in ServiceNow Studio.
- Instance Scan

Feedback on Build Agent

To give feedback on Build Agent, see the Idea portal on Now Support.

Build Agent configuration

Build Agent connects your instance to AI-powered design workflows. Set up the required application and MCP server connections to start using it.

Note: Build Agent is enabled by default to create apps with AI, for example in ServiceNow Studio. To use other ServiceNow Otto products, such as the app generation skill, disable Build Agent. For example, using the setting in your ServiceNow Studio preferences.

Configuring Build Agent involves several sequential steps:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Installing the ServiceNow Otto for Creator application from the ServiceNow Store (Check your entitlements to determine whether you have access to this application.)
- Activating the required plugins for your version
- Establishing MCP server connections
- Enabling additional settings, such as for tests and custom instructions

Complete these steps to verify that all dependencies and integrations are in place before using Build Agent.

Because Autonomous Engineer runs through Build Agent as its execution layer, most settings you configure here also apply when Autonomous Engineer runs. For more information, see Configuring Autonomous Engineer settings in Build Agent.

Install Build Agent

For the Premium version of Build Agent, install the ServiceNow Otto for Creator application from the ServiceNow Store.

Before you begin

The Premium version of Build Agent is part of ServiceNow Otto for Creator, and is a ServiceNow AI Platform® feature that's activated by default when ServiceNow Otto for Creator is installed. You don't need an entitlement to start exploring Build Agent.

- Review the ServiceNow Otto for Creator application listing in the ServiceNow Store for information on dependencies, licensing or subscription requirements, and release compatibility.
- You can use Build Agent on a Personal Development Instance (PDI) or in Developer Sandboxes. For more information, see Accessing Build Agent in ServiceNow Studio.
- Build Agent supports the following models:
  - Azure OpenAI 5.4
  - Azure OpenAI 5.5
  - Azure OpenAI GPT 5.6 Sol
  - Google Gemini 3.7 Flash
  - Google Gemini 3.5 Flash
  - Gemini 2.5 Pro
  - Claude Opus 4.6
  - Claude Opus 4.8
  - Claude Opus 5.0

Role required: admin

About this task

Build Agent is enabled by default to create apps with AI, for example in ServiceNow Studio. To use other ServiceNow Otto products, such as the app generation skill, disable Build Agent. For example, using the setting in your ServiceNow Studio preferences.

Note: The trial app was formerly called "Build Agent" and has been renamed to "Build Agent (Trial)."

Instances running Australia Patch 5 or Zurich Patch 12 and later that have Build Agent installed receive automatic upgrades when a new version is published to the ServiceNow Store. That is,

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

you don't need to manually install new versions on qualifying instances. For more information on how automatic upgrades work, including blackout windows and failure resolution, see Auto-upgrade.

Note: Autonomous Engineer is a separate application available in the ServiceNow Store. Installing Autonomous Engineer also installs Build Agent as a dependency. Installing Build Agent alone does not install Autonomous Engineer.

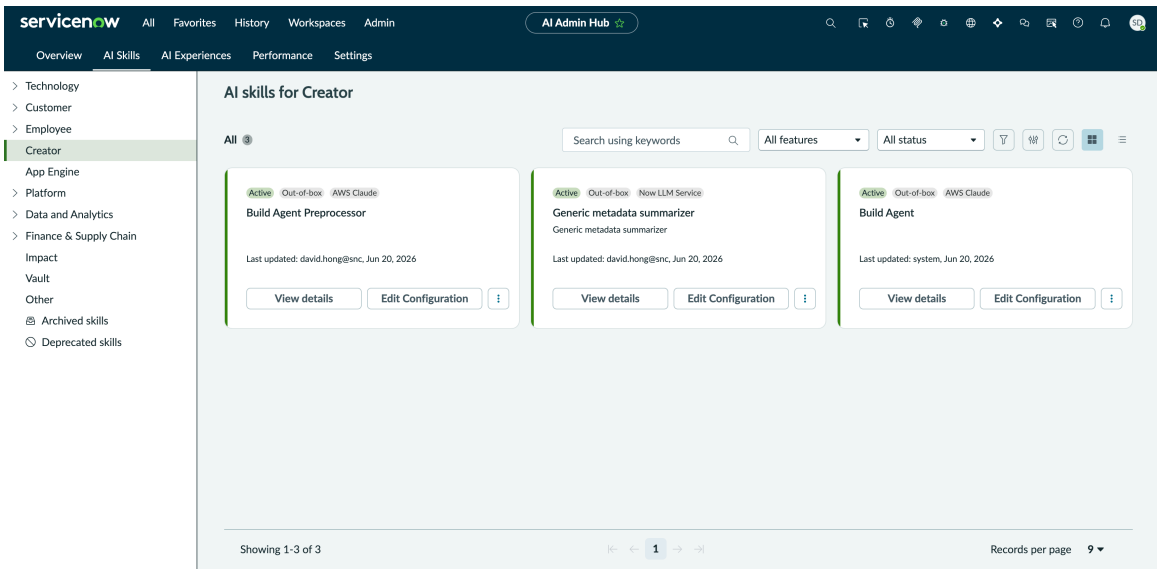
For instructions on installing ServiceNow products from the ServiceNow Store, see Install an application or plugin.

Procedure

1. From the ServiceNow Otto for Creator application page on the ServiceNow Store, select **Buy**.
2. After approval has been granted, on your instance, navigate to **All** > **System Applications** > **All Available Applications** > **All**.
3. Using the search bar, search for the ServiceNow Otto for Creator application (sn_now_creator).
4. Select **Install**.
5. Enable the Build Agent skill:

   a. Navigate to **All** > *AI Admin Hub*Skills.

   b. Select the **Creator** tab.



   c. Select **Turn on** to enable the skill.

   The skill is enabled for all users.

Related topics

Install ServiceNow Otto for Creator

Build Agent plugins

Plugins are required for Build Agent, and vary by version.

Build Agent is available in both ServiceNow Studio and the ServiceNow IDE.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: The trial app was formerly called "Build Agent" and has been renamed to "Build Agent (Trial)."

Plugins for Build Agent (Trial/Free)

Build Agent is available as a trial app on a freemium model, and is automatically installed on all instances.

The following plugins are required:

1. `sn_glider`
2. `sn_build_agent`

Plugins for Build Agent (Premium)

If you exceed the free interaction limit, you must wait 30 days for a reset, or install the paid version of Build Agent.

The `sn_now_creator` plugin is required, which contains the `sn_build_agent_pro` plugin.

Connect Build Agent to a supported MCP server

Connect a supported MCP server to Build Agent to access external tools and resources in the chat panel when building and editing apps.

Before you begin
Before you can enable an MCP server in Build Agent, an administrator must approve it as an AI asset in AI Control Tower. Each MCP server requires this approval, regardless of whether the server is enabled by default.

You must have Connect Hub installed.

Note: MCP setting currently apply to Build Agent, not Autonomous Engineer.

The end-to-end flow for making an MCP server available is:

1. The administrator adds the MCP server as a Workflow Data Fabric (WDF) connection.
2. The administrator approves the server as an AI asset in AI Control Tower.
3. You authenticate the connection in Personal Integrations.
4. You enable the MCP server in Build Agent settings.

For details on enabling MCP connections, see Client registration using custom connector.

Note: An allowlist process is required to connect the Build Agent MCP client to the Figma MCP server. Contact Now Support to initiate the process.

Individual MCP servers are enabled by default, but the complete flow must be completed before any server is available for use.

For details on adding a new MCP connection in Workflow Data Fabric, see Model Context Protocol connectors.

See MCP connections and Build Agent for a list of supported MCP servers.

Note: MCP connections are available in both ServiceNow Studio and the ServiceNow IDE.

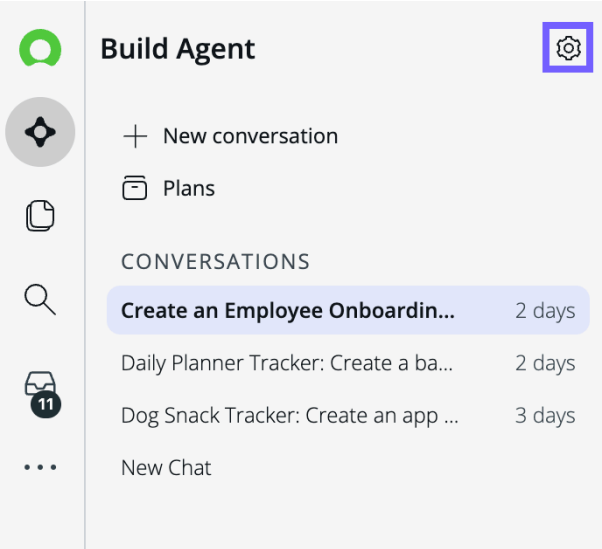
Role required: admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

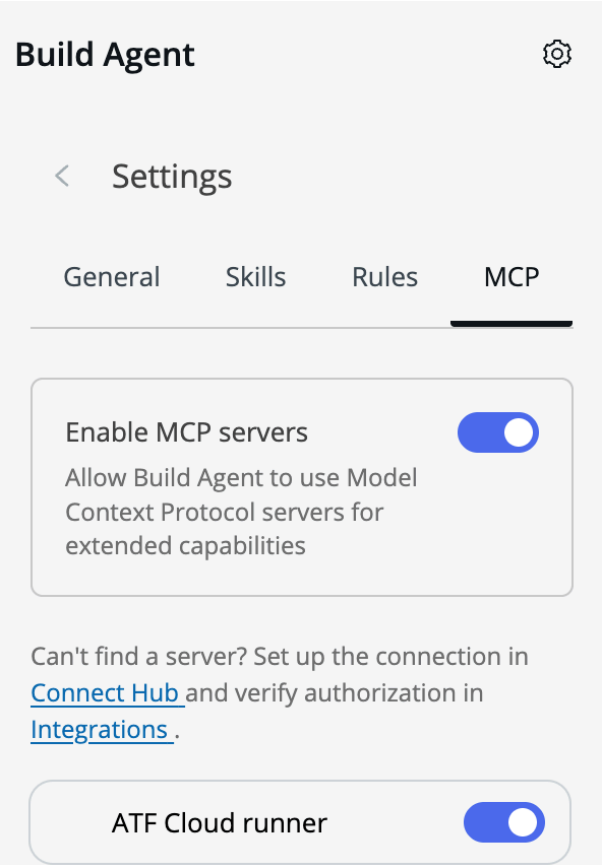
1. Navigate to **All** > **App Development** > **ServiceNow Studio**.
2. Select the Conversations icon in the Navigator panel to open Build Agent.
3. Select the Settings icon in the Build Agent chat panel.

Build Agent panel with Settings icon



4. Select the **Enable MCP servers** toggle on the **MCP** tab.
5. Select the MCP server that you want to enable.

Enable MCP servers setting

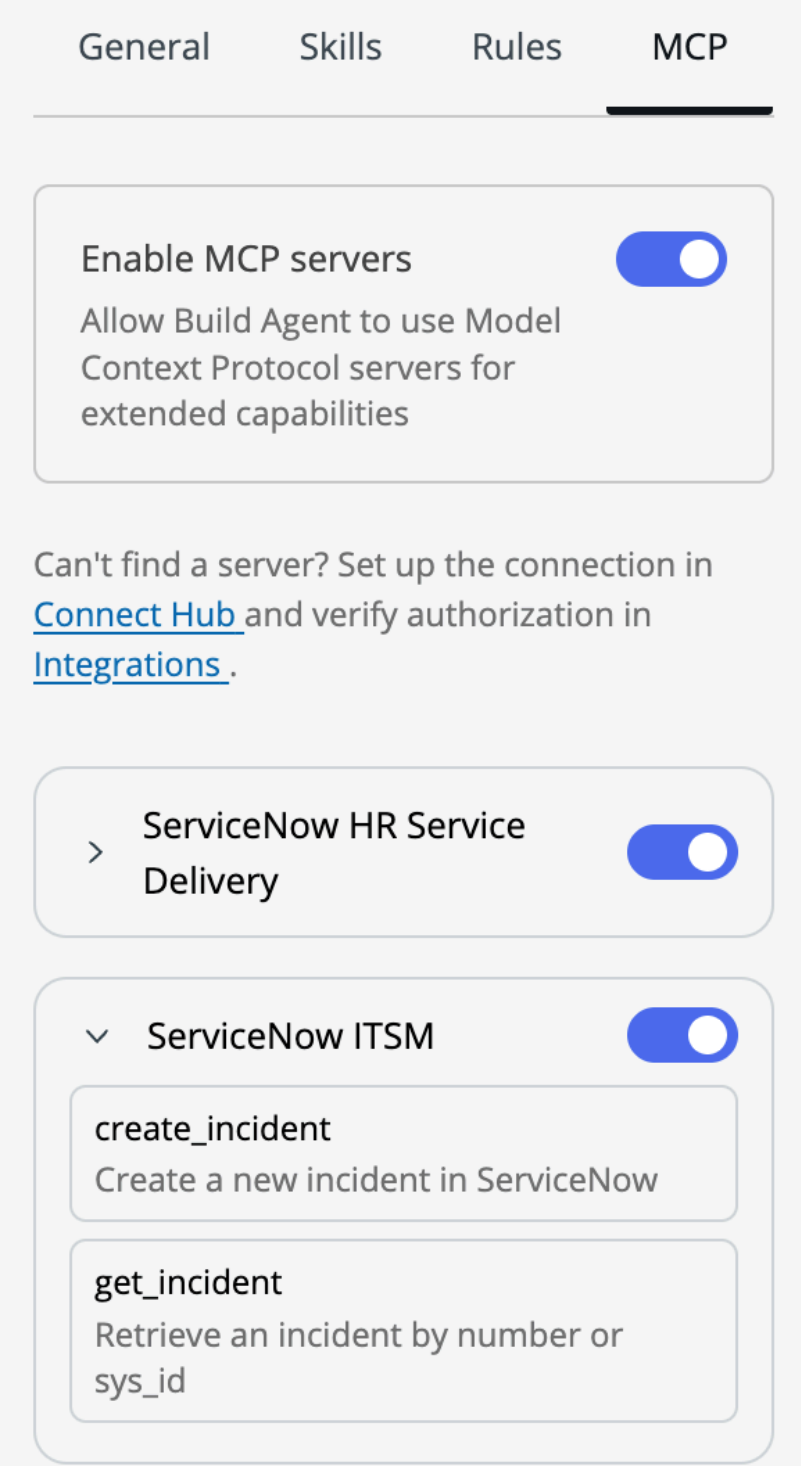


© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: If the MCP server you want isn't available, you must first set it up in Personal Integrations.

6. Select **Allow** in the confirmation dialog.
   You must do this only for the initial connection.
7. **Optional:** View the list of available tools that you can call for each MCP server by expanding the server name.

MCP tools list



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Result
After Build Agent authenticates the connection, you can chat with it to ask questions related to the MCP server.

Configure custom skills and rules

Create and manage custom skills and rules, or instructions to control how Build Agent behaves during a session. Rules are preloaded into every session automatically. Skills are available on demand when you or the agent invokes them by name.

Before you begin
Role required: admin

About this task
Both skills and rules are stored as records in the Build Agent Instruction table. Each record has a type that determines how Build Agent loads the instruction during a session.

- Use skills to provide task-specific guidance that Build Agent can use when needed. For example, internal guidelines for creating a specific type of record or handling a particular workflow pattern.
- Use rules to enforce consistent behavior every time the agent runs. For example, naming conventions, required fields, or organizational standards.

Each skill or rule instruction has an **Applies To** setting that controls which Build Agent sessions can use it. The coverage levels from broadest to narrowest are as follows:

Instance

Applies to all Build Agent sessions on the instance. Instance-level instructions take the highest precedence.

Application

Applies only when Build Agent is working within a record's application scope. Use application-level instructions to provide guidance specific to one scoped application.

User

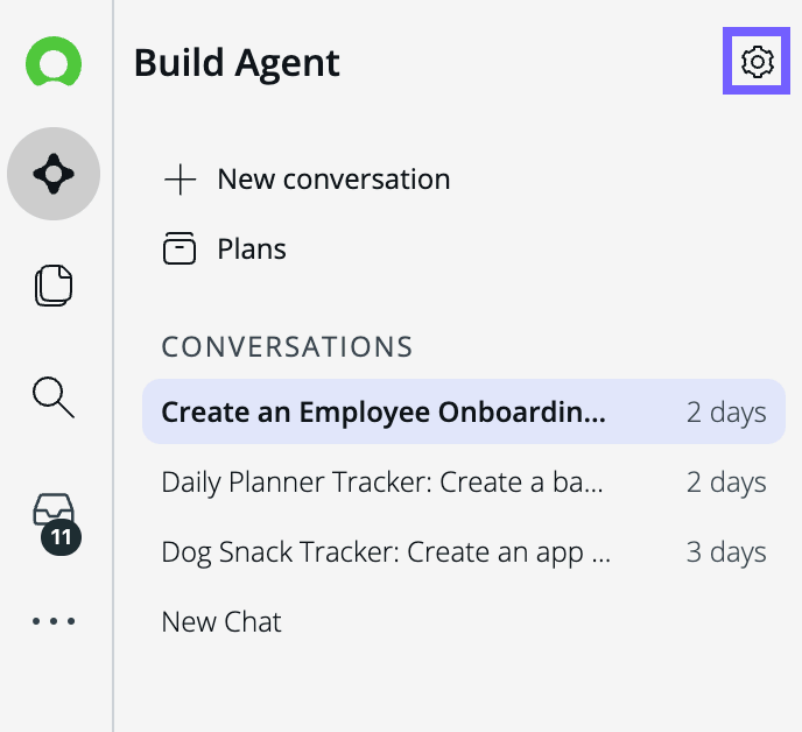
Applies only to sessions for the user who created the instruction. User-level instructions are stored separately from instance and application instructions.

Note: Check your entitlements to determine whether you have access to custom skills and rules in Build Agent.

Procedure
1. Navigate to **All** > **App Development** > **ServiceNow Studio**.
2. Select the Conversations icon in the Navigator panel to open Build Agent.
3. Select the Settings icon in the Build Agent chat panel.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Build Agent panel with Settings icon



4. Select the tab for the type of instruction you're creating.
   - Select the **Skills** tab to create a custom skill.
   - Select the **Rules** tab to create a custom rule.
5. Select the **Create skill** or **Create rule** button.
6. On the form, fill in the fields.

Build Agent create skill or rule fields

Field	Description
Name	Name of the instruction. Maximum 100 characters.
Type	Defines the type of instruction. ○ Select **Skill** for on-demand guidance ○ Select **Rule** to apply the instruction automatically to every session
Applies To	Controls which Build Agent sessions use the instruction. ○ Select **Instance** to apply to all sessions on the instance ○ Select **Application** to apply only when the session is in the record's application scope ○ Select **Me** to apply the instruction only to you
Active	Availability of the instruction for use. Selected by default.
Description	Description of what the skill or rule does. Maximum 1,000 characters.
Instructions for Build Agent	Instruction text that Build Agent follows. Plain text only, maximum 65,000 characters. Writing instructions in clear, structured prose helps Build Agent interpret the guidance accurately.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Some example instructions are:

- Rules:
  - `reuse_before_create`

    `Check for existing artifacts before generating new ones. Before creating a table, field, business rule, script include, flow, or UI action, search the instance for something that already covers the requirement. Prefer extending an approved existing artifact over creating a near-duplicate. If a close match exists, tell me about it and wait before proceeding.`

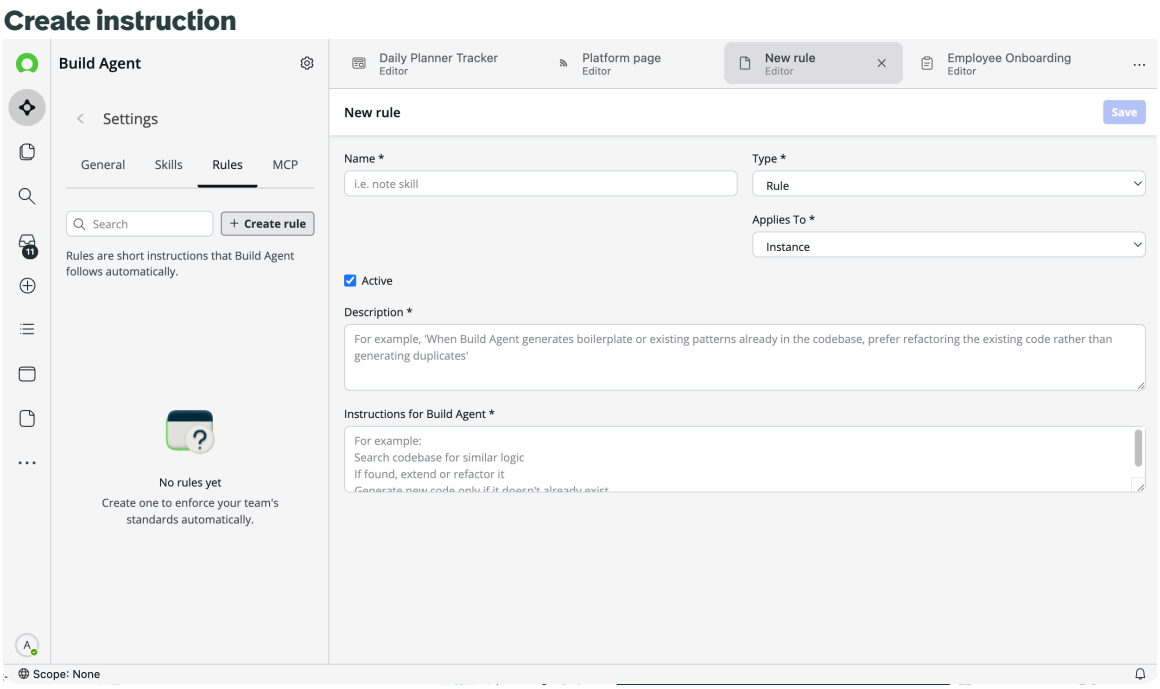
  - `naming_convention`

    `Use our application prefix. All tables and fields you create use our approved prefix followed by the application short name. Match existing naming in the app you're working in rather than introducing a new pattern.`

- Skill:

  `security_review`

  `Audit the app for access control and risky patterns. Review the current application and report on tables with missing or overly broad ACLs, client scripts using GlideRecord, scripts running with elevated privileges they don't need, and roles granted more broadly than the workflow requires. For each finding, give the artifact name, the risk in one sentence, and the change you recommend. don't make changes. Report only.`

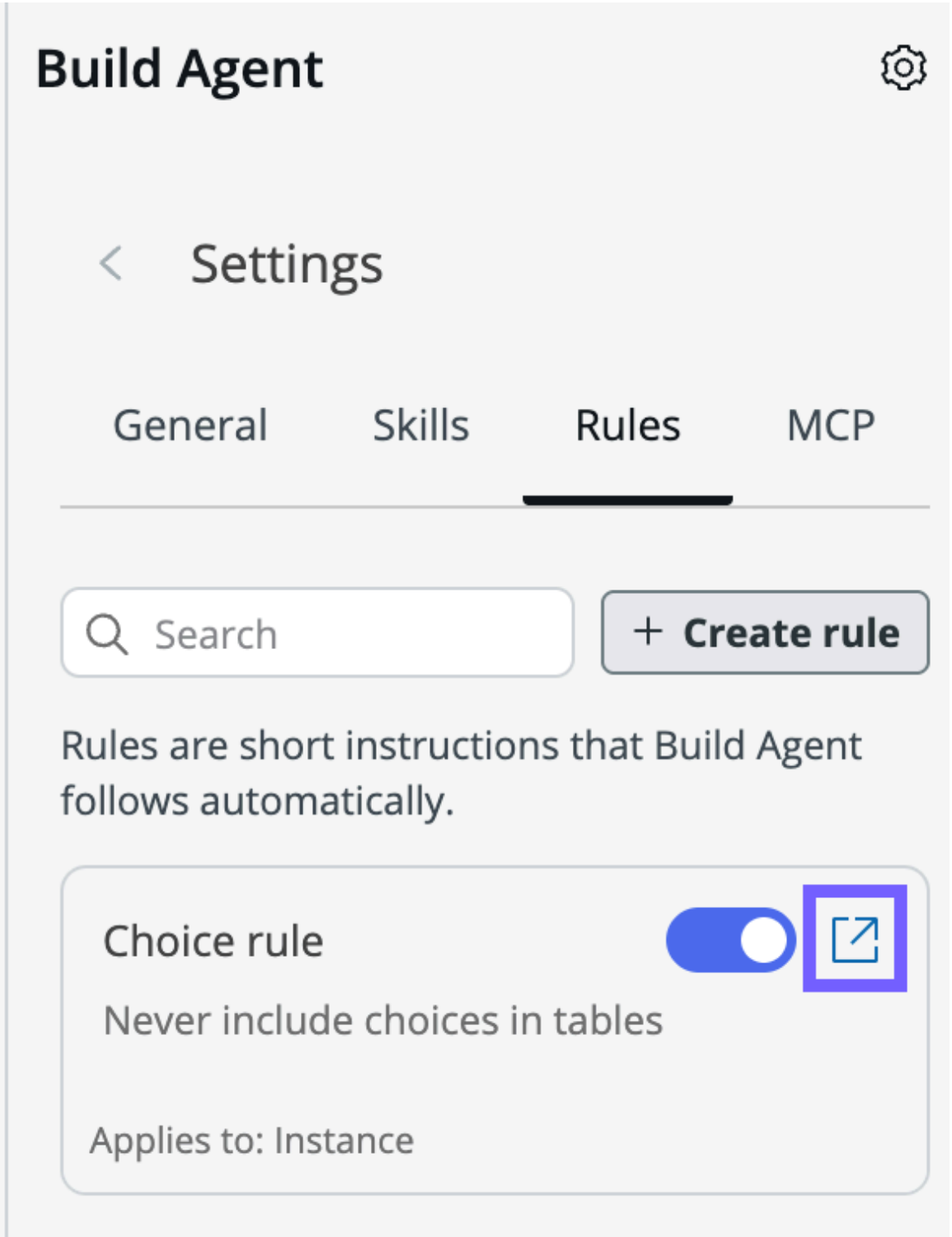


7. Select **Save**.
   The new skill or rule appears in the settings panel list. The panel updates automatically to reflect the change.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

8. **Optional:** Edit an existing instruction by selecting its Open rule record icon. Make changes to the instruction, then select **Save**.

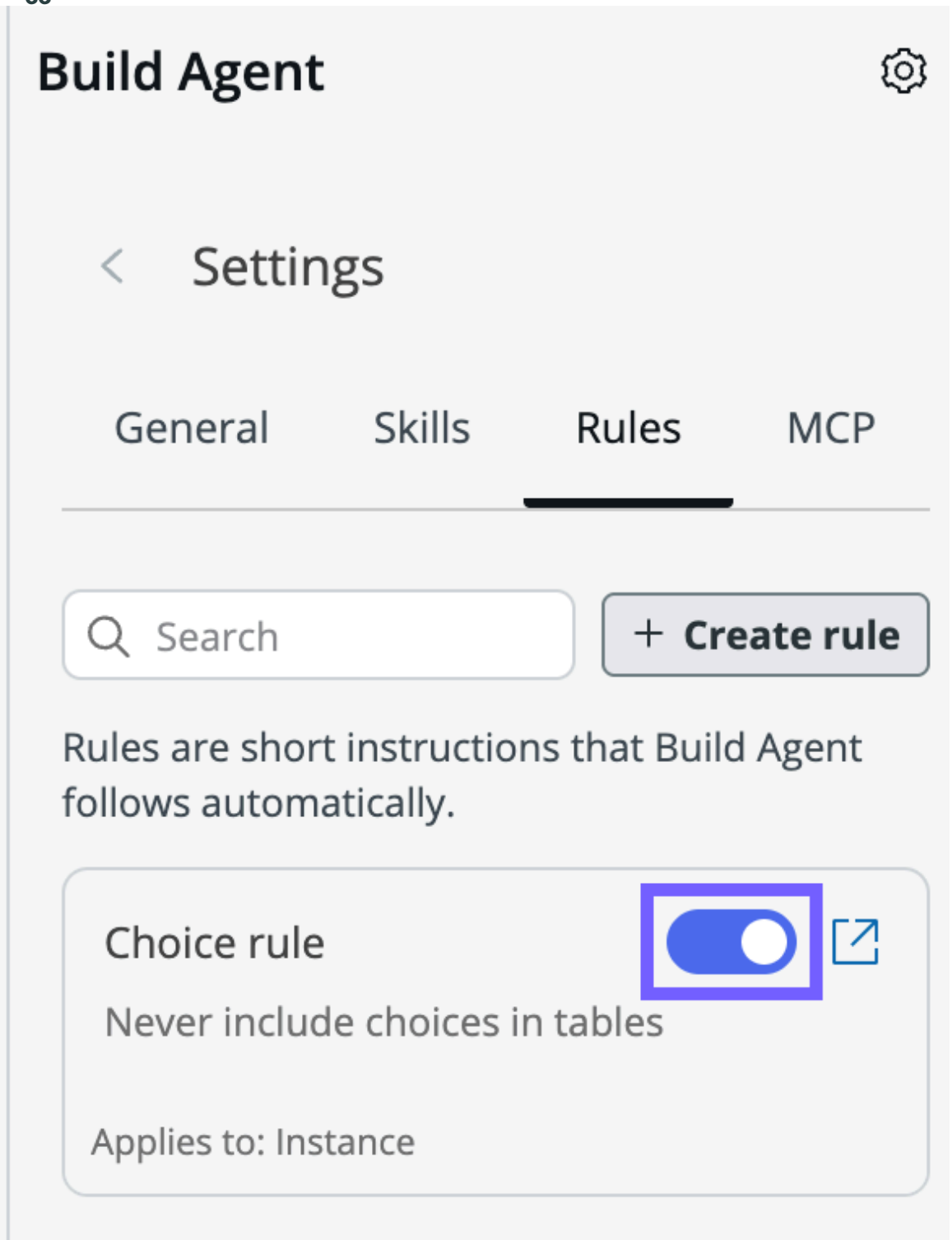
Edit instruction



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

9. Optional: Enable or disable an instruction by selecting its toggle.

Toggle an instruction on or off



What to do next

Active rules get dynamically loaded per turn in a Build Agent session, so they're available immediately. Active skills are available immediately on demand during a session.

Configure auto test prompting and UI tests

Configure test settings to enable automatic test prompting and execution of UI tests in Build Agent and Autonomous Engineer.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

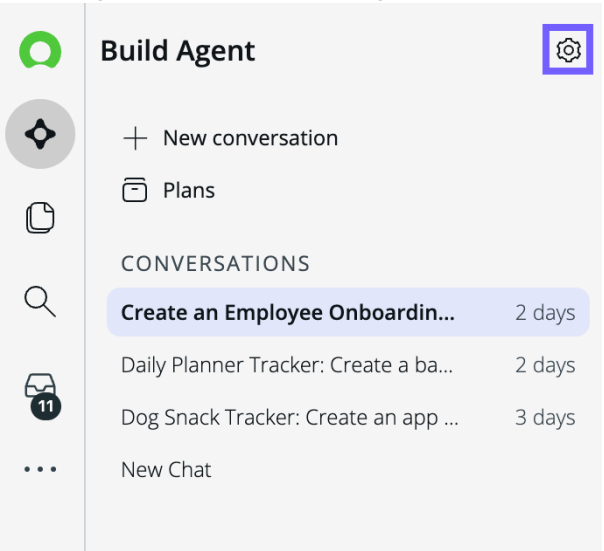
Before you begin
Role required: admin

About this task
Note: As of Brazil Patch 1, the Test Agent setting for Autonomous Engineer are enabled by default.

For more information on UI tests, see UI Test Script in Automated Test Framework (ATF).

Procedure
1. Navigate to **All** > **App Development** > **ServiceNow Studio** or **All** > **App Development** > **ServiceNow IDE**.
2. Select the Conversations icon in the Navigator panel to open Build Agent.
3. Select the Settings icon in the Build Agent chat panel.

Build Agent panel with Settings icon



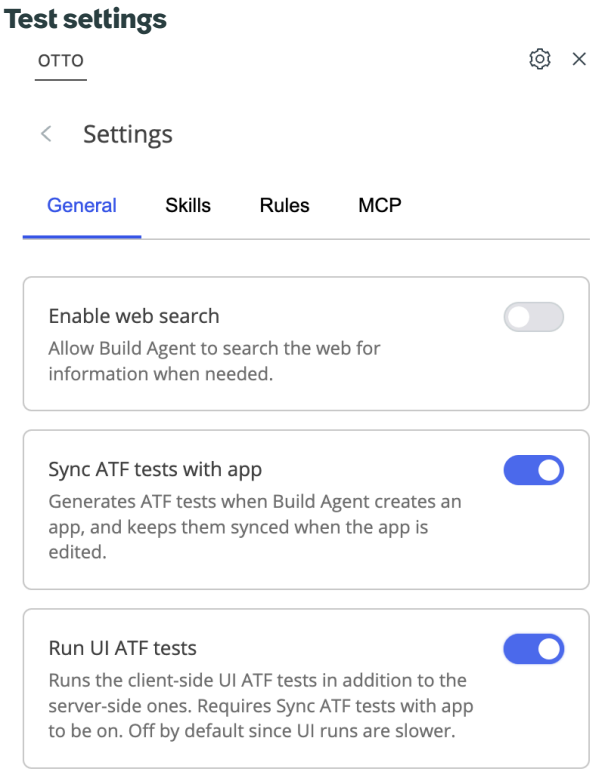
4. Enable the toggles for the following settings in the **General** tab as needed.

Test settings

Setting	Description
Sync ATF tests with app	Generates ATF tests when Build Agent or Autonomous Engineer creates an app, and keeps them synced when the app is edited.
Run UI ATF tests	Runs the client-side and server-side UI ATF tests.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Setting	Description
	○ The **Sync ATF tests with app** setting must be enabled. ○ This setting is off by default since UI runs are slower. ○ This setting applies to both Build Agent and Autonomous Engineer



Result
Build Agent and Autonomous Engineer run ATF tests in conversations based on your settings.

Configure web search

Enable the web search tool so that Build Agent can search the public web and retrieve relevant information when responding to queries.

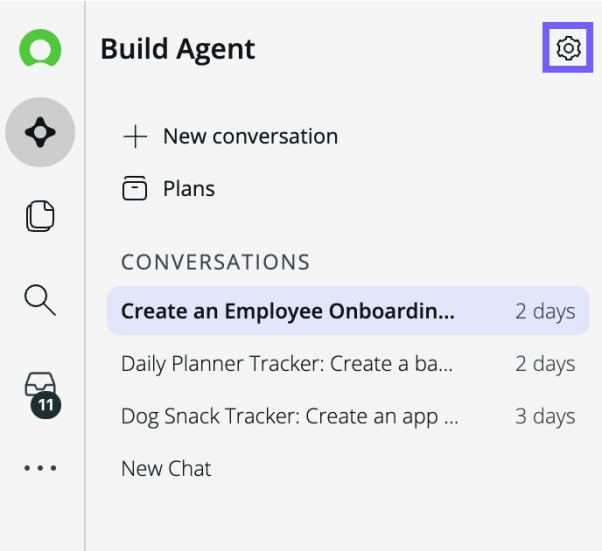
Before you begin
Role required: admin

Procedure

1. Navigate to **All** > **App Development** > **ServiceNow Studio** or **All** > **App Development** > **ServiceNow IDE**.
2. Select the Conversations icon in the Navigator panel to open Build Agent.
3. Select the Settings icon in the Build Agent chat panel.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Build Agent panel with Settings icon



4. In the **General** tab, turn on **Enable web search**.

What to do next

For more information about the web search tool, see Build Agent tools.

Use Build Agent

Build Agent is an autonomous AI agent native to ServiceNow that translates plain language instructions into ready-to-deploy applications and metadata, using the platform's domain language and guardrails. It's purpose-built for full-stack creation and editing across tables, flows, UI, and scripts, and it operates as the core engine for agentic development on the ServiceNow AI Platform.

Available in the ServiceNow IDE and ServiceNow Studio, Build Agent appears as a chat panel with a multi-turn conversation interface. Build Agent processes natural language prompts, autonomously generates full-stack applications, manages the entire build process, and responds to feedback. The agent generates all necessary code, organizes files clearly, and manages both the core logic and user interface components of applications.

Build Agent supports the following development tasks:

- Generate and edit enterprise-grade applications: Create full-stack ServiceNow applications from natural language prompts with enterprise-grade governance.
- Accelerate development workflows: Support flow design, catalog item creation, and Automated Test Framework (ATF) test creation.
- Enhance understanding: Explain how to best perform tasks on the ServiceNow AI Platform.
- Perform diverse code tasks: Refactor tables, explain and document code, validate and enhance existing applications, fix errors, run queries, and more.
- Support governance: Work inside platform scopes, roles, and testing workflows rather than as a detached external bot.
- Enable developer learning: Answer ServiceNow development questions, summarize documents, and provide practical examples.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

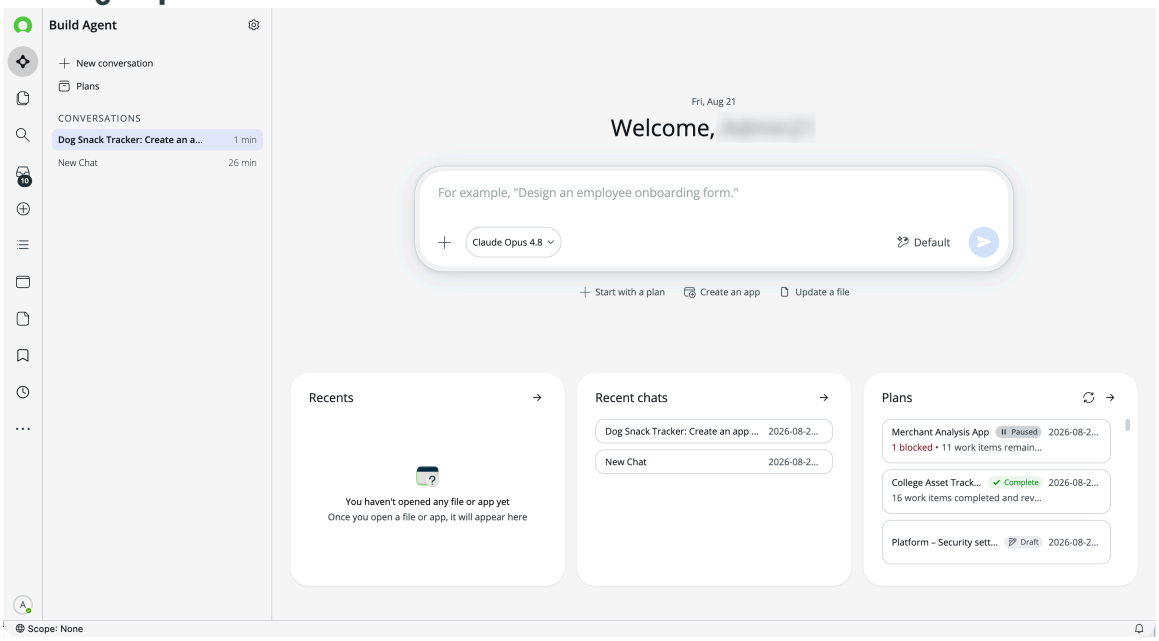
Accessing Build Agent in ServiceNow Studio

Build Agent is available in ServiceNow Studio for UI-first, declarative workflows. You can also use Build Agent the ServiceNow IDE within ServiceNow Studio for code-first, autonomous full-stack development.

Opening Build Agent

When you open ServiceNow Studio, the central chat area on the home page is where you chat with Build Agent to start a new conversation.

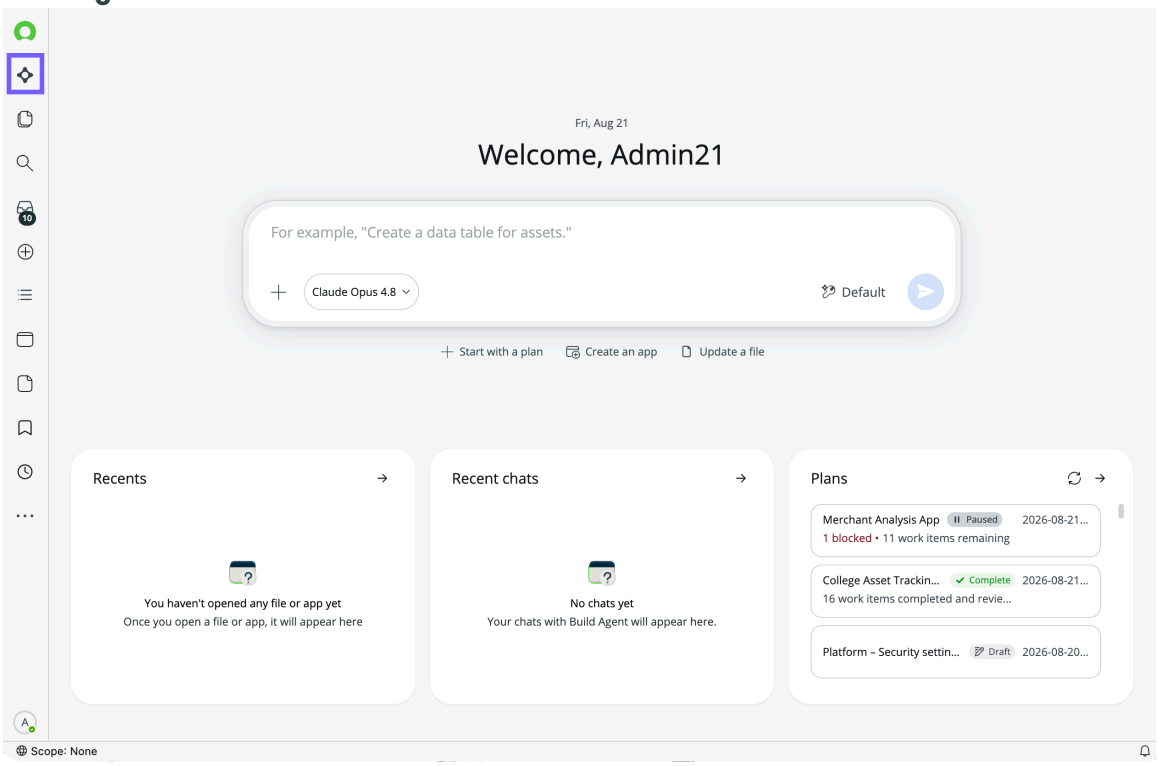
Build Agent panel in ServiceNow Studio



To open an existing conversation, select the Conversations icon in the Navigator panel. The Build Agent panel then opens on the left.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Build Agent in ServiceNow Studio



Note:

- Currently, only admins have permissions to use Build Agent.
- You must have the correct plugins installed to access Build Agent. For more information, see Install Build Agent.

You can start a Build Agent session with an active working set already loaded by right-clicking an artifact in the ServiceNow AI Platform, which displays a list of related metadata that you can edit. Then select the **Configure** option, which opens the record in ServiceNow Studio. From there, you can start a new Build Agent conversation about what you want to do with the metadata.

Build Agent environments

You can use Build Agent in both ServiceNow Studio, which contains the ServiceNow IDE if you want to create apps in source code. For more information, see Building apps in source code in ServiceNow Studio.

Your choice on where to access Build Agent depends on your role and workflow.

Build Agent environments

Usage	ServiceNow Studio	ServiceNow IDE
Generally best for	ServiceNow AI Platform developers, app owners, admins, citizen developers, and business analysts	Developers who prefer using an IDE on platform, TypeScript, and ServiceNow Fluent workflows
Code approach	Metadata-driven (tables, fields, business rules, scripts); embedded Build Agent	ServiceNow Fluent DSL (`.now.ts` files), TypeScript, React

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Build Agent environments (continued)

Usage	ServiceNow Studio	ServiceNow IDE
Deployment	System Update Sets; ServiceNow Studio package and install	ServiceNow SDK build, deploy, and install
UI creation	Forms, lists, workspaces (Next Experience), and catalog items	React UI pages and custom interfaces
Source control	System Update Sets; source control via linked repositories	Git built-in with branching
Local dev option	No; runs on the instance	Yes, VS Code with `@servicenow/now-sdk`

Key differences between ServiceNow Studio and the ServiceNow IDE

Area	ServiceNow Studio	ServiceNow IDE
Primary style	UI-first, declarative, metadata-centric	Code-first, conversational, full-stack
Typical users	Low-code builders, admins	Pro-code developers
Interaction model	Guided steps with suggestions, diffs, and summaries; selectable modes (guided, batch, one-shot)	Chat-driven autonomous generation; user approves edits, then build and deploy
Scope of automation	Create or update platform metadata (tables, flows, experiences) with dependency awareness	Generate and edit entire scoped or global apps (UI and backend), explain or repair code, run queries, create documentation
Change control	Strong guardrails; preview via ServiceNow Studio diff surfaces	Approval gates before writing; build and deploy workflow in the ServiceNow IDE
Best fit	Iterative configuration, edits, low-code delivery	Greenfield app creation, deep refactors, debugging, multi-artifact edits
Dependencies	Uses the ServiceNow Studio agentic experience layer and metadata explorers	Relies on the ServiceNow IDE workspace, file and metadata explorers, and build pipeline

Note: You can do conversational checkpoints with Build Agent and roll back to the last conversation checkpoint in both ServiceNow Studio and the ServiceNow IDE.

Build Agent is available in both ServiceNow Studio and the ServiceNow IDE, but each environment emphasizes a different development style. ServiceNow Studio provides a guided, UI-first experience that focuses on metadata creation and controlled, iterative changes. The ServiceNow IDE provides a code-first experience with an autonomous agent capable of generating and modifying full-stack applications through conversational prompts.

Choose the environment based on your skill set and the type of work:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- ServiceNow Studio: Low-code builders and admins who prefer declarative, metadata-driven workflows with previews, diffs, and guardrails.
- ServiceNow IDE: Pro-code developers who need conversational, code-centric generation, advanced customization, and end-to-end build and deploy steps.

Handing off ServiceNow Otto conversations to Build Agent

You can continue a ServiceNow Otto conversation from another product in Build Agent without repeating your intent. When ServiceNow Otto detects that you want to build or modify an application, it creates a handoff record. ServiceNow Studio then uses that handoff record to open Build Agent with your conversation context already loaded.

The handoff record stores a versioned summary of the conversation and a reference to the full transcript. ServiceNow Studio reads the handoff record and opens Build Agent with the summary pre-filled as an editable prompt in the chat panel. When that happens, Build Agent doesn't run automatically. You must review and edit the pre-filled prompt before submitting it. The full ServiceNow Otto transcript is attached to the session so you can reference details from the original conversation that the summary doesn't include.

Note: To use conversation handoff:

- You must have access to both ServiceNow Otto® and Build Agent.
- You must be working in ServiceNow Studio.

Build Agent and sandboxes

Use Build Agent in an isolated development environment with Developer Sandboxes. Sandboxes provide parallel development for distributed developers, with isolated metadata and Git integration. For more information, see Developer Sandboxes.

Build Agent and PDIs

You can access Build Agent on a Personal Development Instance (PDI). Developers using PDIs get 25 prompts per instance per 30-day cycle.

PDIs are updated to match the latest Build Agent for a consistent experience across both personal and production-track instances. Developers testing and building on PDIs have access to the same capabilities available in production environments. For more information on PDIs, see Personal developer instance guide.

Notes and limitations

Keep the following in mind when using Build Agent:

- Build Agent generates metadata supported by ServiceNow Fluent. Verify artifact compatibility before approval.
- Feature availability and UI details might differ between monthly releases. Confirm behavior against your instance version.

For more information on limitations, see Build Agent limitations.

Create an application using Build Agent default mode

Create a custom application by describing your requirements in plain language to Build Agent. The AI agent generates and builds the application code based on your input.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

Install and enable Build Agent. For more information, see Install Build Agent.

You can also use Autonomous Engineer to implement complex plans. For more information, see Autonomous Engineer in Build Agent.

If you prefer to access Build Agent in the ServiceNow IDE, you must first create a workspace. For more information, see Create a workspace in the ServiceNow IDE.

Note: Build Agent requires the ServiceNow SDK version 4.0 at a minimum. If you're using an older version, Build Agent prompts you to upgrade to ServiceNow SDK 4.0.

For some prompting guidelines and ideas, see Example prompts.

Role required: admin

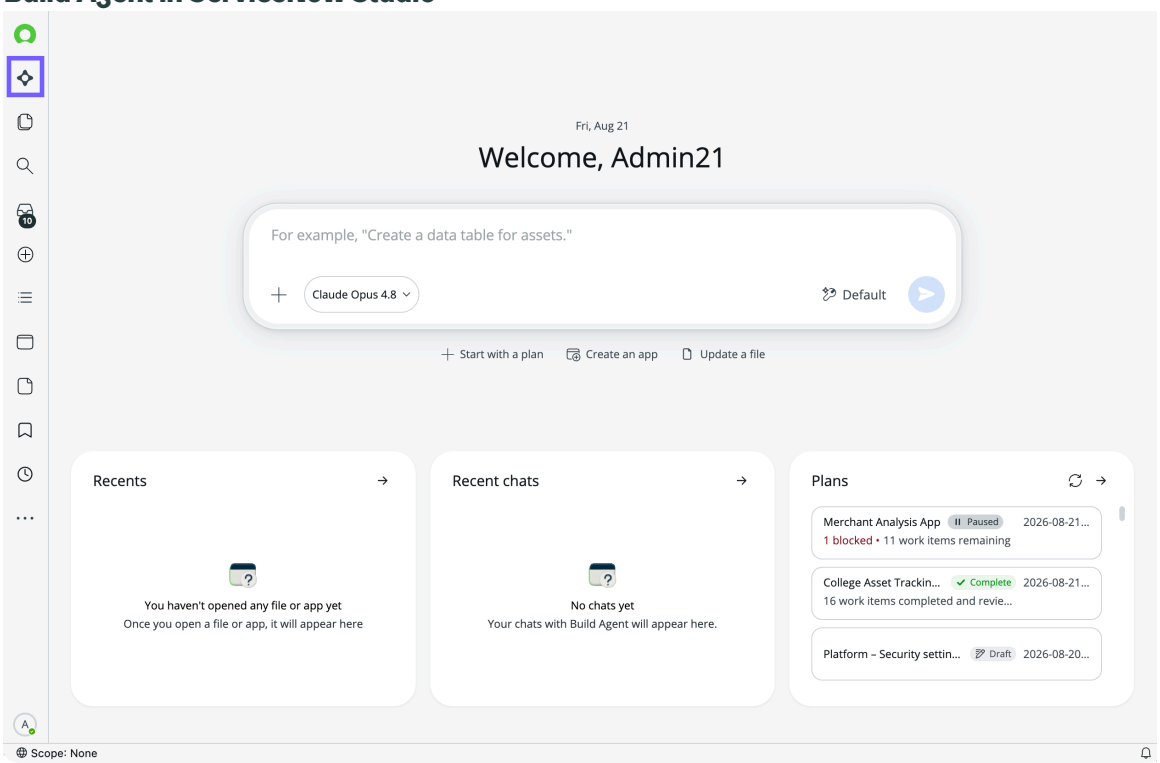
About this task

A ServiceNow app is a package that performs a specific task for a specified group of users. Think of an app as a container with a set of rules around who can access and edit it. For example, ServiceNow apps can include an API, a table, a workspace, a form, a flow, or any combination of those things.

Procedure

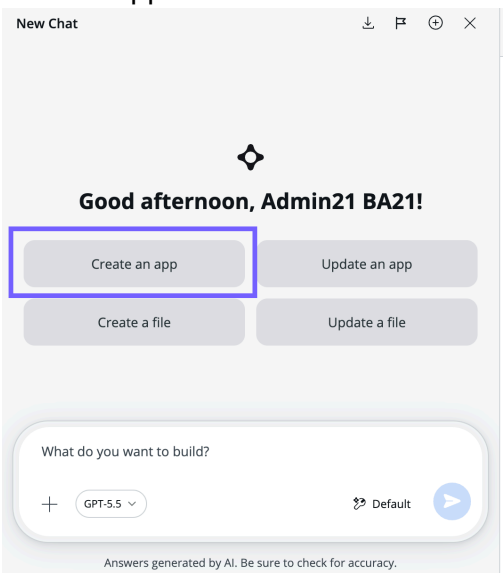
1. Navigate to **All** > **App Development** > **ServiceNow Studio**.
   You can also open Build Agent in the ServiceNow IDE if you prefer a more code-centric experience.
   ServiceNow Studio opens. Use the central chat area on the ServiceNow Studio home page to start a new Build Agent conversation. You can access an existing conversation by selecting the Conversations icon in the Navigator panel.

Build Agent in ServiceNow Studio



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

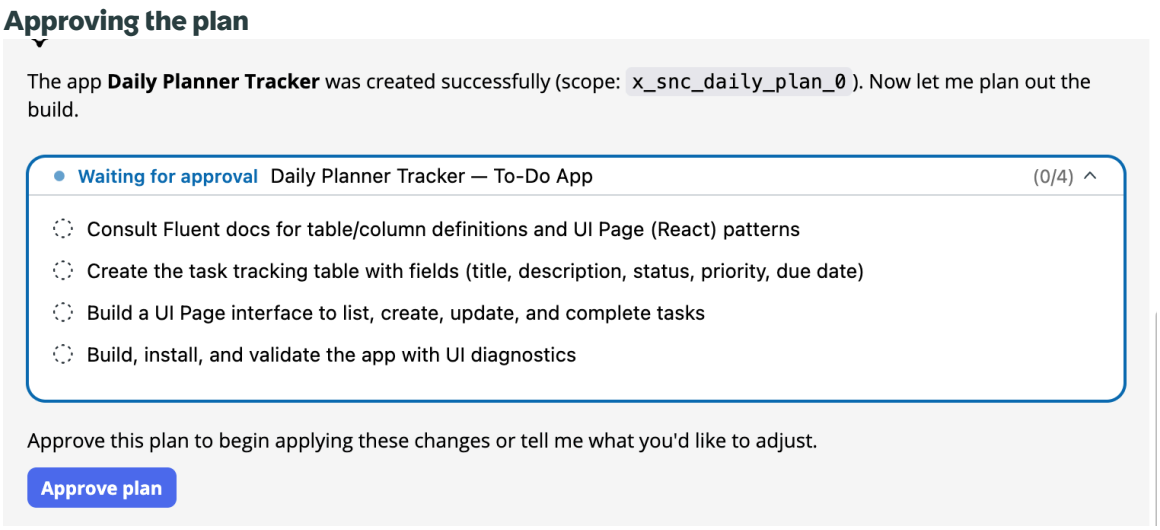
2. In the chat panel, with **Default** mode selected, describe the application that you want to create in plain language or select a prompt.

Scenario	Actions
Describe the app you want to create	Describe the application that you want to create, and then select the Send icon. For example, `Create a basic to-do task tracker app with a user interface. Name the application Planner.` You can also attach images, such as architectural diagrams or UI wireframes, to provide context for prompts.
Select a predetermined prompt	**a.** Select **Create an app** from the Application list.  **b.** Describe the application that you want to create, including as much context as possible, and then select the Send icon. For example, include roles, data requirements, and success criteria.

After reviewing your requirements, Build Agent requests confirmation to proceed with creating the application.

3. Instruct Build Agent to start developing the application by selecting **Approve plan**.
Or, if you're not satisfied with the plan, tell Build Agent what you change, and continue to prompt until you have a plan you like.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Build Agent can access ServiceNow knowledge sources and tools, which enable it to learn, analyze, and then create applications.

After creating the application, Build Agent builds the application.

Note: You can preview the code files before approval. But to see the actual metadata output, you must build and install the application on the instance.

4. If your application has a user interface, you can preview the app in a tab in ServiceNow Studio to inspect how it looks.
   You can keep prompting to refine the app, its appearance, and functionality. For more information, see Edit an existing application using Build Agent default mode.

Result

Review the application and its metadata in ServiceNow Studio using the change log. For more information, see Checkpoints and conversation change log. You can also review the application using the File Navigator in ServiceNow Studio. For more information, see Find an app or app file using the Navigator panel.

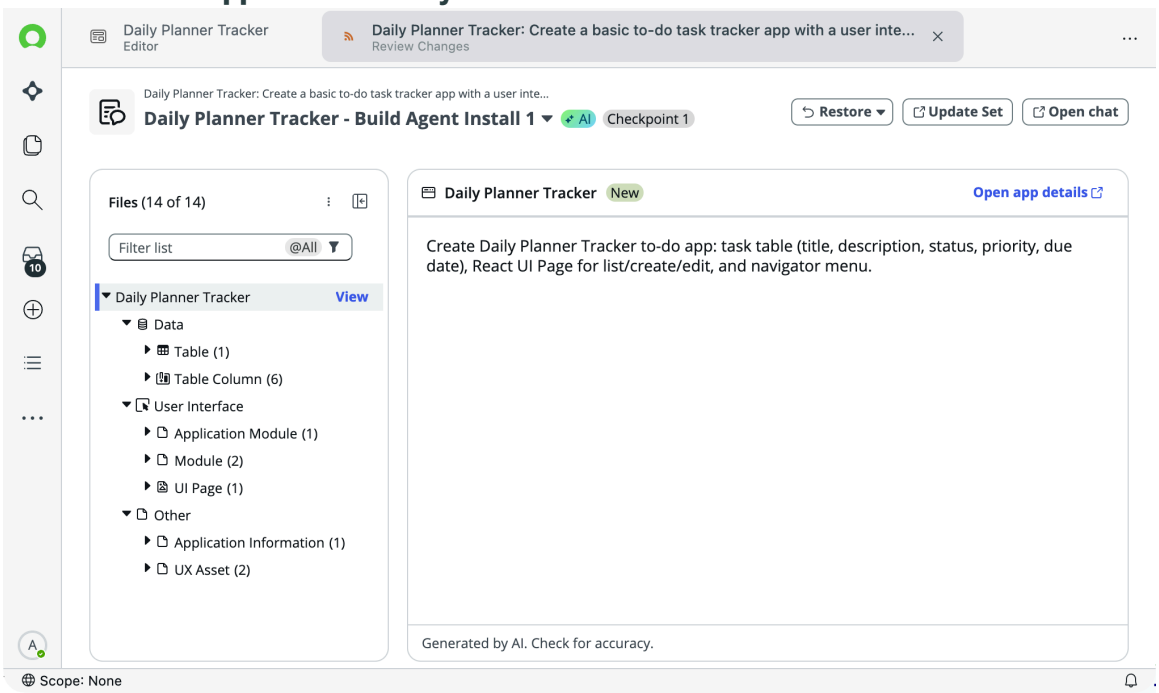
For information on deploying your application, see Deploying what you built with Build Agent.

For information on troubleshooting issues, see Issues and solutions in Build Agent.

Build Agent displays the details of your app in a new ServiceNow Studio tab.

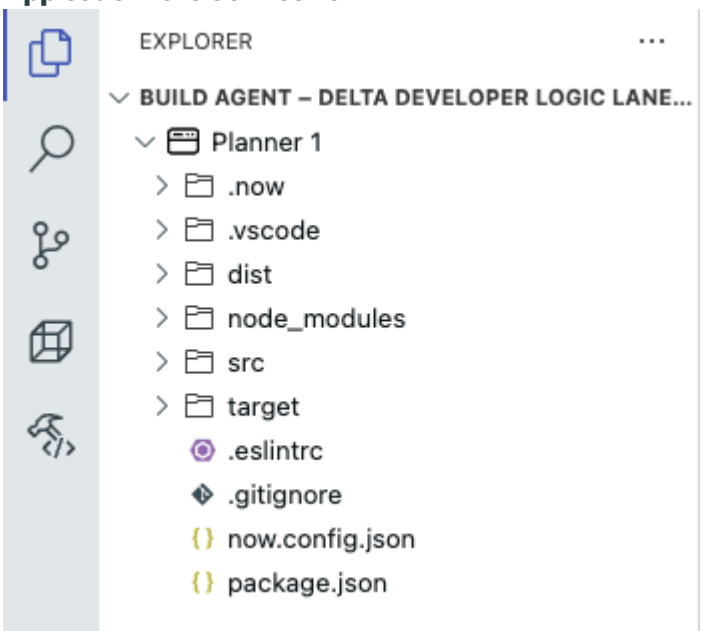
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Planner tracker app build summary



If you want to view source code, open the ServiceNow IDE within ServiceNow Studio and select the **Explorer** view from the Navigator panel. The ServiceNow Fluent application code and other source code in the `src` directory appears.

App code in the ServiceNow IDE



Edit an existing application using Build Agent default mode

Use natural language prompts to modify an existing application in Build Agent. You can describe changes in plain language and review, refine, and approve updates before the application is built and installed.

Before you begin

Install and enable Build Agent. For more information, see Install Build Agent.

For some prompting guidelines and ideas, see Example prompts.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

You can convert existing apps to ServiceNow Fluent and work on them in Build Agent. For more information, see Convert an application with the ServiceNow SDK.

Role required: admin

About this task

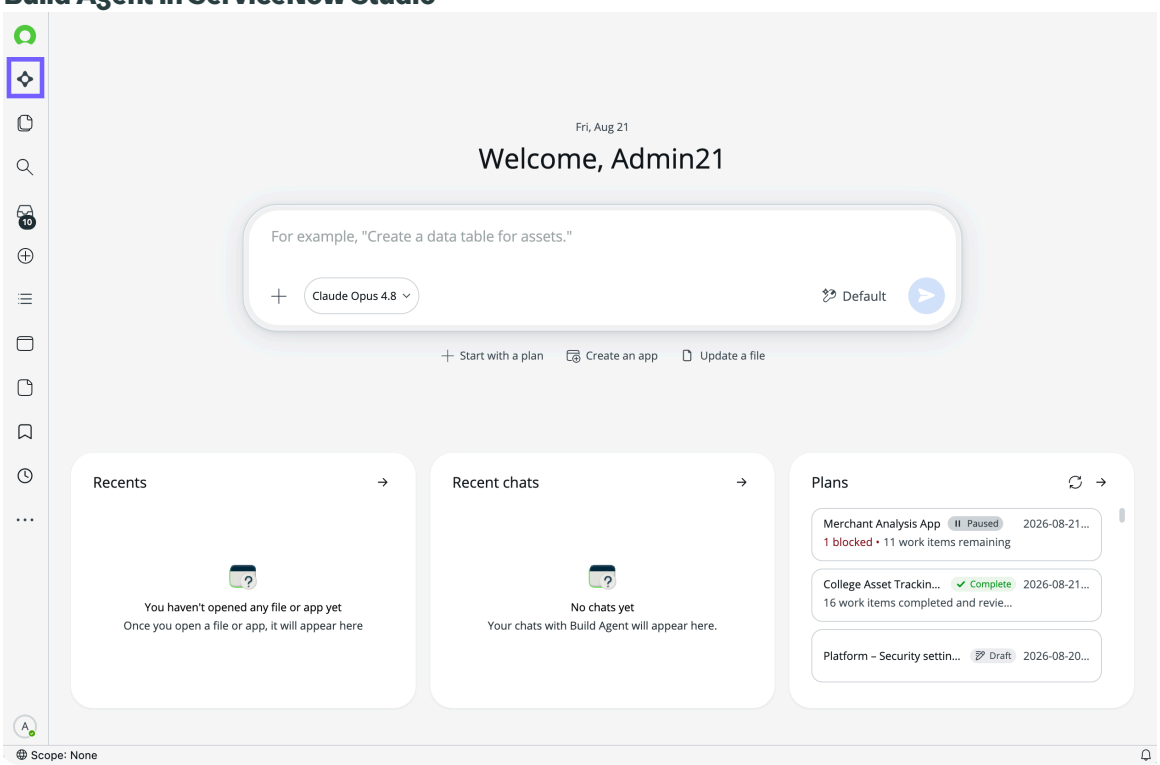
You can edit an application with Build Agent in both ServiceNow Studio and the ServiceNow IDE. If you're using the ServiceNow IDE, the application must be open in your workspace. You can add it to the workspace in the following ways:

- Applications created with Build Agent are automatically added to the workspace in which they were created. They can also be opened in other workspaces in the ServiceNow IDE.
- For applications that were not developed using the ServiceNow IDE, ServiceNow Studio, or the ServiceNow SDK, you must convert them into Fluent format to enable development within the Build Agent. You can prompt the Build Agent to use the open app tool to locate the application you want. Alternatively, you can search for an application directly within the Build Agent, and it will automatically use the open app tool. The open app tool can find an application, convert it to Fluent format, and then add the converted app to your workspace.
- Copy an existing application created with the ServiceNow IDE or ServiceNow SDK from a Git repository. For more information, see Clone a Git repository with the ServiceNow IDE.
- Right-click a record or artifact and select **Configure** to open the metadata editor in ServiceNow Studio.

Procedure

1. Navigate to **All** > **App Development** > **ServiceNow Studio**.
   You can also open Build Agent in the ServiceNow IDE if you prefer a more code-centric experience.
   ServiceNow Studio opens to a Build Agent chat box.

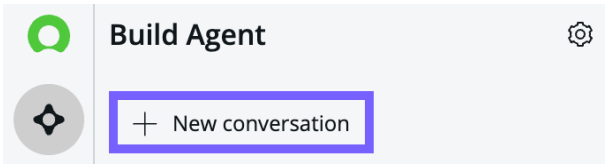
Build Agent in ServiceNow Studio



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

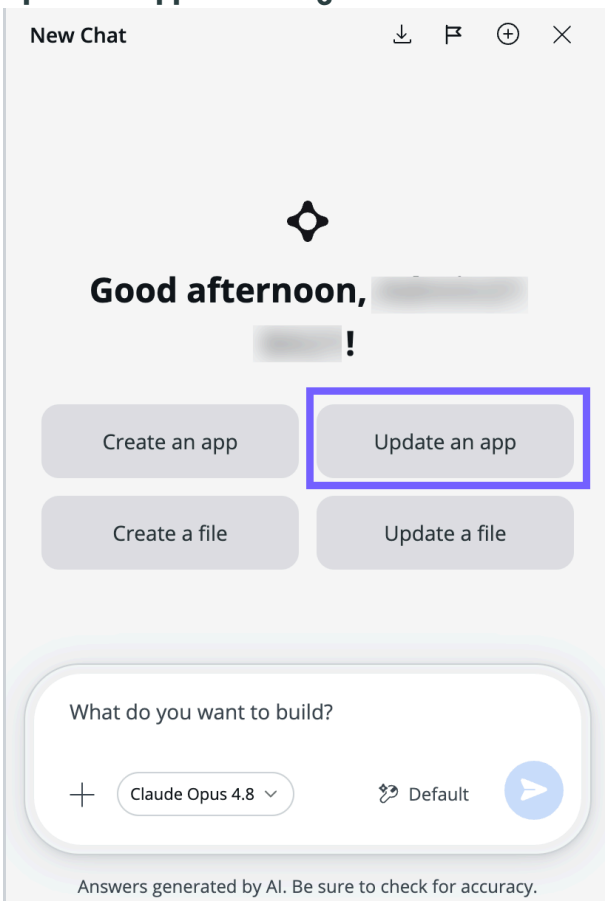
2. Display the Build Agent chat panel by selecting the Conversations icon in the Navigator panel.
3. Select **New conversation**.

Start a conversation



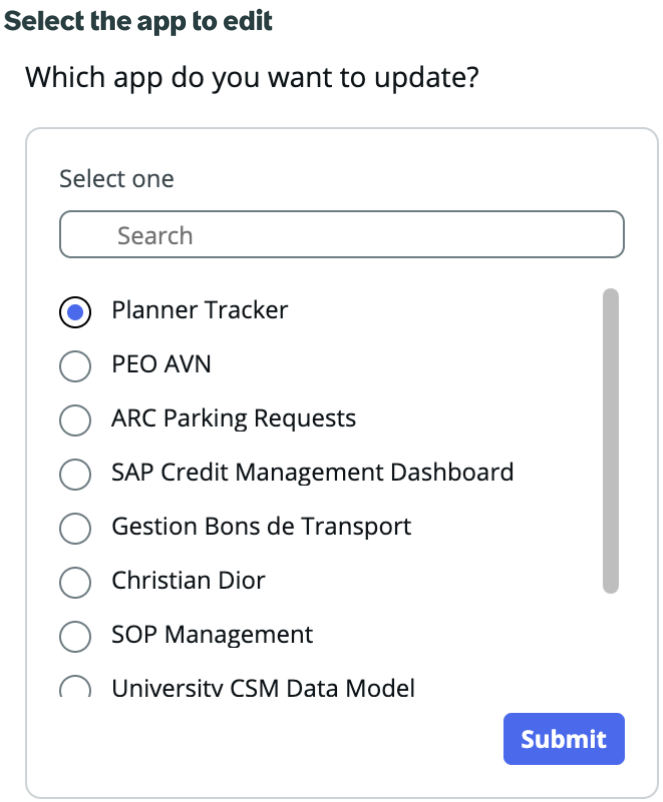
4. Select **Update an app** in the chat panel.

Update an app in Build Agent



5. Select the application that you want to edit and select the **Submit** button.
   Applications converted into ServiceNow Fluent code appear in the Application list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



6. Describe what you want to change in the application in plain language.
   Build Agent can provide you with some suggestions, such as `Update a table or add/modify fields`.
7. Select the Send icon .
   Build Agent starts updating the application.
8. Review the changes in the Change Log in a tab in ServiceNow Studio and continue iterating to refine the app.
   For more information on the change log, see Checkpoints and conversation change log.

   Review updates to generated tables, flows, and scripts, and preview any user interfaces created in your application. You can interact with the preview to make additional edits, for example, select a button in the **Preview** tab and tell Build Agent to `Change the color of the button to purple`.
9. Approve the changes:
   a. If prompted, select **Review all edits** to view the changes.
   b. If the changes are correct, select **Approve plan**.
10. **Optional:** Prompt and **Approve** Build Agent to build and deploy the application to an update set if it doesn't do so automatically.

Result

The application is built and installed.

Note: Follow the on-screen instructions, as the Build Agent functions interactively.

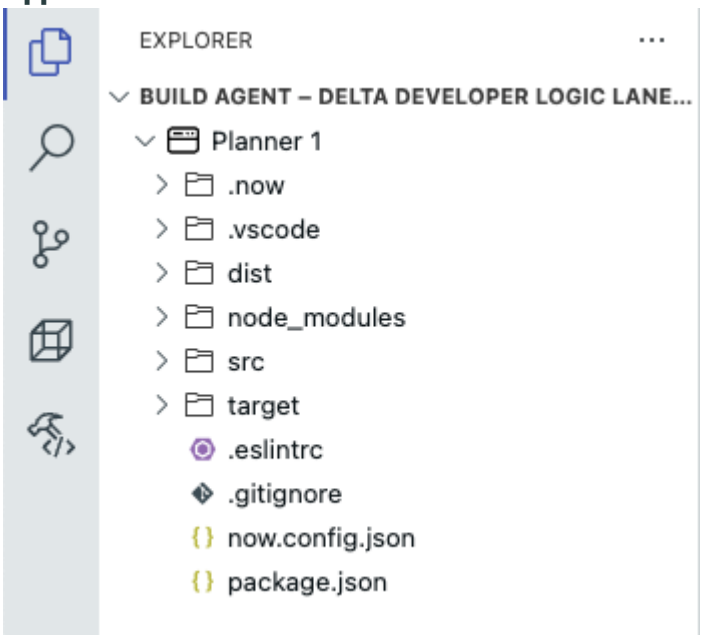
What to do next

For information on deploying your application, see Deploying what you built with Build Agent.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

If you want to view source code, open the ServiceNow IDE within ServiceNow Studio and select the **Explorer** view from the Navigator panel. The ServiceNow Fluent application code and other source code in the `src` directory appears.

App code in the ServiceNow IDE



Creating or updating an app file with Build Agent default mode

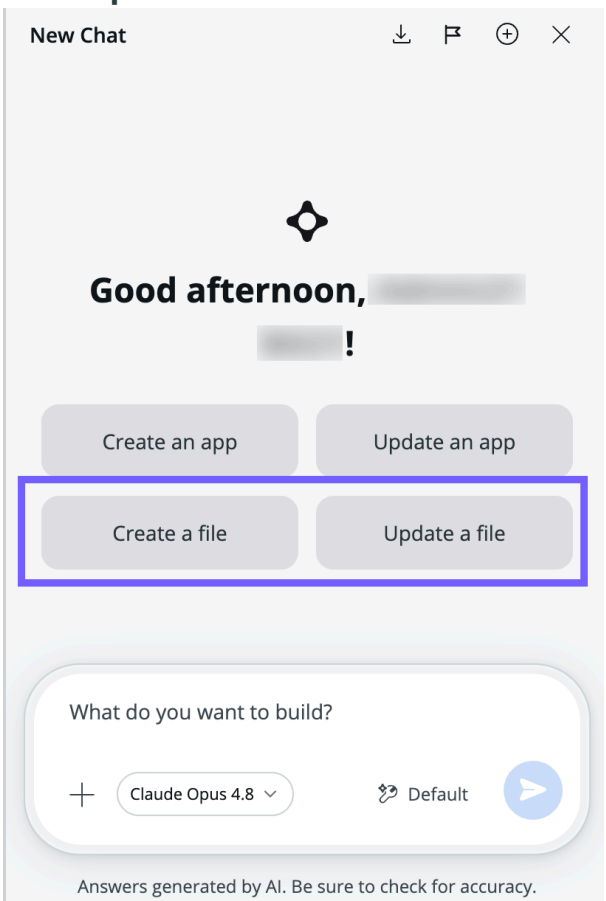
Use Build Agent to add new files or modify existing files in ServiceNow applications to support ongoing app development and maintenance.

The process for creating or updating an app file is very similar to creating an application. The main difference is that Build Agent prompts you to enter the application name or scope so it knows which application to update files in.

In a new chat with Build Agent, select **Create a file** or **Update a file** to get started, and continue the conversation from there.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add or update a file



Continue the conversation just like you did with creating an application, using the change log and checkpoints as you go. For more information, see Checkpoints and conversation change log and Revert app changes with Build Agent.

Review the application and its metadata in ServiceNow Studio using the change log. For more information, see Checkpoints and conversation change log. You can also review the application using the File Navigator in ServiceNow Studio. For more information, see Find an app or app file using the Navigator panel.

For information on deploying your application, see Deploying what you built with Build Agent.

Agentic workflows, agents, and skills

Build Agent can generate agentic workflows, agents, and skills scoped to your custom app.

Verify that you have access to ServiceNow Studio or the ServiceNow IDE.

Note: Your entitlements determine whether you have access to agentic workflows, agents, and skills in Build Agent.

Skills must be published before they can be used as a tool by another skill.

Build Agent supports the following tool creation:

- Skills:
  - *FlowAction
  - ServiceNow Otto Skill
  - Script - Inline

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Script - Explicit
- *SubFlow
- WebSearch

- Agents:
  - Catalog item: For more information, see Add a catalog item to an AI agent.
  - Conversational topic: For more information, see Add a conversational topic to an AI agent.
  - Flow action: For more information, see Add a flow action to an AI agent.
  - ServiceNow Otto Skill: For more information, see Add a generative AI skill to an AI agent.
  - Record operations: For more information, see Add a record operation to an AI agent.
  - Script: For more information, see Add a script to an AI agent.
  - **Search retrieval: For more information, see Add a search retrieval to an AI agent.
  - Subflow: For more information, see Add a subflow to an AI agent.
  - WebSearch: For more information, see Add a web search to an AI agent.

Note:

1. *You can use only FlowActions and Subflows that are already part of your app scope. Build Agent reuses existing FlowActions and Subflows instead of creating new ones.
2. **When you create an agent that requires knowledge retrieval, Build Agent can configure a search retrieval tool scoped to your use case. The tool queries your existing search profiles to identify the best match, retrieves the associated search sources and return fields, and checks whether semantic search indexes are available.

The agentic capability turns business requirements into fully configured agents, skills, and agentic workflows for custom applications. It first evaluates existing app context, such as tables, roles, business rules, and metadata. Then Build Agent applies a four-phase pattern to generate artifacts scoped to your application's data model, roles, and ACLs:

Assess

Evaluates data model readiness for AI.

Analyze

Identifies use cases from existing tables, roles, and flows.

Design

Proposes an architecture with a recommended quick-win.

Build

Generates agents and skills scoped to the application's data model, roles, and ACLs.

In-App agents can work with any custom scoped application. Applications with real data, established roles, and defined automation produce more targeted, domain-specific artifacts. Applications with no data or configuration produce agents that aren't scoped to a specific data model.

Note: Generated agents are automatically registered in AI Control Tower for centralized governance, and artifacts deploy through standard update sets.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Create agentic workflows, agents, and skills

Build custom agentic workflows, AI agents, and skills for your applications using automated generation tools with Build Agent. You can streamline development by creating the necessary instructions, tools, and access controls based on your requirements.

Before you begin

Skills must be published before they can be used as a tool by another skill.

Note: Check your entitlements to determine whether you have access to agentic workflows, agents, and skills in Build Agent.

Role required: admin

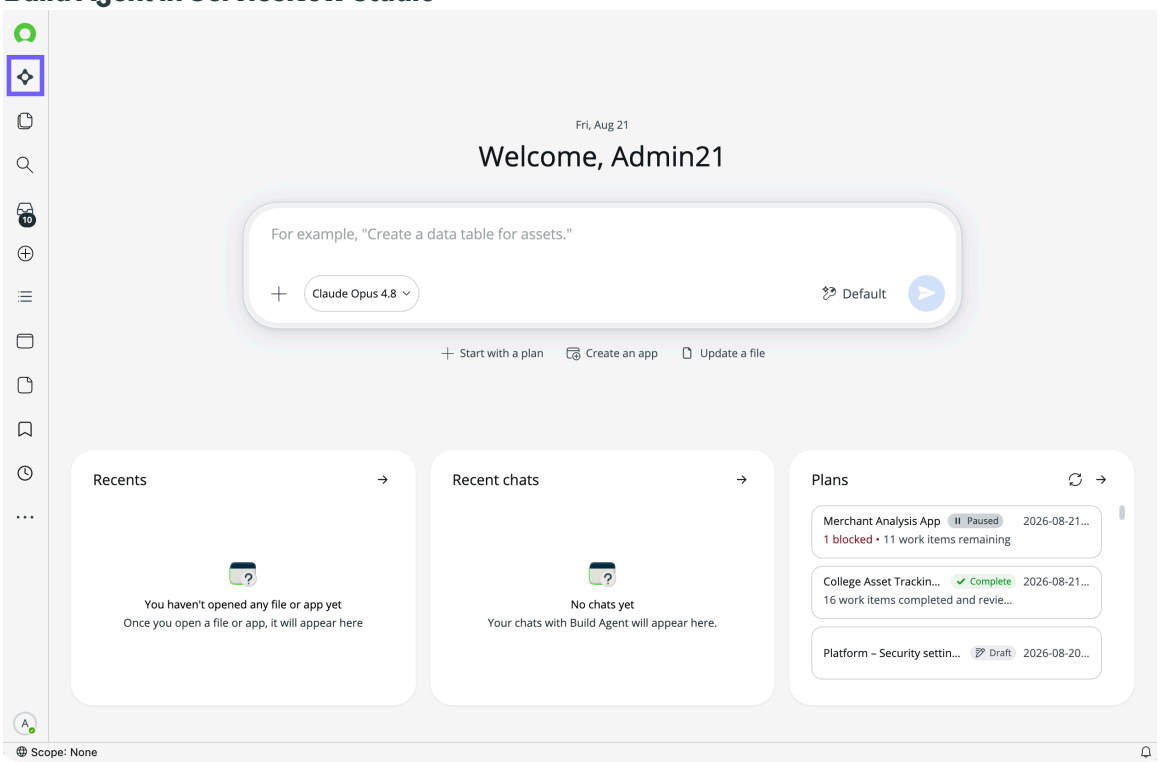
About this task

For details on creating agentic workflows, agents, and skills, see Agentic workflows, agents, and skills.

Procedure

1. Navigate to **All** > **App Development** > **ServiceNow Studio**.
   You can also open Build Agent in the ServiceNow IDE if you prefer a more code-centric experience.
   ServiceNow Studio opens. Use the central chat area on the ServiceNow Studio home page to start a new Build Agent conversation. You can access an existing conversation by selecting the Conversations icon in the Navigator panel.

Build Agent in ServiceNow Studio



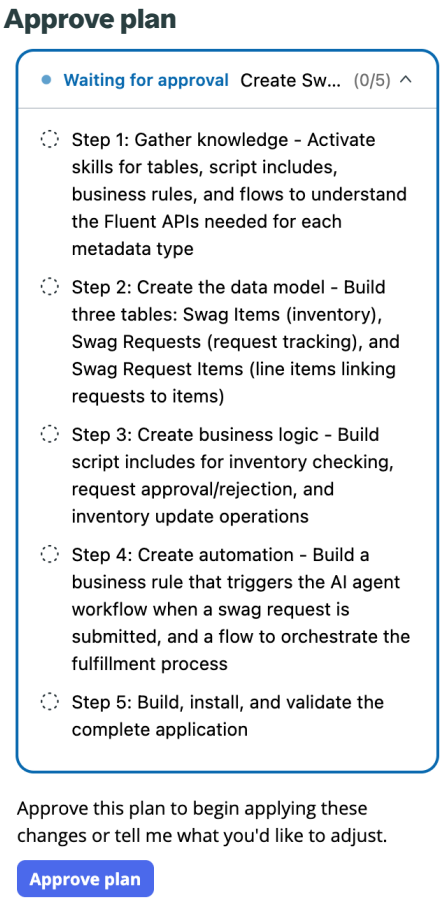
2. **Optional:** Select your application from the Build Agent chat panel drop-down list.
3. Prompt to add AI to your app.
   Enter a prompt describing the agentic workflow, agent, or skill that you want to create. For example:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- To create an agentic workflow: `Create an onboarding agentic workflow for contractors featuring three sequential AI agents:`
  - `Intake Agent: Validates requests, conducts online background checks, verifies budget, and calculates costs using web searches and scripts.`
  - `Background Check Agent: Assesses risk and performs tiered checks on identity, employment, credit, and references. Also verifies credentials and identifies necessary training.`
  - `Provisioning Agent: Evaluates access needs, assigns equipment, generates credentials, provisions system access, finds training resources, and sends welcome packages.`
- To create an agent: `Create a Swag Fulfillment AI agent that can review and approve swag requests. The agent must review incoming swag requests, check inventory availability for all requested items, approve requests autonomously, update inventory, and request records when taking actions.`
- To create a skill: `Create a summarization skill that summarizes travel requests and provides approval suggestions based on the company's travel policy.`
- If you don't know what agents or skills you need: `Analyze my application. Review the tables, fields, business rules, and workflows. Identify the most repetitive manual tasks that fulfillers perform and suggest which ones could benefit from AI agent or skill.`

4. Review and approve the plan by selecting **Approve plan**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Build Agent starts creating the agentic workflow, agent, or skill, along with its instructions, tools, access control lists (ACLs), and supporting scripts.

5. When the build is complete, confirm the installation.
   Build Agent deploys the workflow, agent, or skill to the development environment.

What to do next

After Build Agent generates agents and skills, complete the following steps to move them from creation to production.

1. Test each skill in ServiceNow Otto Skill Kit to validate prompt behavior against sample records and review output quality before publishing.
2. Test the agent in AI Agent Studio to validate the end-to-end workflow and confirm that tool invocations work correctly.
3. Activate triggers in AI Agent Studio. Triggers are not activated automatically and must be enabled separately after generation.
4. Deploy the custom app with its agents and skills as a standard update set.

Related topics

- AI Agent Studio overview
- Create an AI agent
- Manually test the execution of an AI agent
- Exploring AI Skill Kit
- Test a prompt

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Revert app changes with Build Agent

Restore your development to a previous state when you want to undo recent changes. Use checkpoints created during Build Agent conversations to revert both code and chat history.

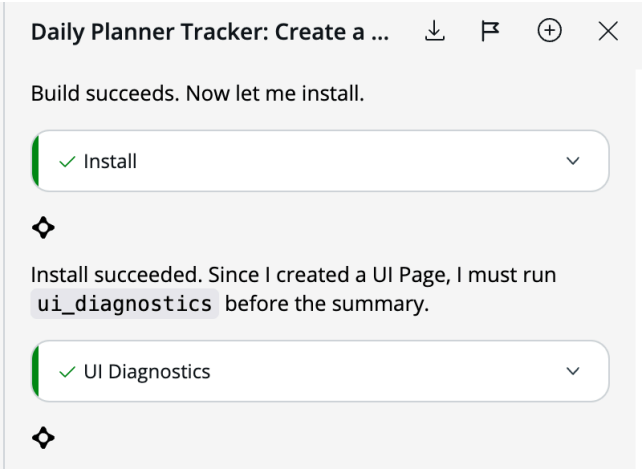
Before you begin

Role required: admin

Procedure

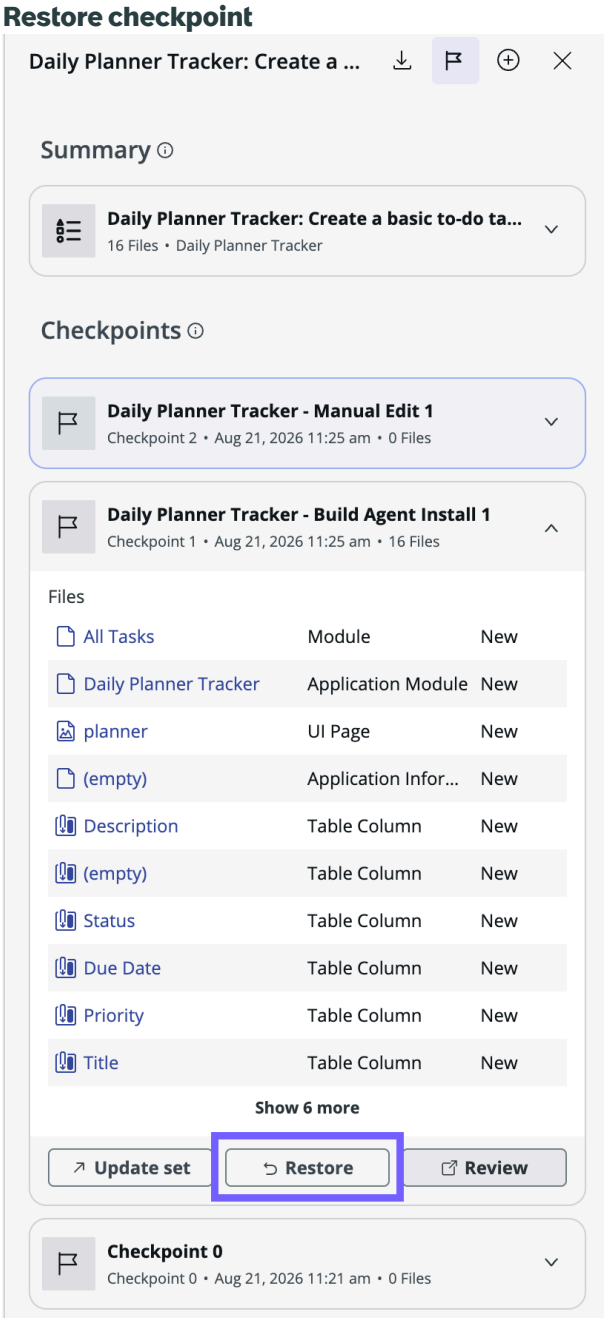
1. Navigate to **All** > **App Development** > **ServiceNow Studio**.
   You can also open Build Agent in the ServiceNow IDE if you prefer a more code-centric experience.
2. Select the Conversations icon in the Navigator panel to open Build Agent.
3. Select the chat that contains checkpoints you can revert to.
4. View all available checkpoints by selecting the checkpoints icon

Checkpoints in chat panel



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

5. Select the checkpoint you want to revert back to, and select **Restore**.



Result

Build Agent reverts your changes both in your application and in the chat.

Test what you built with Build Agent

Test Agent generates test coverage for code created by Build Agent, executes tests, and performs root cause analysis (RCA) on failures. Prompt Test Agent to complete build-to-test workflows in a single development session without manual test authoring or failure investigation.

Test Agent extends Build Agent by making every build safe before release. After Build Agent produces code changes in a development instance, Test Agent consumes the same prompt and code context. It uses those to author functional Automated Test Framework (ATF) tests, execute those tests, and triage any failures automatically.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- For details on configuring tests in Build Agent, see Configure auto test prompting and UI tests.
- For more information ATF test generation, see ATF test generation in Build Agent.
- For complete documentation on using Test Agent, see Test Agent.

If a test fails, Test Agent performs an RCA. Then it either auto-applies safe fixes or surfaces actionable guidance in the chat panel so you can resolve the issue without leaving ServiceNow Studio or the ServiceNow IDE.

Customer outcomes

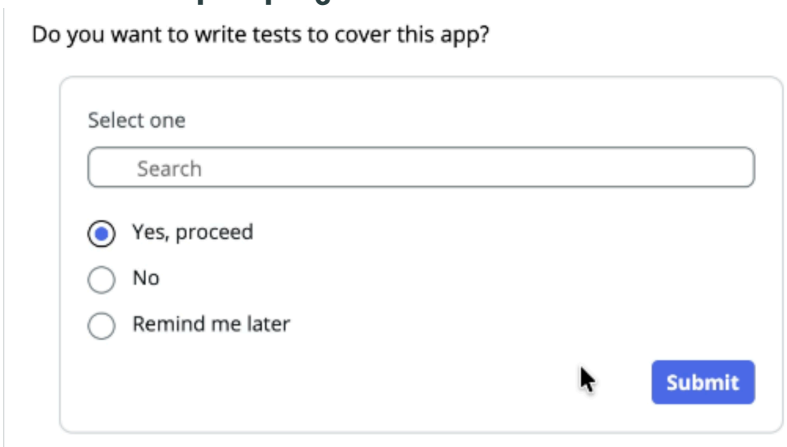
Test Agent delivers the following measurable outcomes:

- Build and test in one session: You no longer need to context-switch between authoring code and writing tests. Both happen sequentially within the same session.
- Faster failure triage: Automated RCA and proposed fixes reduce the time you spend looking through logs after a test run.
- More release confidence: Measurable quality gates enforced by automated test execution give you verifiable evidence of code health before promotion to production instances.
- Generated ATF tests are stored in the sys_atf_tests [sys_atf_tests] table under the app scope for which they were created. You can schedule regression test runs using the generated tests.

Test generation prompting

After each development action in Build Agent, Test Agent asks whether to generate ATF tests for what you just built. You can accept or decline the prompt at each step. When you accept the prompt, Test Agent generates tests before returning control to you in the chat panel.

Automatic test prompting



Automatic prompting for tests reduces the chance of skipping test coverage during rapid development by keeping test generation as part of the standard development loop.

To manually prompt for tests, ask Build Agent to `Generate ATF tests for all the feature permutations on the app we built`. Then, you can tell Build Agent to `Execute all ATF tests.`

Test Agent workflow

The end-to-end workflow is:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

1. Create or edit an app in a development instance using Build Agent in ServiceNow Studio or the ServiceNow IDE, driven by your prompt.
2. Alternatively, start from an Autonomous Engineer application plan. Test Agent reads the stories and acceptance criteria in the plan and generates tests automatically as implementation begins, without a separate prompt. For more information on running tests in Autonomous Engineer, see Autonomous Engineer in Build Agent.
3. Test Agent consumes the prompt and the resulting code changes to generate contextually relevant functional and UI ATF tests.
4. Respond to Build Agent asking whether you want to run the tests.
5. Failures are automatically triaged. Test Agent produces an RCA and either applies safe fixes autonomously or proposes them to you through the chat panel.
6. Build Agent ingests the RCA from Test Agent and re-executes tests until a passing status is achieved, completing the auto-heal loop. Stale tests are automatically updated to reflect the newest functionality.
7. Alternatively, ask Build Agent to create or run a test suite to execute multiple tests as a group and review consolidated results in the chat panel.

Key developer experiences for Test Agent in Build Agent

Autonomous test authoring

When you use Build Agent to implement a story, you can prompt it to create functional and UI tests automatically.

Test generation from Autonomous Engineer plans

When you use Autonomous Engineer to produce an application plan, Build Agent hands off that plan to Test Agent as implementation begins. Test Agent reads each story and its acceptance criteria in the plan and automatically generates Automated Test Framework tests to validate the expected behavior. Generated tests are executed immediately, and any failures are triaged by Test Agent before control returns to you in the chat panel. This closes the loop from planning to tested implementation within a single Build Agent session, without a separate prompt to create or run tests.

UI testing

Generate comprehensive UI tests for applications you build with Build Agent. UI testing extends the existing functional test capability to cover browser-level interactions, such as multi-step page navigation flows. Request a UI test by prompting Build Agent to generate a UI test for the application or flow you want to validate. For more information, see UI Test Script in Automated Test Framework (ATF).

List and related list step generation

Test Agent can generate ATF tests that use list and related list test steps, including validate related list visibility and apply filter to list tests. List step support extends test coverage beyond form-based interactions to include list view interactions on the ServiceNow AI Platform. For more information on list and related list test steps, see List and Related List.

Test suite authoring and execution

Create ATF test suites and edit existing ones from Build Agent. Group individual tests under a suite and execute the suite from the chat panel to run regression testing without selecting individual tests. Execution status and any errors are reported in the chat panel. For more information on test suites, see Building and running automated test suites.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Automatically updated tests

Test Agent automatically identifies outdated ATF tests as your application changes, and updates or removes them to keep your test suite aligned with your current code. Tests that no longer map to application artifacts are removed. Tests that partially match updated artifacts are revised to reflect the current implementation. Automatic test maintenance reduces the overhead of keeping ATF tests synchronized with ongoing development and removes the need to manually audit and update tests after each code change.

Auto-healing

Build Agent consumes the RCA produced by Test Agent and applies fixes to code or tests, then re-executes the test suite until all tests reach a passing status. This removes the need for developers to manually patch and maintain tests during a session.

Assisted troubleshooting

Test Agent automatically troubleshoots failed tests, generates RCAs, and proposes targeted fixes, eliminating manual log investigation.

Scope and availability

Test Agent is available in the following environments and scopes:

Test Agent scope and availability

Dimension	Supported values
Authoring environment	ServiceNow Studio, the ServiceNow IDE, and the ServiceNow SDK (execution tool only)
Application scope	Global, custom, store
Test types	ATF functional tests, UI tests, and test suitesATF functional and UI tests
Execution target	Cloud Runner lanes

Note: Test execution requires the ATF Test Generator and Cloud Runner app to be installed and a cloud user set up. For more information, see ATF Test Generator and Cloud Runner.

Document an application using Build Agent

Generate documentation for the structure, tables, and UI components of an application. Build Agent reads the codebase and creates a README file describing the application architecture.

Before you begin

Role required: admin

About this task

Build Agent processes the application's codebase and produces a README file summarizing the application's tables, sample data, and UI structure. Review the generated documentation for accuracy before sharing or publishing it. AI-generated results may not be accurate in all cases.

Procedure

1. Navigate to **All** > **App Development** > **ServiceNow Studio**.
   You can also open Build Agent in the ServiceNow IDE if you prefer a more code-centric experience.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

2. Enter a prompt requesting documentation for the application, such as `Document this app`, in the Build Agent chat box that appears on the ServiceNow Studio home page.

   You can also select the Conversations icon in the Navigator panel to open Build Agent.
3. Review the response in the Build Agent chat panel.
   Build Agent processes the existing files in the application and generates a README file.
4. In the File Explorer, open the README file that Build Agent creates.
   The README file displays a description of the application's tables, sample data, and UI page.

Result
The README file is available in the File Explorer for review and distribution.

Checkpoints and conversation change log

The conversation change log tracks every change Build Agent makes to your application. It appears automatically in an integrated tab in ServiceNow Studio and lets you view updates, roll back to a previous checkpoint, and deploy changes to an update set.

Conversation change log

Use the conversation change log to do the following actions:

- View the changes you have made since a checkpoint
- Roll back changes to the previous checkpoint
- Approve and deploy changes to an update set

When viewing the conversation change log, you can also view update sets created by Build Agent from within the chat panel. Each checkpoint includes a button that opens the relevant update set in a new tab. For information on how the change log and checkpoints work with update sets, see Update sets and Build Agent.

The change log displays the following:

- The **Files** list displays all the files that Build Agent updated during your conversation. The categories mirror the ones in the ServiceNow Studio Navigator panel.
- The menu at the top shows the current checkpoint. You can use the list at the end of the checkpoint name to select a different checkpoint to view. When there are multiple checkpoints, a **Summary** option appears, which shows changes from each checkpoint all together.
- The overview section shows the plan and actions Build Agent took based on the conversation.
- The action in the corner menu to pen the chat, **Restore** your app to a previous checkpoint, or open the app details page.

Changes from each checkpoint in the conversation are packaged together into an update set. You can find each update set on the **Deployment** tab on the ServiceNow Studio home page. For more information on deployment, see Deploying what you built with Build Agent.

Checkpoints

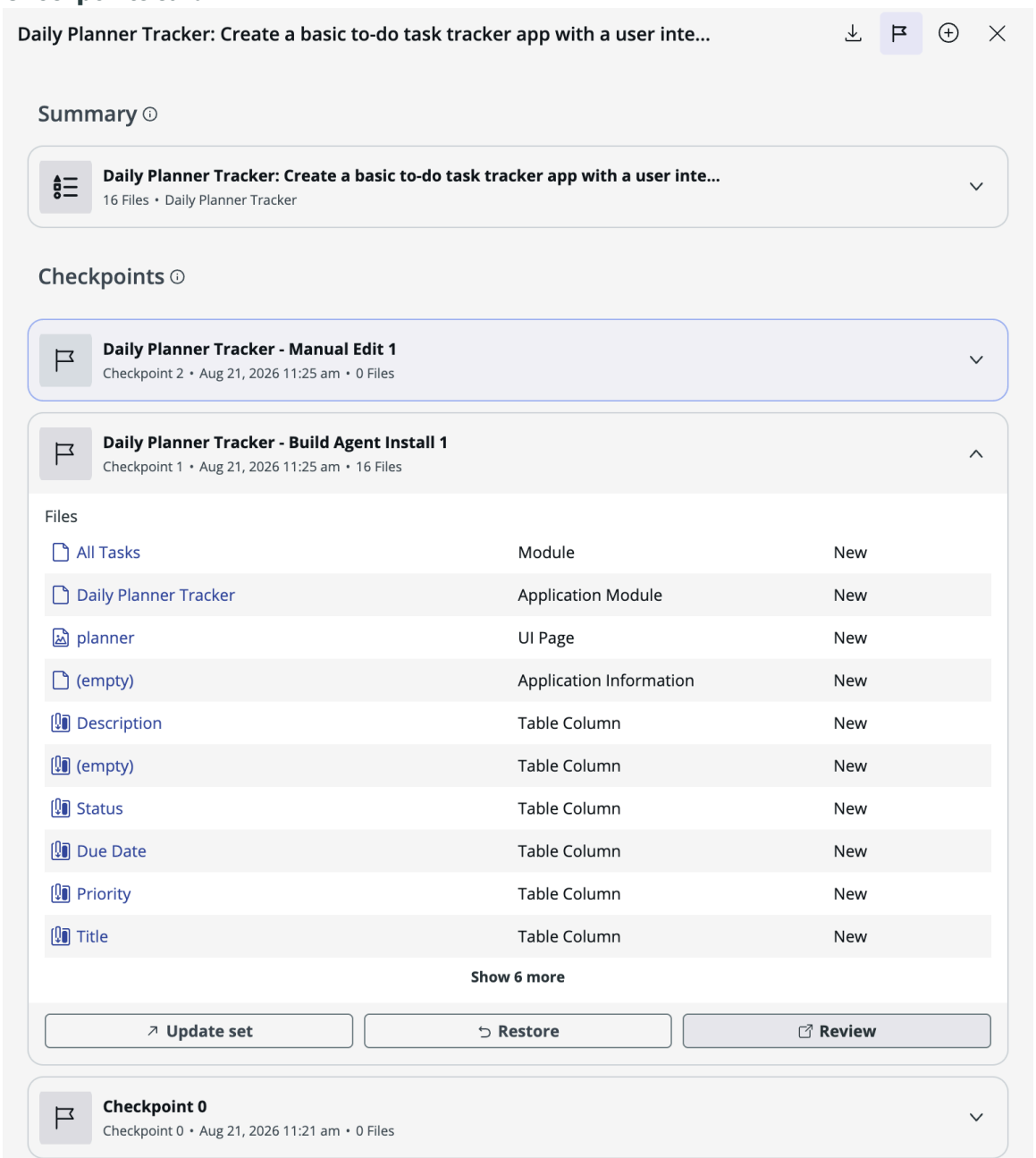
Checkpoints are groups of changes that you made to an application that you can revert to at any time during your conversation with Build Agent. Checkpoints are created automatically after you approve each task plan.

Note:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Checkpoint 0 creates an update set in global scope, and subsequent checkpoints created update sets in your application scope. However, update sets that exist across different scopes and can't be merged.
- Checkpoint 0 doesn't create an update set. Instead, it serves as a placeholder in the UI that you can use to restore your application to its initial state.
- Checkpoint 1 is the base update set for your application. The checkpoint 1 update set is the parent, or chain root, for all subsequent update sets created during the conversation. Because all update sets exist in the same application scope, you can merge them together when you're ready to move your changes to another environment. For information on update sets, see Update sets and Build Agent.

Checkpoints card



When Build Agent prepares to create the next checkpoint, it checks whether any changes exist in the manual edit update set. If no changes are present, Build Agent removes the update set. If changes exist, Build Agent completes the update set and renders it with the other update sets

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

from your conversation. You can access manual edit update sets directly from the checkpoints panel in your Build Agent conversation.

Update sets and Build Agent

When you work with Build Agent, your changes are automatically tracked in update sets so you can review, revert, and deploy them without leaving ServiceNow Studio.

Build Agent tracks every change it makes to your application in update sets. Changes from each checkpoint in a conversation are captured together in a single update set. You can access, review, and open the update sets directly from the Build Agent chat panel. You can also open the update sets from the Current Changes List (CCL) page in ServiceNow Studio, without navigating to the platform.

Update sets use descriptive names to help identify what each update set contains. Names follow this pattern: *`application-name`* `build agent install 1`, `build agent install 2`, and so on.

For general information about update sets on the ServiceNow AI Platform, see System update sets.

How Build Agent tracks changes

Build Agent automatically captures changes in a unified update set as you work. Manual edits you make in the ServiceNow Studio editor and automatic edits made by Build Agent within the same work session are bundled into a single update set. The consolidated update set merges all the update sets created for previous agentic and manual development into a single update set for ease of deployment. The consolidation occurs after you tell Build Agent that you're done working on a task and no longer require rollback.

Build Agent automatically captures changes to the app and metadata that you're working on.

A checkpoint is created automatically after you approve each task plan. When Build Agent reaches a checkpoint, the changes associated with that checkpoint are captured in the update set. You can view and open the relevant update set directly from each checkpoint in the chat panel. For more information on checkpoints, see Checkpoints and conversation change log.

When Build Agent prepares to create the next checkpoint, it checks whether any changes exist in the manual edit update set. If no changes are present, Build Agent removes the update set. If changes exist, Build Agent completes the update set and renders it with the other update sets from your conversation. You can access manual edit update sets directly from the checkpoints panel in your Build Agent conversation.

Access update sets from Build Agent

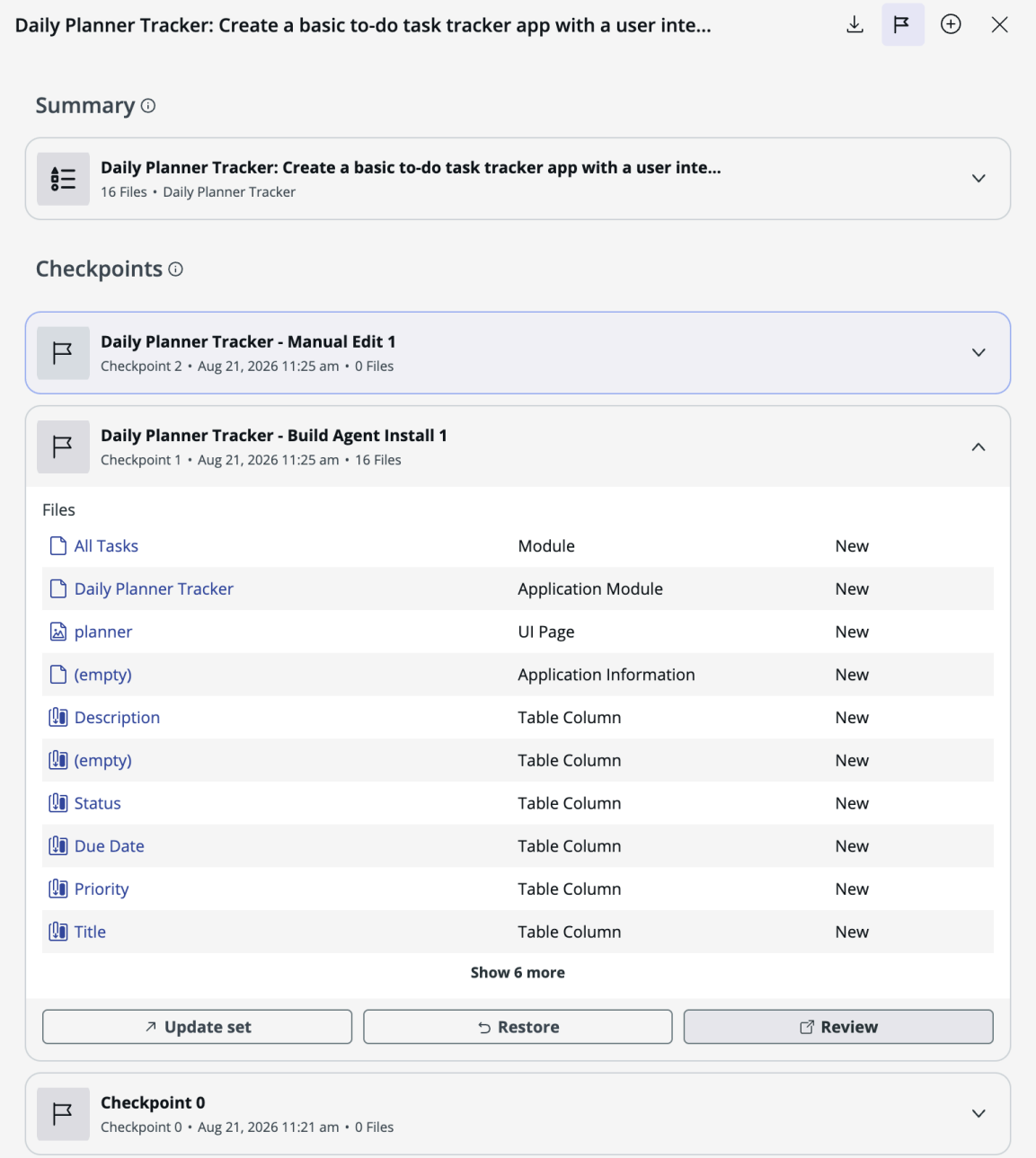
You can access update sets created during a Build Agent session from two locations in ServiceNow Studio.

Chat panel

Each checkpoint in the chat panel includes a **Review** button that opens the relevant update set in a new ServiceNow Studio tab. The update set button in the summary card opens all update sets associated with the changes shown in that card.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Summary card



Current Changes List page

The Current Changes List (CCL) page in ServiceNow Studio shows Build Agent edits. Each checkpoint on the CCL page includes a button that opens the relevant update set in a new tab.

Revert changes using checkpoints

You can revert your application to any previous checkpoint during a Build Agent session. For more information about checkpoints and how to restore a previous state, see Checkpoints and conversation change log.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Deploy update sets

After your changes are ready, you can find the update sets from your Build Agent session on the **Deployment** tab on the ServiceNow Studio home page. For more information about deploying your changes, see Deploying what you built with Build Agent.

Deploying what you built with Build Agent

Learn about deployment methods and workflows for moving applications created with Build Agent from development to production environments. Choose the right deployment approach based on your application complexity and organizational requirements.

Workflow for deployment

After development, review, and testing are complete, a typical deployment workflow includes the following steps:

1. Collaborative design: Business owners and IT collaborate on requirements and ideas using their preferred tools.
2. AI-driven app development: Build Agent, Autonomous Engineer and ServiceNow Otto process files, chat history, and diagrams to generate and implement app updates.
3. Review and testing: Teams preview updates, make revisions, and run rounds of performance and readiness testing.
4. Developer review: A developer reviews the AI-generated changes, compares versions, and confirms the changes are ready for deployment.
5. Deployment approval: The project is handed off to a deployment manager, who initiates the deployment approval process.
6. Autonomous checks: AI agents automatically scan for issues (such as sensitive data exposure or model integrity problems) and remediate them before deployment.
7. Final deployment: After all readiness scans and approvals, the new app is deployed securely and efficiently.

Important: Changes are isolated until you explicitly deploy and install the application. You can modify and iterate without affecting the live running application.

The way isolation works depends on the environment:

- In the ServiceNow IDE, isolation runs through source and build artifacts until you deploy and install.
- In ServiceNow Studio, changes are tracked in update sets and promoted when you move the update set between instances.

Deployment methods for Build Agent

Build Agent and Autonomous Engineer support the following deployment methods for apps created and edited with agentic development:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Git-based source control integration: ServiceNow supports Git-based workflows for version control and CI/CD.
  - You can push scoped apps to Git repositories, enabling branching, merging, and automated deployments. ServiceNow supports bring-your-own Git integration, such as GitHub or Bitbucket.
  - You must be on Brazil Patch 1 to use source control in ServiceNow Studio.
  - For more information, see Integrating source control with the ServiceNow IDE and Fluent source control in ServiceNow Studio.
- Update sets and application packaging: Standard ServiceNow deployment uses System Update Sets to track changes.
  - Advanced guidance includes packing update sets into scoped applications for easier transport and installation across instances, for example using Application Repository (AppRepo).
  - For information on update sets for Build Agent, see Update sets and Build Agent.
  - For information on update sets for Autonomous Engineer, see Update sets and Autonomous Engineer.
  - For more information on System Update Sets, see System update sets.

Options for moving apps through instances

After you create an app using Build Agent or Autonomous Engineer, you have several options to move the app to the test instance.

1. Wrap the entire scoped application in an update set. The workflow is as follows:
   a. Go to the **Custom Applications** list, select an app and swap to its scope.
   b. Convert the app to AppRepo.
   c. Publish the update set with demo data.
   d. Put the update set in a deployment request for ReleaseOps, or follow your standard update set process for deployment.
2. Publish the app to AppRepo:
   - You can use a Git-based process or update sets to publish to AppRepo.
   - Scoped apps and apps that are ready for testing can be published to the AppRepo for distribution across environments.
   - After an app is in AppRepo, you can move it through a ReleaseOps pipeline. If ATF tests are included in the pipeline, they automatically run.
   - Register and entitle apps before publishing.
   - For more information on Application Repository, see ServiceNow application repository.

If your app is connected to Git-based source control, you can also push changes to a remote repository directly from the Build Agent chat panel. For more information, see #unique_89.

Additional deployment tools

The ServiceNow AI Platform has additional deployment tools that include the following tools:

- App Engine Management Center (AEMC):
  - After developing an app, submit it to AEMC for governance checks.
  - AEMC validates ACLs, roles, and compliance settings before deployment.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Use ReleaseOps pipelines to move apps through environments with ATF tests and approval gates.
- AEMC provides dashboards for monitoring deployments and managing app versions throughout the lifecycle.
- For more information on AEMC, see Using the App Engine Management Center.

- ReleaseOps:
  - Move changes from development to production through multiple instances using customizable playbooks.
  - Automate preview, commit, and validation of update sets before deployment.
  - Run Automated Test Framework (ATF) tests as part of the pipeline to validate quality.
  - Deploy changes immediately or schedule releases for controlled rollouts.
  - Enforce checks, scans, and approvals before production deployment.
  - For more information on ReleaseOps, see ReleaseOps.
- ServiceNow SDK:
  - Use the ServiceNow SDK to move applications to and from your instance to your local machine. You can integrate the ServiceNow SDK with your off-instance CI/CD process if you have one.
  - Install the ServiceNow SDK locally and use the command line interface (CLI).
  - Authenticate to a ServiceNow instance from the ServiceNow SDK.
  - Push to or install an application on the authenticated instance from your local environment.
- Automated Test Framework (ATF)
  - Tests can be generated by Build Agent and executed in ServiceNow Studio or ServiceNow IDE to confirm functionality after changes.
  - For more information on ATF, see Automated Test Framework (ATF).

Mixing application repository and update set deployment

For any given application on a given instance, the application repository deployment path and the update set deployment path are mutually exclusive. After you deploy an application to an instance using the application repository, you can't switch to update sets for that application on that instance, and the reverse is also true.

Note: You can include global apps in the Application Repository and source control.

For more information, see the Can't mix Update Set and App Repo deployment for the same application [KB0715422] article in the Knowledge Base.

Issues and solutions in Build Agent

Identify and resolve common issues in Build Agent.

General debugging approach

Copy the error message and paste it into the Build Agent chat. Build Agent can often diagnose and fix the issue directly. If not, the error message is your starting point for debugging.

If several fixes in the same conversation don't resolve the issue, use a structured approach before starting a new chat.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Tip:

Use the following prompt to structure your debugging session:

```
I'm seeing this error: [paste exact error]. This happens
when: [describe scenario]. Before suggesting fixes: 1) Help me
reproduce this consistently. 2) Review the debugger output and
logs. 3) Check for access or configuration issues. 4) Examine
if this is a known issue.
```

Test hypotheses from simplest to most complex:

- Is this an access or role issue?
- Is this a configuration problem?
- Is this a naming convention mismatch?
- Is this a logic edge case?

If structured investigation doesn't resolve the issue within a few attempts, start a fresh conversation with a more specific prompt.

Save any debugging information to convey if you need to contact Now Support.

Issues and solutions

Build Agent issues

Issue	Resolution
ServiceNow Otto panel not visible	• Quick check: Required plugins are installed and your account has the required roles. The ServiceNow IDE needs the admin role plus the documented Build Agent plugins (`sn_glider` and `sn_build_agent` for Trial; `sn_now_creator` for Premium). In ServiceNow Studio, confirm you're on a supported release. • Try: Refresh the page or sign out and back in. • If still failing, capture: The exact error or behavior, your release, and the role list on your user.
Build Agent unresponsive	• Quick check: Browser cache and the active session. • Try: Clear the cache, try a different browser, and check the ServiceNow status page for service announcements. • If still failing, capture: The request that hung, your browser version, and the ServiceNow status state.
Scope or name conflict	• Quick check: Whether the scope or app name is already in use on the instance. • Try: Let Build Agent self-correct with a unique scope, or pick a new name and try again. • If still failing, capture: The conflicting scope or app name and the action that triggered it.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Build Agent issues (continued)

Issue	Resolution
Stuck in error loop	• Quick check: How many fixes Build Agent has attempted on the same symptom. • Try: Use Git checkpoints or update set rollback to revert, then start a fresh conversation that names the specific error. • If still failing, capture: The exact error text, recent commits, and a brief description of what you were trying to accomplish.
App not deploying	• Quick check: Build output for errors before deploying. Build Agent only deploys the latest successful build. If your updates are not available after deployment, the build may have failed even though the deployment succeeded. • Try: In the ServiceNow IDE, run ServiceNow SDK build first, check errors, then deploy and install in sequence. In ServiceNow Studio, confirm your update set is current and complete. Paste any error messages into the chat panel and ask Build Agent to resolve the build issues and redeploy. For more information, see Deploying what you built with Build Agent. • If still failing, capture: The build log, your SDK version, and your update set state.
Empty UI page	• Quick check: The browser console for errors. • Try: Open the browser console, copy the error messages, paste them into the chat panel, and ask Build Agent to fix the UI page. • If still failing, capture: The full browser console output and the prompt that produced the empty page.
Context limit exceeded	• Quick check: Whether you have received a `Context Window Exceeded` error. Build Agent maintains a history of your session and can resume from the point of failure, but has a context limit. • Try: Open a new chat window and describe the application again. Build Agent does not retain context from the previous session, but it reviews the work already done and continues from where it stopped. • If still failing, capture: The last prompt before the error and a summary of what had been built.
Out of prompts	• Quick check: Your remaining prompt count for the current 30-day cycle. • Try: Wait for the cycle to reset, or upgrade to ServiceNow Otto for Creator. Plan approvals do not count against your prompt allotment; only submitted prompts do. • If still failing, capture: Your entitlement state and the cycle reset date.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Build Agent issues (continued)

Issue	Resolution
Rate limit error	• Quick check: Whether the model provider is receiving too many requests. • Try: Wait a minute or two before trying again. • If still failing, capture: The exact error message and the time it occurred.
Unsupported configuration	• Quick check: Whether the requested metadata or feature is in the supported list. • Try: Complete that step directly on the platform, then continue with Build Agent for everything else. • If still failing, capture: The configuration you needed and the workaround you used.
Missing dependencies (ServiceNow IDE)	• Quick check: `package.json` for the entries Build Agent referenced. • Try: Let Build Agent self-correct first. If that does not resolve it, add the package manually and run `npm install`. • If still failing, capture: The missing module name and the install error output.
Long-running prompt times out	• Quick check: Your session timeout setting. • Try: For labs and workshops, set *`glide.ui.session_timeout`* to `1440` in the System Property [sys_properties] table. Coordinate with the platform admin on a shared instance before changing it. • If still failing, capture: The session length and the prompt that timed out.

Build Agent reference

Reference topics provide additional information about using Build Agent to create and edit apps, as well as ServiceNow metadata.

Reference topics provide metadata compatibility information and example prompts to support your development workflow. Use the following topics alongside the task and concept documentation to get the most out of Build Agent.

Supported metadata in Build Agent

Metadata and app file types that Build Agent can create and manage.

For metadata types not listed in this table, query Build Agent directly.

Important: Build Agent and Autonomous Engineer only create metadata supported by ServiceNow® Fluent. For more information, see ServiceNow Fluent. For the latest API reference, see https://servicenow.github.io/sdk/

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Supported metadata

Supported metadata types

Metadata type	Description
Automated Test Framework (ATF) tests	Automated test cases for validating application behavior.
Application menus and modules	Application menus are the top-level categories in the application navigator sidebar. Modules are the individual links within those categories and can link to tables, URLs, lists, or other platform resources. When Build Agent or Autonomous Engineer creates a new application with tables, it generates the corresponding navigator structure so users can access the application from the sidebar.
Assessments	Assessment instances, metric types, and related configurations for evaluating records against defined criteria. For more information, see Assessments.
Assignment rules	Rules that automatically assign records to users or groups based on field conditions.
Business rules	Server-side scripts that run when records are displayed, inserted, updated, or deleted.
Choice lists	Field value options for select fields on forms and tables.
Client scripts	Scripts that run in the browser to control form behavior.
Client-side logic	Browser-based logic for controlling UI interactions and form behavior.
Condition builder and query conditions	Filter conditions used in business rules, workflows, and list views.
Configuration	System and application configuration settings.
Connection and credential alias	External system connections and stored credentials for authenticating with third-party services and platforms.
Custom AI agents	AI agents scoped to application data models, roles, and ACLs. For more information, see Create agentic workflows, agents, and skills.
Database views	Database view definitions (`sys_db_view`) and database view table configurations (`sys_db_view_table`) that combine fields from multiple tables into a single queryable view.
Data policies	Rules that enforce field-level data requirements, such as mandatory or read-only constraints, across all interfaces including the API.
Data import pipeline	Build Agent creates the full data import pipeline: data sources, staging tables, import sets, and transform maps. Supported source types include CSV, Excel, JSON, XML, JDBC, LDAP, and REST. Specify the external data source, target table, and field mapping, and Build Agent or Autonomous Engineer generates the pipeline artifacts. The capability covers both one-time bulk loads and recurring scheduled imports.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Supported metadata types (continued)

Metadata type	Description
Data lookup	Metadata definitions for data lookup fields that retrieve values from related tables or external sources.
Dictionary overrides	Field-level customizations that override base system dictionary entries without modifying the original.
Event registry configurations	Event definitions that trigger notifications, scripts, or other platform actions.
Field-level overrides	Attribute overrides applied at the field level on extended tables.
Flows	Automated processes built in Workflow Studio using a no-code or low-code interface.
Forms	Record views that define which fields are displayed and how they are arranged.
Inbound email actions	Rules that process incoming email and create or update records based on message content.
JavaScript modules	Build Agent or Autonomous Engineer creates JavaScript modules for organizing reusable server-side code within an application. Modules support standard import and export patterns and can reference server-side APIs. Use this pattern when multiple business rules, script includes, or other server-side scripts must share utility functions, constants, or logic without duplicating code.
Knowledge base access	Configurations that control which users and groups can read, contribute to, or manage knowledge bases and knowledge articles.
LDAP server configurations	Build Agent or Autonomous Engineer creates LDAP server configurations and LDAP server URLs for connecting to external directory services such as Active Directory and OpenLDAP. LDAP server configurations include failover and load-balancing URL configurations and SSL settings. To import records from the directory into ServiceNow tables, use a data import pipeline instead.
List controls	Settings that control list view behavior, including column display and row counts.
Playbooks	Representations of cross-enterprise business processes that organize tasks and activities into logical stages to guide users through a record lifecycle. Playbooks combine triggers that specify when to start, stages that group sequences of activities, and activities that define the automation and user-facing experience. Build Agent or Autonomous Engineer can create playbook configurations and activity definitions to help organizations digitize and standardize their business processes. To generate a playbook in Build Agent or Autonomous Engineer, attach a file, such as an image, an XML file, or a text description. • For details on playbooks support in Build Agent and Autonomous Engineer, see Playbook authoring with Build Agent and Autonomous Engineer. • For more information on playbooks, see Workflow Studio playbooks.
Record insertion	Build Agent or Autonomous Engineer inserts data into any table, including tables that don't have a dedicated build skill. The functionality covers seed data, demo data, reference data, and configuration records. The same

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Supported metadata types (continued)

Metadata type	Description
	approach applies to creating metadata records on platform tables that Build Agent or Autonomous Engineer don't have a specialized skill for yet.
REST message and HTTP method	REST message definitions and HTTP method configurations for building API integrations and outbound integrations.
Scheduled jobs	Automated tasks that run on a defined schedule.
Script includes	Reusable server-side JavaScript libraries callable from other scripts.
Scripted REST APIs	Custom REST endpoints built with server-side scripting.
Security	Access control lists (ACLs), roles, and related security configurations.
Server-side logic	Scripts and business rules that execute on the server.
Service Catalog items and configurations	Catalog items, variables, and fulfillment flows for the Service Catalog. Includes `dependentQuestion` support for variables, enabling dynamic default values that auto-populate based on a referenced variable's record. For example, when a user changes the **Requested for** field, related fields such as **Department** and **Location** can be populated automatically.
Service Portal configuration, pages, and widgets	Portal structure, page definitions, and custom widgets.
Skills	ServiceNow Otto skills for AI-powered responses and actions.
Sys wizard answers	Answer options for platform interceptors stored in the `sys_wizard_answer` table. Interceptors are the choice dialogs that route users to different destinations when they navigate to a form or module. Each answer record defines a label and the target destination for one choice.
Tables	Database tables that store application data, including fields, relationships, and access controls.
Transition conditions	Conditions that determine when a record can move from one state to another in a workflow or flow, enforcing stage-gate logic and data requirements.
UI actions	Buttons, context menu items, and links that trigger scripts or navigation on forms and lists.
UI components	Reusable interface elements built with UI Builder.
UI pages	Custom pages built outside of standard form and list views.
UI styles	Style definitions stored in the `sys_ui_style` table that control the visual appearance of forms and fields.
UI policies	Client-side rules that show, hide, require, or lock fields based on conditions.
UI views	Named configurations of form layouts for different user contexts.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Supported metadata types (continued)

Metadata type	Description
User criteria	Rules and criteria definitions for filtering and selecting users based on field conditions and organizational structures.
Workflows	Automated process sequences built in the legacy workflow editor.
Workspaces	Agent-facing interfaces built with configurable workspace components.

Example prompts

Explore example prompts for building apps, as well as adding governance, UI and other ServiceNow metadata to help you get started with prompting Build Agent.

To learn more about prompting, see this Community article on The fastest way to learn Build Agent prompting? Ask Build Agent.

Prompting overview and tips

Prompts are counted each time you submit a message to Build Agent. If Build Agent asks a clarifying question and you respond, that response counts as a prompt. Approving a plan that Build Agent presents does not count as a prompt. To get the most value from each prompt, draft your message in a text editor before you submit it.

For more guidelines on prompting, see General guidelines for Build Agent.

Prompt reference

Use the following prompts as starting points for common Build Agent goals. Replace the bracketed placeholders with your specific details.

Prompt reference by goal

Goal	Prompt
Plan	`I want to build [app idea]. Create a detailed plan including tables, fields, roles, and business rules. Wait for my approval before creating any files.`
Prototype	`Build me a [app name] application with [key requirements]. Include basic roles, flows, and a working UI.`
Brainstorm	`I need to build something for [business problem]. What tables, flows, and catalog items would I need?`
Discover	`List the metadata types you can create or edit in this scope.`
Coach	`For [metadata type], coach me on an ideal prompt and common pitfalls.`
Extend	`Open [app name]. Add a field called [field] to the [table] table. Update the ACLs so only [role] can edit it.`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Prompt reference by goal (continued)

Goal	Prompt
Extend out-of-box	`Add a custom field to the incident table called [field]. Create a business rule that [logic]. Show the plan before applying.`
Learn	`Explain [ServiceNow concept] to me. What is it, when do I use it, and show me an example.`
Debug	`I'm seeing this error: [paste exact error]. This happens when: [describe scenario]. Before suggesting fixes: 1) Help me reproduce this consistently. 2) Review the debugger output and logs. 3) Check for access or configuration issues. 4) Examine if this is a known issue.`
Document	`Create a README with setup instructions and a user guide for this application.`
Test	`Create ATF tests to validate [specific functionality] works correctly.`
Security	`Add role-based access so only [role] can edit; others are read-only. Show the plan and ACL changes before applying.`
Diagram	`Generate a Mermaid ER diagram showing all tables, fields, and relationships in this application.`
Flow	`Create a flow that triggers when [condition] and sends an email notification to [recipient] with [details].`
Ideal prompt	`Now that you've built this app, what's the ideal prompt to recreate it from scratch?`

Tip: UI policies handle the common cases declaratively: show, hide, require, or lock fields based on conditions. For example, "when Category is Hardware, show the Asset Tag field and make it required" produces a UI policy that applies the rule without scripting.

Get started with prompting

To get started with Build Agent, try one of the following approaches.

- Prototype a simple app: Describe a straightforward application in one or two sentences. Review the plan that Build Agent presents, then approve it to start the build. This approach lets you learn the workflow with low risk.
- Document an existing app: Open a custom scoped app and ask Build Agent to summarize its architecture, data model, roles, and key components. This produces useful documentation with no risk to the app.
- Extend an existing app or base system table: Add a field, a business rule, or an ACL to an app you're familiar with. Build Agent supports global-scope tables such as Incident, as well as custom and ServiceNow Store apps.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Starter prompts

Use the following prompt patterns to get started with common Build Agent tasks. Replace the bracketed placeholders with your specific details.

Extend safely

```
Open [app name]. Before making changes, document the current state. Then add [feature]. Return a plan for approval. After changes, generate ATF tests and a README.
```

Prototype

```
Build me a [app name] application with [requirements]. Include basic roles, a working UI, and flows for [automation]. Then generate ATF tests and documentation.
```

Extend base system

```
Add a custom field called [field] to the incident table. Create a business rule that [logic]. Show the plan and ACL changes before applying.
```

Document

```
Summarize the app architecture, data model, roles, and key UI components. Don't make any changes.
```

Additional prompt suggestions

After you have the foundation of an app created, save your session and keep prompting to refine and extend your app. For example, you can create Automated Test Framework (ATF) tests, generate release notes, and prepare summaries. Some additional prompt ideas include the following examples:

- Improve generated user interfaces
- Create ATF tests
- Create QA and UAT test plans
- Create a summary of all application functionality
- Add additional industry-specific use cases
- Create an announcement
- Create a user guide
- Generate release notes
- Draft change management plans
- Create training materials for end users
- Build data migration plans
- Generate workflow optimization suggestions
- Prepare executive summary for stakeholders

Example Build Agent prompt to build an app

Prompt: `Create an application that handles several types of issues related to cash management. Create a custom data model with tables for each type of request. Add fields that are typical for these types of issues to each table. Request tables should extend`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

the Task table. Create custom states for each request type to track issue resolution.

Example Build Agent prompt to add security to an app

Prompt: Create a new role and define permissions on the request tables for two types of users:

1. Requesters who submit issues and can only view their own requests.
2. Fulfillers who can view and edit all requests.

Example Build Agent prompt to add business rules to an app

Prompt: Create business rules for each request that is triggered by state changes in the workflow. Design the business rules to work together to create business processes aligned to industry best practices for each type of issue.

Example Build Agent prompt to generate different UIs for an app

Prompt 1: Create an easy to use UI for users with the cash management fulfiller role to view all issues, update and complete them.

Prompt 2: Create a UI for cash management requesters to submit issues and view or edit issues that they have previously created.

Example prompts for governance

Prompting with governance requirements helps produce secure and compliant applications on the ServiceNow AI Platform. By embedding governance requirements directly into your prompts, you help the AI produce results aligned with your organizational standards for security, compliance, and quality.

For more information on governance, see Build Agent governance.

Tip: If you have explicit security requirements, include them in your initial prompt.

Example: Prompt to include security roles and ACLs

Create an Employee Onboarding app with HR and IT roles, enforce ACLs so only HR can edit personal data.

Example: Specify data sensitivity

Generate a table for medical records with encryption enabled and restrict access to compliance officers.

Example: Request audit and logging

Build a workflow for expense approvals and include audit logging for all status changes.

Example: Mention testing requirements

Create a catalog item for laptop requests and generate ATF tests for submission and approval flows.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example: Call out compliance standards

```
Develop a leave management app that meets GDPR requirements and includes consent tracking.
```

Example: Ask for documentation

```
Generate a summary of the app and flow logic for governance review.
```

Supported file types for Build Agent

Build Agent accepts images, documents, code files, and file types specific to ServiceNow as uploads. Supported formats, size limits, and general guidelines apply when generating apps and metadata.

Supported file types

Category	Extensions
Code and scripts	.js, .ts, .tsx, .jsx, .py, .java, .css, .html, .xml, .json
Documents	.txt, .md, .csv, .log
Images	.png, .jpg, .jpeg, .gif, .svg, .webp
Specific to ServiceNow	.now.ts (Fluent DSL), .xml (update sets)

You can also use MCP server connections, such as for Figma, to provide context for Build Agent. For more information, see MCP connections and Build Agent.

Size limits

Size limits

Category	Description
Individual file	Up to 10 MB per attachment.
Images	View and analyze screenshots and diagrams directly in the chat.

File upload guidelines

- Upload text-based files to let Build Agent read and analyze the full content, then incorporate it into your app.
- Upload images or screenshots to provide UI mockups, error screenshots, or architecture diagrams as context for development.
- If an attachment is not visible after a window reload, re-upload the file.

File uploads and your project

Uploaded files are available only within the current conversation and aren't added to your ServiceNow project or application automatically.

Upload scenarios

Upload	What Build Agent can do
A script file (.js, .ts)	Read the file, then write the content into your app using the appropriate source file tool.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Upload scenarios (continued)

Upload	What Build Agent can do
A screenshot of an error	Analyze the screenshot and help resolve the issue.
A CSV or data file	Use the data to inform table design or generate test data.
A mockup or diagram	Use the file as a reference to build a UI page.
An XML update set	Analyze the file and recreate the metadata using Fluent APIs.

Note:

- Uploading a file does not add it to your app. To incorporate file content into your project, ask Build Agent to do so explicitly, for example, "Add this script to my app as a script include."
- Uploaded files don't persist across sessions. If the window reloads, re-upload the file for Build Agent to access the content again.

Domain separation and Build Agent

Domain separation is supported for Build Agent. Domain separation enables you to separate data, processes, and administrative tasks into logical groupings called domains. You can control several aspects of this separation, including which users can see and access data.

Build Agent creates standard ServiceNow metadata, such as tables, business rules, and access control lists, that participates in domain separation when domain separation is active on the instance. Domain separation is a platform-level feature configured by a ServiceNow AI Platform administrator, not by Build Agent.

Build Agent creates scoped applications with their own namespace. When domain separation is active, the platform manages domain visibility of application data automatically through the `sys_domain` field.

How domain separation works in Build Agent

Domain separation configuration occurs at two layers.

Instance level

A platform administrator activates the Domain Support - Domain Extensions plugin (com.glide.domain.msp_extensions) and configures the domain hierarchy through Domain Admin. The configuration is outside the scope of Build Agent.

Application level

Build Agent creates the application metadata that operates within a domain-separated environment. After you install the application, a platform administrator configures domain separation settings for any new tables and assigns roles to the appropriate domains.

ServiceNow Fluent SDK level

The ServiceNow Fluent SDK supports setting the `sys_domain` field directly on records and on APIs that support domain separation. You can also read and write `sys_override` fields to apply domain-specific overrides through the ServiceNow SDK. In addition to tables, you can create other metadata objects in specific domains.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Build Agent supports creating the following in a domain-separated environment.

- Tables that inherit domain separation when they extend a domain-separated parent table
- Business rules and scripts that run within the domain context of the current session
- Fluent SDK records and APIs that set the `sys_domain` field and use `sys_override` fields in domain-separated environments

The following general guidelines apply to data model behavior when working with Build Agent:

- When Build Agent creates a table that extends a base system table with `sys_domain`, for example `task` or `cmdb_ci`, the domain column is automatically inherited. Don't redeclare it.
- GlideRecord queries in scripts that Build Agent creates are automatically filtered by domain. Queries that must run across domains require elevated privileges configured by a platform administrator.
- Reference fields that point to a domain-separated table display only records visible in the current domain.

Roles and responsibilities

The following roles and responsibilities apply when administering domain separation for applications that Build Agent creates.

Instance owner

- Controls which domains can access the application, configures domain-default field values.
- Manages domain-specific overrides for business rules.
- Determines which business logic applies globally vs. per domain.

Tenant domain administrator

- Creates, updates, and deletes records within their domain and any child domains.
- Configures domain-specific values such as categories, assignment groups, and approval workflows.
- Can't modify application metadata including business rules, access control lists, or table structure unless the application explicitly permits it.

Special conditions

Build Agent domain separation support includes the following conditions and exceptions.

- Build Agent creates application metadata but can't configure domain separation settings directly. A ServiceNow AI Platform administrator must configure domain hierarchy, table participation, and visibility rules on the instance.
- Flows that Build Agent creates execute in the domain context of the triggering record. Flows that must operate across domains, such as an escalation from a tenant domain to a global domain, require additional design and platform administrator configuration.
- Metadata created in global scope is visible across all domains by default. Scoped applications provide better domain isolation.
- When using the Fluent SDK, you can set `sys_domain` on records and APIs that support domain separation. The `sys_override` field is also supported, which lets you apply domain-specific field value overrides through the SDK without modifying the base record.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Related topics

Domain separation for service providers

Developer Sandboxes

Developer Sandboxes provide admins and delegated developers the ability to request, access, and manage individual sandboxes on top of the same underlying development instance. Delegated developers can write and merge code and configuration changes without the risk of their changes getting overwritten on the instance mid-development.

https://player.vimeo.com/video/1179947416?h=848548c93a&badge=0&autopause=0&player_id=0&app_id=58479

Get started

Explore	Install	Administer
		
Explore Developer Sandboxes	Install Developer Sandboxes	Administer and allocate Developer Sandboxes

Use Developer Sandboxes to create isolated development environments where users can develop on the ServiceNow AI Platform in parallel. Sandboxes help enhance Git-based workflows of the software development life cycle by reducing conflicts and enabling faster time to production.

Effective management of Developer Sandboxes enables companies to do the following:

- Provision isolated development environments on demand
- Align workspaces to specific stories, developers, test plans, or any custom criteria
- Enable true parallel development with controlled, independent environments

Once allocated, developers can access their sandbox by one of two ways:

- Selecting the sandbox name on the Sandbox Management Home page.
- Using the context menu and copy the URL to the clipboard and paste it to the browser.

Note: If your instance is in a regulated environment, check with ServiceNow Support for more information about support for Developer Sandboxes.

Troubleshoot and get help

- Ask questions and explore other resources for Developer Sandboxes in the ServiceNow Community
- Search the Known Error Portal for known error articles
- See the Developer Sandboxes FAQ and getting started guide
- Take the Developer Sandboxes Basics class on ServiceNow University

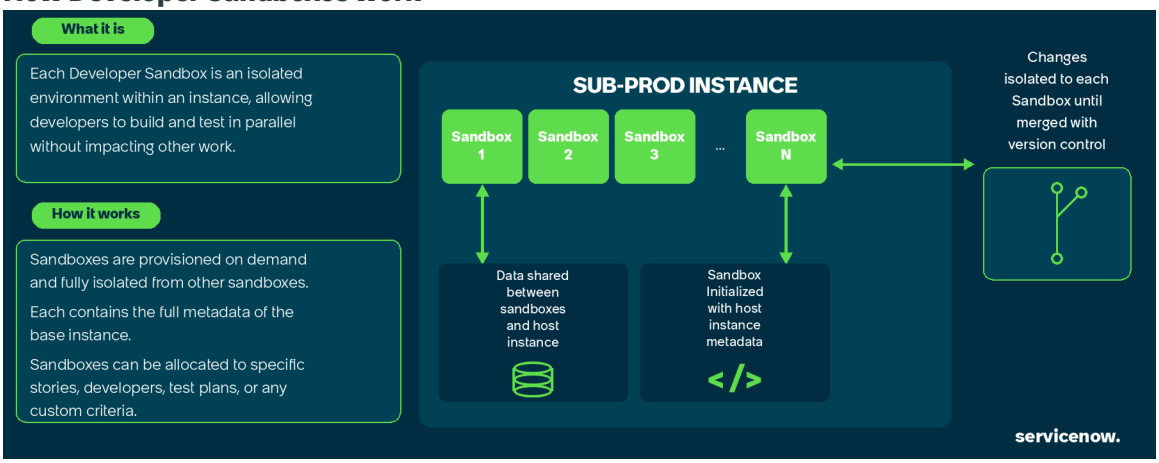
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Read a blog about Developer Sandboxes
- Watch a video demo of merging changes in Git
- Contact Customer Service and Support

Exploring Developer Sandboxes

Developer Sandboxes provide isolated development environments that enable parallel building and testing on top of a shared development instance. Use sandboxes to reduce code conflicts, accelerate delivery, and safely test configurations without affecting other team members' work.

How Developer Sandboxes work



- Developer Sandboxes are isolated environments for parallel building and testing.
- Each sandbox is provisioned on demand and fully isolated from others.
- Every sandbox contains the full metadata of the base instance.
- Sandboxes can be assigned to specific stories, developers, test plans, or custom criteria.

Developer Sandboxes overview

Developer Sandboxes aim to provide lower-cost developer isolation and parallelism for customer development environments and instances. Developer Sandboxes are workflow-agnostic and are broadly applicable across workflows for both smaller and larger companies.

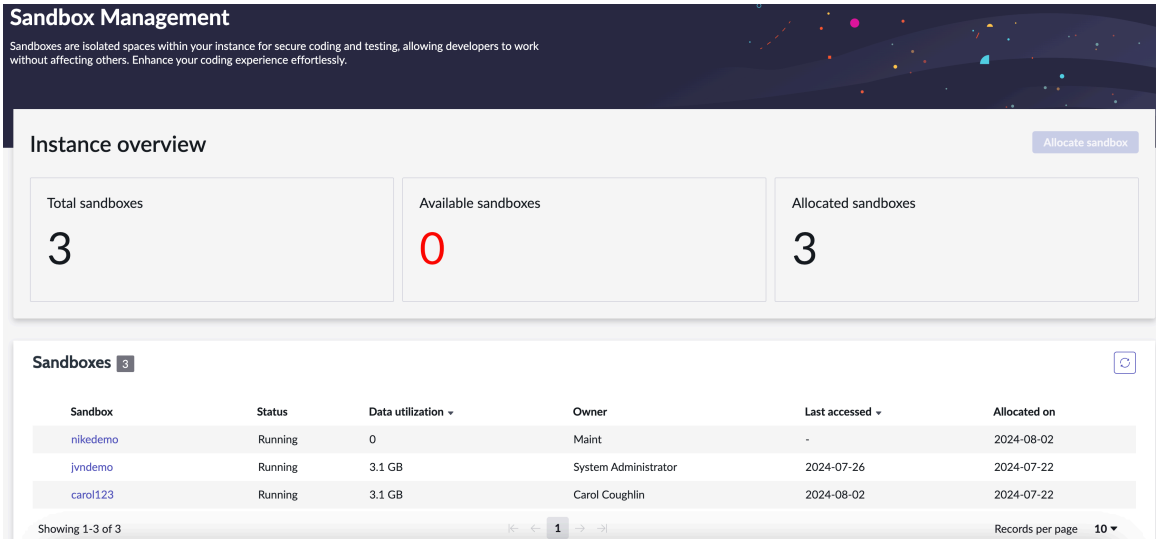
Organizations can face challenges when addressing urgent defects and critical feature enhancements simultaneously in applications. Traditional shared development environments introduce risks such as code conflicts, configuration overlaps, and deployment delays, making it difficult to manage parallel workstreams efficiently. Development teams struggle to deliver urgent fixes and new features concurrently without disrupting each others' progress. The absence of isolated, independent development environments slows down delivery, increases rework, and hampers overall agility.

Developer Sandboxes enables better development in the following ways:

- Isolation: Each developer works in an independent sandbox, ensuring changes do not affect other team members’ work.
- Faster delivery: Teams can work concurrently, reducing development cycle time and enabling faster turnaround for urgent fixes and enhancements.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Safe testing: Developers can test configurations, workflows, and integrations within their sandbox without risking system stability.
- On-demand provisioning: Admins and developers can quickly provision sandboxes for specific tasks or experiments without waiting for shared resources.



The Sandbox Management home dashboard displays the total, available, and allocated sandboxes in your instance. The dashboard also displays information relevant to each sandbox, including, the status, data utilization, owner, last accessed date, and when the sandbox was allocated.

Check your entitlements to determine whether you have access to Developer Sandboxes. For more information, see Developer Sandboxes entitlements.

Warning: Because sandboxes are retired automatically after an upgrade or clone, ensure any work that you want to keep is preserved before upgrading or cloning.

- For upgrades, you can restore work from the remote update sets that Developer Sandboxes automatically created from prior sandboxes.
- For clones, you must manually save and restore all work in sandboxes.
- Any custom table configuration changes or fixes must be reapplied after an upgrade. Contact Now Support to open a case.

For details, see Cloning and upgrading considerations for Developer Sandboxes.

Build Agent is supported in Developer Sandboxes. For more information, see Build Agent.

Integrate sandboxes with source control

Developer Sandboxes provide an isolated environment that integrates with source control, such as Git. Using merge tools helps eliminate conflicts and enables parallel development. For more information, see Source control and Developer Sandboxes.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Developer Sandboxes users

Users

User	Description
Sandbox license admins	Sandbox license admins can assign purchased sandbox packs to non-production instances from App Engine Management Center (AEMC).
Delegated developers	Delegated developers can request sandboxes from an admin or sandbox manager.
Admins	Admins can allocate or retire sandboxes.
Sandbox managers	Sandbox managers can administer the lifecycle of all sandboxes without full admin privileges.
Sandbox users	Sandbox users can request and view Developer Sandboxes.

Sandbox users use the same login credentials for their sandbox as the base instance. If you use Single Sign-On (SSO), enabling it on the base instance also authenticates Developer Sandboxes [var.developer-sandboxes-long] using the same credentials. For information on enabling SSO, see Installing Developer Sandboxes.

Note: Instances with vanity URLs support SSO.

Developer Sandboxes workflow

The delegated developer or admin of a sandbox procures a sandbox, makes changes or experiments with development, tests their changes, pushes their changes, and waits for an admin to clone the instance.

1. A sandbox manager or admin allocates a sandbox to start story work.
2. The developer makes development changes and tests them out in their fully isolated sandbox.

   Note: The work done in one sandbox doesn't appear in other sandboxes or other instances.

3. Once the developer is content with their work and ready to promote their changes to a shared, integrated environment, they push their changes to the desired upstream shared instance. For example, a `test`/QA instance. There are two ways to promote changes:
   a. Using source control (preferable) or exports through Git
   b. Using update sets and imports. Update set sources between the sandbox and the base instance are automatically created when the sandbox is ready. For more information, see Update sets transfer between sandboxes and base instance.
4. Further testing can be done on the shared instance.
5. The admin of a sandbox instance clones changes from the `test`/QA instance to make those changes the default for all future allocated sandboxes.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Developer Sandboxes benefits

Developer Sandboxes benefits

Benefit	Feature	Users
Enable parallel development	Enable multiple developers to work on different stories or features at the same time using the same starting source code, all while keeping the non-production baseline instance clean.	• Admins • Delegated developers • Sandbox managers
Reduce merge conflicts with source control	Enables integration with source control for more successful co-development. For more information, see Source control and Developer Sandboxes.	Delegated developers, sandbox managers
Self-serve license assignment	Assign purchased sandbox packs to non-production instances directly from App Engine Management Center (AEMC), without opening a support case. For more information, see License assignment.	Sandbox license admins
Pooling for faster allocation	When allocating a sandbox, a sandbox is claimed from a pre-created pool rather than waiting for a new instance to be provisioned. The pre-created pool helps users start working in sandboxes faster.	• Admins • Delegated developers • Sandbox managers
Sandbox alias	Reference name for the sandbox you want to allocate.	• Admins • Delegated developers • Sandbox managers

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Developer Sandboxes benefits (continued)

Benefit	Feature	Users
Allocate to	Enables you to allocate a sandbox to yourself, or for an admin to allocate a sandbox to someone else.	• Admins • Delegated developers • Sandbox managers

What to explore next

To learn more about installing and configuring Developer Sandboxes, refer to Installing Developer Sandboxes.

Supported ServiceNow AI Platform features in Developer Sandboxes

Developer Sandboxes supports almost all development-related ServiceNow AI Platform features.

Development features and apps

Supported features include the following:

- Metadata updates
- System Update Sets
- Legacy source control

> **Note:** Each sandbox can connect to source control with its own git branch. For more information, see Source control and Developer Sandboxes.

- ServiceNow IDE
- Workflow Studio
- App and plugin installation

> **Note:** Apps and plugins can be installed and upgraded in a sandbox independently.

- Outgoing integrations

> **Note:** You must manually update incoming integrations to support the unique URLs of sandboxes.

Sandbox considerations and limitations

Upgrading an instance automatically backs up update sets to the base instance and recreates the sandboxes on that instance. Sandboxes are recreated in a base state, so you must restore backed up update sets. For details, see Cloning and upgrading considerations for Developer Sandboxes.

> **Note:** Sandboxes are not automatically recreated after a clone. You should save your work from a sandbox before the clone so you can recreate it.

You can have up to 30 sandboxes per instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Developer Sandboxes does not support self-hosted instances by default, though you can set up your own networking and routing changes to support sandboxes.

Note: Developer Sandboxes are not supported on UI/Worker node instances.

Source control and Developer Sandboxes

Use source control with Developer Sandboxes to enable parallel development and prevent merge conflicts.

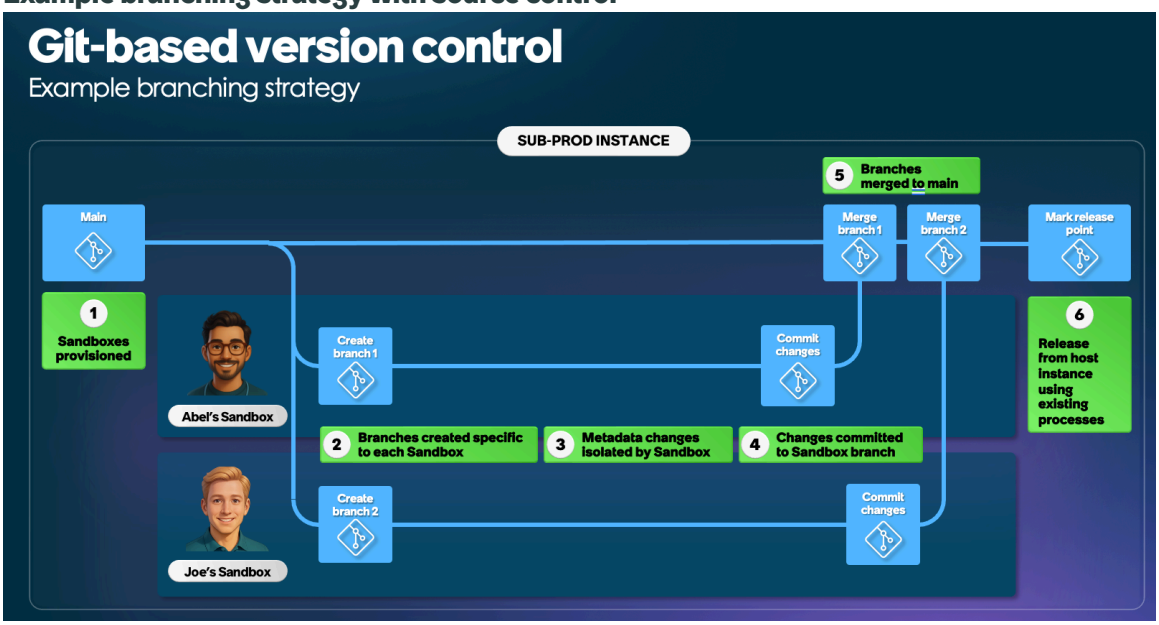
Source control overview

Traditional, single-instance development is an outdated model that requires oversight and maintenance. Sharing a single instance increases the risk of merge conflicts and restricts teams to a single version of code. Enterprise IT architecture standards are increasingly mandating full version control for architectural compliance and IP management.

In support, use Developer Sandboxes in tandem with Git source control to synchronize changes between the base and sandbox environments.

Using source control, developers can work on independent branches and merge them according to your organization's best practices.

Example branching strategy with source control



Git credentials

Each sandbox inherits Git credentials from the base instance. However, developers can add or change Git credentials locally within the sandbox. A new branch should be created for each sandbox.

Update sets transfer between sandboxes and base instance

When a sandbox is ready, update set sources are automatically created on both the sandbox and the base instance, enabling transfers in either direction without manual configuration.

When a sandbox is ready, two update set sources are automatically created to enable transfer of completed update sets in either direction without manual configuration:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- On a sandbox, a source named **Base instance** points to the base instance.
- On the base instance, a source named **Sandbox: [sandbox name]** points to the sandbox.

Use these sources with the standard update set process to retrieve, preview, and commit completed update sets from a sandbox to the base instance. You can also use them to transfer from the base instance to a sandbox. For more information on update sets, see System update sets.

When a sandbox is retired, both sources are removed automatically. The base instance source is deleted, and the sandbox source is dropped along with the rest of the sandbox data. No manual cleanup is needed.

Note: Source control is the preferred approach for transferring changes between sandbox and base. Update set sources are available as an alternative for teams that use update-set-based workflows.

Developer Sandboxes and metadata

Each sandbox provisioned with Developer Sandboxes is an isolated environment within an instance. Developers can build and test in parallel without affecting other work.

Sandboxes can be allocated to specific stories, developers, test plans, or any custom criteria.

Developer Sandboxes and metadata configuration

Each sandbox is initialized with the full metadata of the base instance. For example, the metadata includes scripts and business rules.

Any changes made to metadata configuration in the sandbox don't affect the base instance until changes are merged. For example, you can activate a plugin on a sandbox, and it isn't activated on the base instance. Sandboxes isolate risky changes from the base instance and other sandboxes.

User credentials and roles are isolated to each sandbox. Changing a user's role in a sandbox doesn't affect other sandboxes or the base instance.

Any changes to metadata configuration must first be committed from the sandbox into the base instance. The Developer Sandboxes administrator must then merge the changes back into the base instance. The changes are then available for the next provisioned sandbox, but not for existing sandboxes.

Developer Sandboxes and table data

Table data behavior in sandboxes depends on how each table is configured. By default, tables use the Shared configuration. Tables that extend sys_metadata are configured as Full Copy.

Each table in a sandbox instance can be configured with one of the following data modes:

Shared

The default configuration (introduced in the Zurich release). Table data is shared across the base instance and all sandboxes. All changes, additions, and deletions to records are visible across all sandboxes.

Full Copy

Table data is copied in full from the base instance to each sandbox at the time the sandbox is created. Sandboxes receive a snapshot of the data. All subsequent record changes are isolated to the sandbox.

Zero Copy

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The table is isolated, but no data is copied from the base instance. Sandboxes start with no records in that table. All record changes are isolated to the sandbox.

Partial Copy

Data is copied from the base instance based on a filter condition. This option has limited usability due to cross-table references.

Note: To change the Task or CMDB tables, contact Now Support.

To view the current table configurations, view `sys_dsb_table_config`. Table configuration is set at the base or parent table for all inherited tables.

Records created in a sandbox on a Shared table are immediately available on the base instance and any other sandbox that shares that table. For isolated tables (Full Copy, Zero Copy, or Partial Copy), record changes remain isolated to that sandbox. Making a schema change also isolates the table, and any records added after that point are isolated.

To generate synthetic test data, use AI Data Kit. For more information, see AI Data Kit.

Developer Sandboxes and business rules

Business rules are metadata inherited from the base instance. You can see business rules on a sandbox by navigating to **All** > **Administration** > **Business rules**.

Business rules are copied, but isolated. For more information on Business rules, see Classic Business rules.

General guidelines and use cases for Developer Sandboxes

Follow some general guidelines to ensure you're optimizing your use of sandboxes.

Number of Developer Sandboxes users

Each sandbox should have only one person working in it. Multiple users per sandbox would negate the benefits of parallel development.

Entitlements can be stacked, supporting up to 30 sandboxes per instance.

Note: The number of sandboxes does not impact the instance performance.

Users can be signed into the base instance and multiple sandboxes at the same time. Sandbox users use the same login credentials for their sandbox as the base instance. If you use Single Sign-On (SSO), enabling it on the base instance also authenticates Developer Sandboxes [var.developer-sandboxes-long] using the same credentials. For information on enabling SSO, see Installing Developer Sandboxes.

Note: Instances with vanity URLs support SSO.

Timing for Developer Sandboxes

Use Developer Sandboxes to create intentionally transient environments. The longer they last, the farther they get from the original base instance. Thus, Developer Sandboxes should be as short-lived as possible.

Sandboxes are temporary and can be aligned to short development periods. For example, in Agile development, sandboxes can last as long as the following:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

1. A sprint
2. A story
3. A round of testing

Examples of sandbox allocation

The following are examples of how sandboxes can be used:

- Create branches at the start of a sprint.
- A new branch should be created for each sandbox.
- If you need to fix a bug in the middle a sprint, instead of stashing changes, create a second sandbox with a new branch and fix the bug there. Then merge those changes into the release branch and deploy the bug fix to production.
- Create a sandbox to test a story or feature by pulling in the changes to a new sandbox, which you will retire when the testing is complete.

Developer Sandboxes vs. non-production instances

Developer Sandboxes are not a replacement for non-production instances. Developer Sandboxes are intended to be temporary, while non-production instances are more permanent and intended for use long-term.

The following table compares how non-production instances and Developer Sandboxes are most effectively used.

Comparing suggested non-production instances and Developer Sandboxes

Non-production instances	Developer Sandboxes
Partition your company's work by team or project	Isolate developer metadata changes within a project.
Groups start from significantly different configurations	All developers begin with the same baseline instance configuration.
Concurrent workstreams are isolated or minimally aligned	Development activities follow a consistent cadence.
Durable environment for long-term changes	Work can be completed in a temporary environment and committed to version control.

For more information on setting up sandboxes on instances, see Installing Developer Sandboxes.

Developer Sandboxes and ServiceNow Fluent

Developer Sandboxes work best with ServiceNow Fluent and ServiceNow IDE.

Low-code changes represented in XML markup sometimes cause merging issues, because the generated file structure can make it challenging to align changes. When using low-code builders on the ServiceNow AI Platform, the best long-term strategy is to save changes in ServiceNow Fluent instead of XML.

ServiceNow Fluent is a domain-specific programming language that you can use to define application metadata in source code. Developers and admins can easily look up changes in version control, such as Git. For details, see ServiceNow Fluent.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

You can use Developer Sandboxes with System Update Sets, but a more forward-looking solution is to use ServiceNow IDE. Pairing your sandboxes and ServiceNow IDE with version control and deployment apps (such as App Engine Management Center) enables a cleaner, more streamlined deployment ecosystem. For more information, see ServiceNow IDE.

Developer Sandboxes FAQs

See the ServiceNow Community article on Developer Sandboxes: FAQ and Getting Started Guide.

Get help with Developer Sandboxes

To get help with Developer Sandboxes, your ServiceNow instance, plugins, permissions, and more, watch a short video to contact the ServiceNow admin who works in your company. You can also check some helpful resources.

Get help now from your Now Support administrator

The ServiceNow admin in your company can help you with things like activating plugins, kicking off upgrades, unlocking accounts, updating user permissions, and much more.

https://player.vimeo.com/video/1056112088?h=1a5cd3b245&badge=0&autopause=0&player_id=0&app_id=58479

Helpful resources

ServiceNow resources that can provide you with helpful information include:

Developer Sandboxes: FAQ and Getting Started Guide

Browse frequently asked questions and other tips for getting started

Take the Developer Sandboxes Basics course

Developer Sandboxes on ServiceNow University

Now Learning Course: Developer Sandbox Basics

Community post on the Now Learning course

Installing Developer Sandboxes

Developer Sandboxes is a installed on your instance, with licenses from a pack assigned by your admin. Review prerequisites, plugin requirements, and SSO configuration before installation begins.

Developer Sandboxes installation

Check your entitlements to determine whether you have access to Developer Sandboxes. For more information, see Developer Sandboxes entitlements.

To install Developer Sandboxes:

- The com.glide.dsb plugin must be installed on each license management instance that will host sandboxes.

  Warning: Do not install or enable the `com.glide.dsb` plugin on a production instance or any instance where sandboxes will not be hosted.

- The com.glide.dsb.licensing plugin must be installed on your production or controller instance to manage sandbox license assignment from App Engine Management Center.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- After you install the com.glide.dsb.licensing plugin, you must reconfigure the `glide.dev_sandbox.num.controller` property. For more information, see Properties installed with Developer Sandboxes.
- A sandbox license admin must assign licenses. For more information, see License assignment.
- If your instance is in a regulated environment, check with ServiceNow Support for more information about support for Developer Sandboxes.

Prerequisites for installing Developer Sandboxes

Verify the following prerequisites before installing Developer Sandboxes.

The controller instance must be the license management instance.

Non-production instances must be:

- On ADCv3 (Application Delivery Controller version 3)

  Note: If a non-production instance is not on ADCv3, it will be moved it as part of the setup.

- Yokohama or higher

Instances and sandboxes

Sandboxes are instance-specific, which means that you can't move a sandbox from one instance to another. If a new instance needs to be sandbox enabled, it will be a net-new order.

Note: A free pack of four sandboxes is automatically available to all organizations. The free pack can be assigned to one non-production instance and can't be split across instances.

Each sandbox is hosted in its own app-node. Therefore, a key part of the setup is adding the required number of additional nodes on the instance. For example, to support 10 sandboxes, 10 new nodes would be added to an instance.

Developer Sandboxes are available in packs of 10. When you buy two or more packs, you can distribute them across instances using the self-serve license assignment UI in AEMC.

Note: The minimum is one pack of sandboxes (10) per instance, and a maximum of three packs (30 sandboxes) can be assigned to a single instance. For example, if you buy two packs, you can't split them 5-15. Each instance must have at least 10 sandboxes. However, with the free pack, you can have a maximum of 34 sandboxes on one instance. For more information, see License assignment.

Developer Sandboxes does not support self-hosted instances by default, though you can set up your own networking and routing changes to support sandboxes.

For more information on sandboxes and instances, see General guidelines and use cases for Developer Sandboxes.

Enabling SSO

Developers can access sandboxes using instance credentials and Single Sign-On (SSO).

The following properties must be set in the base instance before developers can use SSO to access sandboxes:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Disable ACR: `sys_property glide.sso.acr.enable = false`
- Enable SSO: `glide.authenticate.multisso.enabled = true`

Developer Sandboxes entitlements

Your company's entitlements determine whether you have access to Developer Sandboxes.

A free pack of four sandboxes is automatically available to all organizations. The free pack can be assigned to one non-production instance and can't be split across instances. You can use the four free sandboxes on the same instance as purchased sandboxes, for a maximum of 34 sandboxes on an instance.

In addition to the free sandboxes, Developer Sandboxes entitlements include the following parameters:

- Developer Sandboxes is a standalone product and isn't included with any other product.
- Developer Sandboxes are packaged in increments of ten. One license is 10 sandboxes.
- Developer Sandboxes packs are purchased at the organization level. After purchase, a sandbox license admin can distribute packs across non-production instances using the self-serve license assignment UI in App Engine Management Center (AEMC). Each instance must receive at least one full pack. For more information, see License assignment.
- Entitlements can be stacked, supporting up to 30 sandboxes per instance. For more information on suggested number of entitlements per instance, see General guidelines and use cases for Developer Sandboxes.

For details on how to check your entitlements, see Viewing instance-level entitlements in Subscription Management.

License assignment

License assignment in App Engine Management Center (AEMC) lets Developer Sandboxes admins distribute purchased sandbox packs across non-production instances without opening a support case. Each pack contains 10 sandboxes, and assignments are managed from a license management instance.

The license management instance (or controller) is the production instance where the sandbox licenses are managed and packs are allocated to the non-production instances.

For full details on how to assign a license, see Assign Developer Sandboxes packs in the AEMC documentation.

The license assignment UI is part of the Developer Sandboxes Licensing plugin (com.glide.dsb.licensing) and is accessible from the Developer Sandboxes section of AEMC on your license management instance. The UI is not functional on non-production instances.

Note: The com.glide.dsb.licensing plugin should be used from only one instance, the license management instance.

Prerequisite: Add managed instances

Before a sandbox license admin can assign packs to a non-production instance, the instance must be added as a managed instance in Multi-Instance Management. For example, use Multi-Instance Management to save an instance as a development instance linked to the license management instance.

Multi-Instance Management is required for assigning Developer Sandboxes licenses. It depends on the Multi-Instance Setup app (`sn-app-amf`), which must be installed separately from the

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ServiceNow® Store. This app is not a dependency of the `com.glide.dsb.licensing` plugin and is not installed automatically. For details on adding managed instances, see Connecting controller and managed instances.

Key benefits of self-service license assignment

License assignment for sandboxes:

- Distribute purchased packs across instances without opening a support case.
- View purchased packs, assigned packs, and available capacity in one consolidated dashboard view, so you can see current assignment status before making changes.
- Confirm that the target non-production instance has the required plugin installed before any packs are assigned, helping to avoid misconfiguration.

Details and considerations for license assignment

Consider the following when using developer sandbox license assignment:

- Only users with the sn_dsb_commons.sandbox_license_admin role or the admin role can assign licenses.
- Pack assignments are permanent. After packs are assigned to an instance, the count can't be decreased. Admins can only increase an existing assignment.
- A maximum of three packs can be assigned to a single non-production instance, with a maximum of 30 sandboxes per instance.
- License management is available only from a license management instance. The AEMC UI is not accessible from non-production instances.

How license assignment works

The license assignment dashboard retrieves purchased pack data from the subscription management table and current assignment state from the instance allocation table (`sys_dsb_instance_allocation`). Assignment works as follows:

1. The assignment dashboard displays each purchased sandbox pack and the total number of sandboxes represented. Each pack contains 10 sandboxes.
2. You (a sandbox admin) select **Assign sandboxes**. A modal opens where you select the non-production instance, as configured in Multi-Instance Management, to receive the packs.
3. The ServiceNow AI Platform verifies the instance. If verification fails, an inline error indicates that the Developer Sandboxes plugin (com.glide.dsb) must be installed on the non-production instance before packs can be assigned.
4. You select the number of packs to assign. The modal shows total packs available, packs already assigned to other instances, and an assignment progress indicator.
5. You save the assignment. The status changes to **In progress** while the system processes the assignment, then changes to **Ready** or **Error**.

Cloning and upgrading considerations for Developer Sandboxes

You should understand how plugins and sandboxes work before you clone or upgrade an instance with Developer Sandboxes. Always back up your work in a sandbox before any clone or upgrade, either by exporting the update sets or committing to source control.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Warning: Because sandboxes are retired automatically after an upgrade or clone, ensure any work that you want to keep is preserved before upgrading or cloning.

- For upgrades, you can restore work from the remote update sets that Developer Sandboxes automatically created from prior sandboxes.
- For clones, you must manually save and restore all work in sandboxes.
- Any custom table configuration changes or fixes must be reapplied after an upgrade. Contact Now Support to open a case.

Upgrading instances with Developer Sandboxes

Family, patch, and security upgrades automatically retire all existing sandboxes. Sandboxes are recreated, but any changes from pre-existing sandboxes must be manually restored from the automatic backup created.

After an upgrade, sandbox recreation occurs automatically, but sandboxes may not be available immediately. A status banner in the Sandbox Management Home dashboard indicates when recreation is in progress.

Developer Sandboxes automatically backs up any update sets from the sandboxes and exports them to the base instance. The following rules apply:

- The update set must contain at least one change since the sandbox was created for it to be backed up.
- Incomplete update sets are backed up as long as there's at least one change.

Note: When using sandboxes, make sure to save or backup work consistently. Automatic backups are available only for instances on Australia Patch 3 and higher.

Backups are found in the retrieved update sets table, identified by the name of the sandbox they were backed up from. For example, "DSB [sandboxname] (backup date): [Original update set name]". For more information, see Preview a remote update set and Commit an update set.

Review the update sets, then preview and commit them in each sandbox as needed.

Note: If you're using source control, you should restore your work to the sandbox from there. For more information, see Source control and Developer Sandboxes.

Cloning instances with Developer Sandboxes

Note: Sandboxes are not automatically recreated after a clone. You should save your work from a sandbox before the clone so you can recreate it.

Install the Dev Sandboxes CC (com.glide.dsb.cc) plugin on the source instance to enable clone preservers that protect Developer Sandboxes feature enablement on your target instance. For more information on clone preservers, see Create a clone preserver

If you have a custom clone profile, these records should be linked so that cloning over a sandbox instance maintains sandbox functionality:

- clone_cleanup_script: `/clone_cleanup_script.do?sys_id=2b5e8051ff03221016abffffffffff58`
- clone_data_exclude: `/clone_data_exclude_list.do?sysparm_query=nameSTARTSWITHsys_dsb`
- clone_data_preserver: `/clone_data_preserver_list.do?sysparm_query=tableSTARTSWITHsys_dsb%5ENQname%3DDSB%20Properties`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Developer Sandboxes domain separation

Domain separation is supported for Developer Sandboxes. Domain separation enables you to separate data, processes, and administrative tasks into logical groupings called domains. You can control several aspects of this separation, including which users can see and access data.

Support level: Basic

- Business logic: Ensure that data goes into the proper domain for the application's service provider use cases.
- The application supports domain separation at run time. The domain separation includes separation from the user interface, cache keys, reporting, rollups, and aggregations.
- The owner of the instance must set up the application to function across multiple tenants.

Sample use case: When a service provider (SP) uses chat to respond to a tenant-customer's message, the customer must be able to see the SP's response.

For more information on support levels, see Application support for domain separation.

Overview of domain separation in Developer Sandboxes

Domain separation must be installed and set up on the base instance before sandboxes are created. When creating a sandbox, the domains defined in the base instance are inherited, and users can define new domains only for the lifecycle of that sandbox. Installing the plugin directly in a sandbox will not work correctly.

Related topics

Domain separation for service providers

Components installed with Developer Sandboxes

Several types of components are installed with activation of Developer Sandboxes, including tables and user roles.

Roles installed with Developer Sandboxes

Role title [name]	Description
Sandbox license admin [sn_dsb_commons.sandbox_license_admin]	Assign sandbox licenses to non-production instances in App Engine Management Center (AEMC). **Note:** You must have the com.glide.dsb.licensing plugin installed for this role.
Sandbox manager [sandbox_manager]	Manage the lifecycle of all sandboxes.
Sandbox user	Request and view sandboxes.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Role title [name]	Description
[sandbox_user]	

Plugins for Developer Sandboxes

Plugin	Description
com.glide.dsb	Developer Sandboxes application plugin
com.glide.dsb.licensing	Enables self-serve sandbox license assignment from (AEMC). Install on your base or controller instance to distribute sandbox packs to non-production instances. **Note:** The com.glide.dsb.licensing plugin should be used from only one instance, the license management instance.
App Management Framework (sn-app-amf)	Required for sandbox pack license assignment. This app is available in the ServiceNow Store and must be installed separately. It is not a dependency of the `com.glide.dsb.licensing` plugin and is not installed automatically. **Important:** Install the App Management Framework app from the ServiceNow Store before using sandbox pack license assignment.

Tables installed with Developer Sandboxes

Note:

- Tables that aren't explicitly mentioned in table config are shared across all sandboxes instead of copied separately.
- If you make a schema change to a shared table, the table becomes an isolated table on the sandbox that initiated the schema change. For example, adding a column to a shared table isolates it in the sandbox.

Table	Description
Main Developer Sandboxes table [sys_dsb]	Ensures controlled access to critical sandbox data and prevents unauthorized modifications. Only the admin or sandbox_manager role can read or report_view.
Developer Sandboxes instance allocation table	Stores the sandbox pack allocation state for each non-production instance, including the number of packs assigned and the current allocation status.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Table	Description
[sys_dsb_instance_allocation]	**Note:** This table is available only if the com.glide.dsb.licensing plugin is installed.
Developer Sandboxes alias table [sys_dsb_table_alias]	• Stores table alias data for sandbox operations. • Prevents unauthorized changes to alias records. • Ensures secure management of critical sandbox-related configurations. Only the admin or sandbox_manager role can read or report_view.
Developer Sandboxes configuration table [sys_dsb_table_config]	• Stores configuration data for sandbox tables. • Ensures secure management of sandbox table configurations. • Prevents unauthorized access or modifications. Only the admin or sandbox_manager role can read or report_view.
Developer Sandboxes message table [sys_dsb_message]	Stores messages for sandboxes.
[sys_dsb_lifecycle_log]	Stores all lifecycle-related events for sandboxes along with context describing the events.
[sys_dsb_lifecycle_assign_log]	Stores lifecycle events describing sandbox node assignment.
[sys_dsb_lifecycle_create_log]	Stores lifecycle events describing the creation of a sandbox.
[sys_dsb_lifecycle_destroy_log]	Stores lifecycle events describing the deletion of a sandbox.
[sys_dsb_query_condition]	Stores rules for partial copy tables.

Properties installed with Developer Sandboxes

The system properties available in Developer Sandboxes govern application behavior, enabling developers to configure and optimize their testing environments effectively.

Note: To open the System Properties [sys_properties] table, enter `sys_properties.list` in the navigation filter.

These properties are available for Developer Sandboxes.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties for Developer Sandboxes

Property	Description	Details
`glide.dev_sandbox.num.controller`	Number of nodes that run as a controller. • Type: number • Default value: 2	• Can't be 0. • Increasing co controller no • Decreasing c nodes. • More controll sandboxes. **Importar** property af com.glide.c ensure the nodes for y Failure to re after install sandbox av
`glide.dev_sandbox.node.healthy_time_min`	Time in minutes that a node is considered healthy. An unhealthy node that's assigned to a sandbox can be remediated by assigning it to another sandbox. Remediation is the process of detecting a sandbox that should be running on a node but isn't, and then assigning another node to run on the sandbox. • Type: number • Default value: 10	• Increasing th node up long less frequent • Decreasing it remediation c • Do not set thi time it takes a 3 minutes. Du still considere remediator fr another node
`glide.dev_sandbox.node.poll_interval_seconds`	Number of seconds between status checks for a node expected to go offline in the sys_cluster_state	• Generally do • Used during s determine ho for being set

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties for Developer Sandboxes (continued)

Property	Description	Details
	table, which contains the nodes that are assigned to an instance. • Type: number • Default value: 10	• Increasing m more often. • Decreasing m happens less
`glide.dev_sandbox.node.poll_timeout_min`	Number of minutes to wait for a node to shutdown, and have a **Status** of Offline in sys_cluster_state table. • Type: number • Default value: 2	• Generally do • Used during s determine ho sandbox nod • Increasing m longer. • Decreasing m completes fa
`glide.dev_sandbox.default_table_config`	Default table config in Developer Sandboxes. • Type: string • Default value: shared_table • Other possible values: ○ shared_table ○ full_copy ○ zero_copy	• The full_copy uses much m make sandbo with large dat • The zero_cop with less initi empty and m data is added
`glide.dev_sandbox.dsb_db_copier_threads`	Number of concurrent table creations in a sandbox. • Type: number • Default value: 1 • Other possible values: 1-5	• Increasing th contention ar sandboxes. • ⚠ **Warnin** without c manager

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties for Developer Sandboxes (continued)

Property	Description	Details
`glide.db.dsb.data_copy_processor.threads`	Number of data copy plans to run concurrently during sandbox creation. • Type: number • Default value: 1 • Other possible values: 1-5	• This value is glide.dev_sar value. • Increasing th contention ar sandboxes. • ⚠ **Warnin** without c manager
`glide.db.dsb.data_copy_processor.chunk_copy.threads`	Number of data copiers running at a time for each table. • Type: number • Default value: 5 • Other possible values: 1-5	• This value is glide.dev_sar and glide.db.dsb. values. • Decreasing th sandbox crea
`glide.dev_sandbox.cleanup_retries`	Number of retries the Developer Sandboxes destroy code attempts to do clean up any leftover Developer Sandboxes tables. • Type: number • Default value: 3	
`glide.dev_sandbox.backup_tables`	Tables that will be preserved for recreating developer sandboxes post clone. • Type: string • Default value: sys_repo_config	Comma-delimit If a table has is in each sandbo during clones, t the re-created s This property is Developer Sand glide.dev_sand impacts only sa
`glide.dev_sandbox.export.poll_interval_seconds`	Frequency in seconds that	Decreasing the go faster but at

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties for Developer Sandboxes (continued)

Property	Description	Details
	the controller node checks if a sandbox has completed exporting its altered update sets. This property is used only when exporting sandbox update sets before an instance upgrade begins. • Type: number • Default value: 10	
`glide.dev_sandbox.export.poll_timeout_min`	How long the controller node waits for a sandbox to export its altered update sets before an instance upgrade. If the timeout limit is hit, the upgrade continues regardless of export state. • Type: number • Default value: 10	• Decreasing th unexported u • Increasing th changes are upgrade.
`glide.dev_sandbox.node.healthy_time_min`	Time window in which a node must report to sys_cluster_state to be considered healthy and included in sandbox capacity calculations. • Type: number • Default value: 10	

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties for Developer Sandboxes (continued)

Property	Description	Details
`glide.dev_sandbox.lifecycle.max_concurrent_events`	Maximum limit of concurrent sandbox allocations/ retirements events. • Type: number • Default value: 5	Increasing the v concurrent allo overload the da

Administering Developer Sandboxes

If you have permission, you can allocate and retire the sandboxes you've created in your instance.

Configuration overview

1. Allocate a sandbox

   Allocate the sandboxes created to your development teams.

2. Requesting sandboxes

   Request a sandbox for development.

3. Retire a sandbox

   Retire sandboxes that are outdated or to make room for new sandboxes in your instance.

Allocate a sandbox

Allocate sandboxes to your development teams so they can start using them for development.

Before you begin

You can watch a short video on how to allocate a sandbox.

https://player.vimeo.com/video/1158960658?h=38a1c4703f&badge=0&autopause=0&player_id=0&app_id=58479

Role required: admin or sandbox_manager

About this task

You can allocate a sandbox at the beginning of a story or during the project planning process.

When you allocate a sandbox, you claim one from a pre-created pool rather than waiting for a new instance to be provisioned. This makes sandboxes available faster, so you can start working sooner.

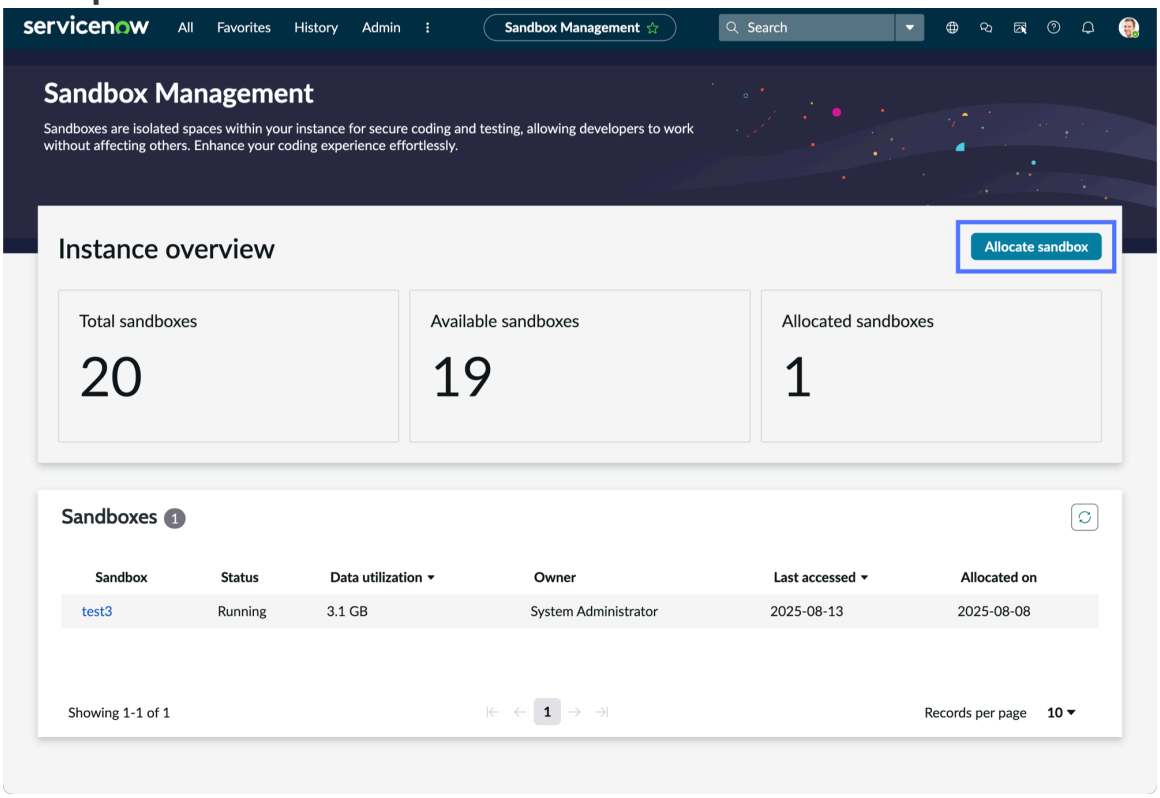
Note: Because pooled sandboxes are pre-created, they may be out of date from the current base instance state, but are refreshed every 24 hours.

Procedure

1. Navigate to **All** > **Sandbox Management** > **Sandbox Management Home**.
2. Select **Allocate sandbox**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

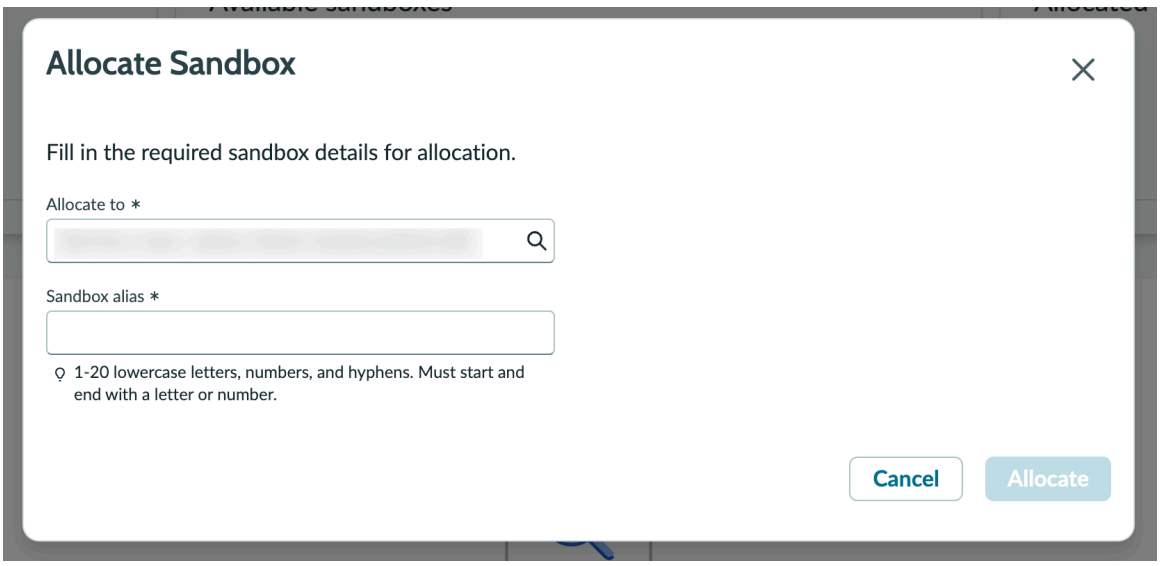
Developer Sandboxes home



3. On the form, fill in the fields.

Field	Description
Allocate to	User that owns the sandbox.
Sandbox alias	Name that's used to identify the sandbox, for example, in the sandbox management table. The name is configurable, but you can't change the URL.

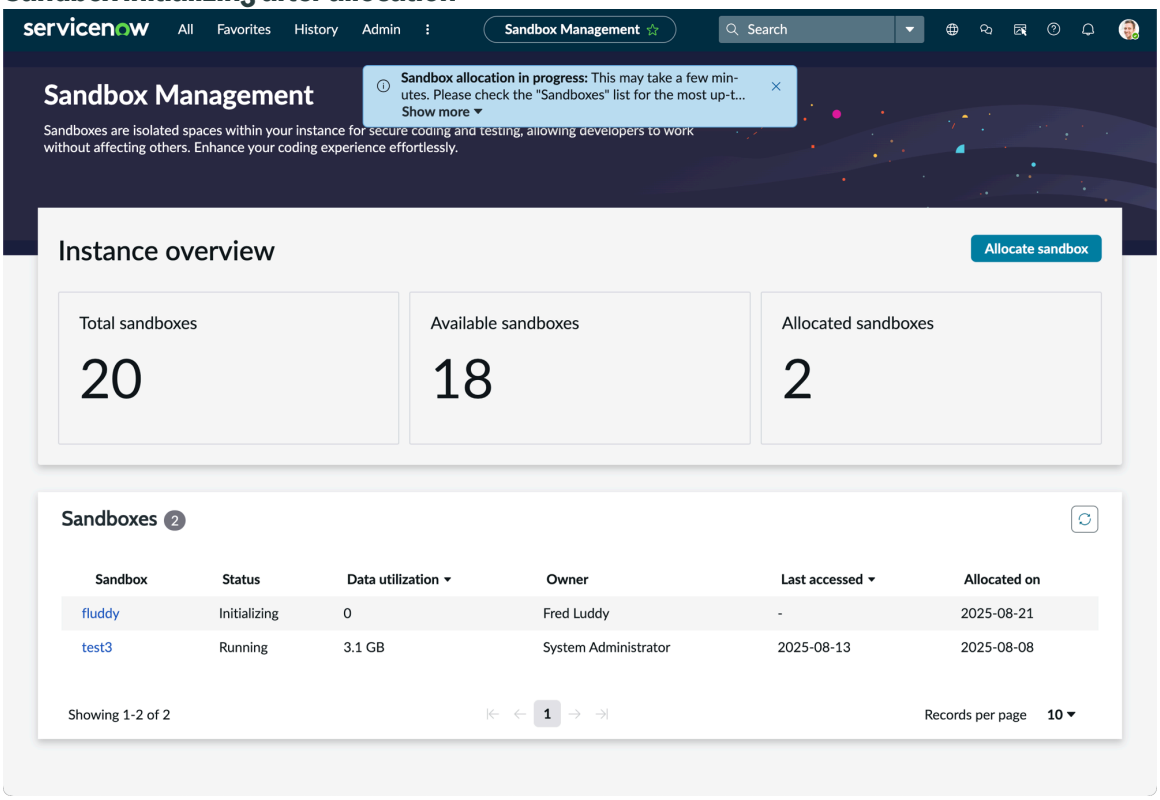
Allocate Sandbox details



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

4. Select **Allocate**.

Sandbox initializing after allocation



Result

Developer Sandboxes starts the process of provisioning the sandbox.

Note: Sandboxes from the pool are allocated quickly. If the pool is exhausted, it can take between ten minutes and two hours to allocate a sandbox. You can select the refresh icon to see updates to the Sandboxes list. The record for the sandbox appears in the list as **Initializing** until the sandbox is ready.

Once allocated, developers can access their sandbox by one of two ways:

- Selecting the sandbox name on the Sandbox Management Home page.
- Using the context menu and copy the URL to the clipboard and paste it to the browser.

Sandbox users use the same login credentials for their sandbox as the base instance. If you use Single Sign-On (SSO), enabling it on the base instance also authenticates Developer Sandboxes [var.developer-sandboxes-long] using the same credentials. For information on enabling SSO, see Installing Developer Sandboxes.

Two automatically created update set sources enable you to retrieve, preview, and commit completed update sets between the sandbox and base instance using the standard update set process. For more information, see Update sets transfer between sandboxes and base instance.

Requesting sandboxes

Developers generally have admin rights on the non-production instance where Developer Sandboxes are allocated, and can thus allocate a sandbox to themselves.

However, if you don't have admin rights and need a sandbox, you should contact your company's ServiceNow administrator.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Get help now from your Now Support administrator

The ServiceNow admin in your company can help you with things like activating plugins, kicking off upgrades, unlocking accounts, updating user permissions, and much more.

https://player.vimeo.com/video/1056112088?h=1a5cd3b245&badge=0&autopause=0&player_id=0&app_id=58479

Retire a sandbox

Retire sandboxes that are outdated or no longer needed to make room for new Developer Sandboxes in your instance.

Before you begin

Save your work, for example commit it to source control or export update sets. For more information, see Update sets transfer between sandboxes and base instance.

Role required: admin or sandbox_manager

About this task

You should manually retire sandboxes when your work is complete to maintain a healthy development lifecycle.

⚠ **Warning:** Because sandboxes are retired automatically after an upgrade or clone, ensure any work that you want to keep is preserved before upgrading or cloning.

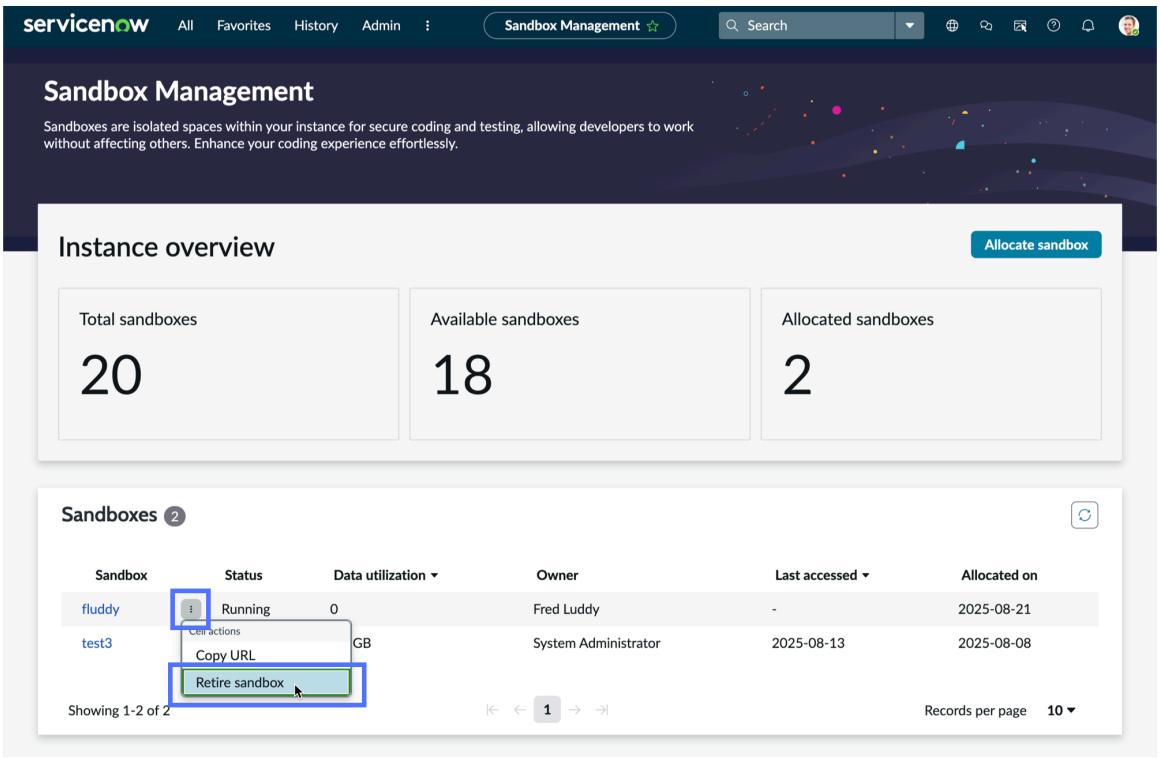
- For upgrades, you can restore work from the remote update sets that Developer Sandboxes automatically created from prior sandboxes.
- For clones, you must manually save and restore all work in sandboxes.
- Any custom table configuration changes or fixes must be reapplied after an upgrade. Contact Now Support to open a case.

Procedure

1. Navigate to **All** > **Sandbox Management** > **Sandbox Management Home**.
2. Select the more options icon ⋮ in the Developer Sandboxes dashboard, and then select **Retire sandbox**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Retire a sandbox



3. Confirm the retirement by selecting **Retire**.

Result
After it's retired, the sandbox is no longer available for use. However, you can allocate new sandboxes as needed. For more information, see Allocate a sandbox.

The update set sources associated with the sandbox are automatically removed during retirement. The source on the base instance pointing to the sandbox is deleted, and the source on the sandbox is dropped along with the rest of the sandbox data. No manual cleanup is needed.

ServiceNow Otto for Creator

ServiceNow Otto for Creator includes generative and agentic AI capabilities that can make developing on the ServiceNow AI Platform more efficient.

Get started

The ServiceNow AI Platform now brings you a new AI experience with three licensing tiers available:

- Foundation: AI basics to deliver insights
- Advanced: AI to boost productivity across relevant use cases
- Prime: Act autonomously with all AI assets, and create your own

For more information, see ServiceNow product tiers.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Install



Install ServiceNow Otto for Creator.

App summary generation



Summarize what an app does using generative AI.

ATF troubleshooting agent



Simplify test failure diagnosis on specific metadata.

Build Agent



Generate a complete scoped application, including core logic and user interface components, in ServiceNow Studio and the ServiceNow IDE.

Catalog item generation



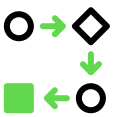
Generate catalog items using generative AI.

Code generation



Get help writing scripts quickly with AI-generated code based on text or code prompts.

Flow generation



Create multiple step flows with generative AI.

Flow recommendations



Generate recommendations for the next step of a flow.

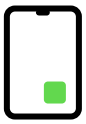
Flow summarization



Summarize what a flow or subflow does with generative AI.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Mobile card generation



Generate cards for mobile applications.

Playbook generation



Generate a playbook from text directions or an image.

Process Mining



Identify the work notes and generate insights about inefficiencies, challenges, and other reasons that cause reassignment of work.

Playbook recommendation



Generate recommendation to replace placeholder activities in a playbook.

Playbook generation from KB



Generate a playbook from an existing KB article using AI.

Playbook summarization



Generate a summary of the playbook to understand the stages, activities, triggers, and inputs of the playbook.

Release lifecycle documentation AI agent



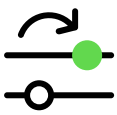
Generate readable documentation across the application lifecycle to improve traceability and governance.

RPA bot generation



Create automations, activities, and add automation logic from text instructions and preview options.

Theme creation



Generate themes from images directly in Theme Builder.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

UI generation	Widget generation and updation	
		
Create experiences by describing what you want using natural language.	Create and modify widgets via natural language prompts.	

Important:

- Not all model providers are available for customers with in-country SKUs, and some AI products/features are currently unavailable for in-country customers. For more information, see the KB1584492 article in the Now Support Knowledge Base. Be sure to check for model provider availability updates in future releases.
- Some AI products/features are currently unavailable for customers in the FedRAMP, NSC DOD IL5, or Australia IRAP-Protected data centers, self-hosted customers, or in other restricted environments. For more information, see the KB0743854 article in the Now Support Knowledge Base. Be sure to check for availability updates in future releases.
- Some AI products/features are currently available only for customers in some regions. Be sure to check for availability updates in future releases.
- Some AI products and skills are not available in Regulated Markets. For more information, see KB2593939: Regulated Markets AI Products/Skills Not Available. Be sure to check for availability updates in future releases.

AI limitations

This application uses artificial intelligence (AI) and machine learning, which are rapidly evolving fields of study that generate predictions based on patterns in data. As a result, this application may not always produce accurate, complete, or appropriate information. Furthermore, there is no guarantee that this application has been fully trained or tested for your use case. To mitigate these issues, it is your responsibility to test and evaluate your use of this application for accuracy, harm, and appropriateness for your use case, employ human oversight of output, and refrain from relying solely on AI-generated outputs for decision-making purposes. This is especially important if you choose to deploy this application in areas with consequential impacts such as healthcare, finance, legal, employment, security, or infrastructure. You agree to abide by ServiceNow's AI Acceptable Use Policy, which may be updated by ServiceNow.

Data processing

This application requires data to be transferred from ServiceNow customers' individual instances to a centralized ServiceNow environment, which may be located in a different data center region from the one where your instance is, and potentially to a third-party cloud provider, such as Microsoft Azure. This data is handled per ServiceNow's internal policies and procedures, including our policies available through our CORE Compliance Portal.

Data collection

ServiceNow collects and uses the inputs, outputs, and edits to outputs of this application to develop and improve ServiceNow technologies including ServiceNow models and AI products. In addition, this application will collect skill usage metrics. Customers can opt out of future data collection at any time, as described in the Now Assist Opt-Out page.

For more information, see the Now Assist documentation.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Exploring ServiceNow Otto for Creator

Explore the AI capabilities, users, benefits, and workflow for using ServiceNow Otto for Creator.

ServiceNow Otto for Creator overview

ServiceNow Otto for Creator includes generative and agentic AI features that make developing on the ServiceNow AI Platform more efficient. The ServiceNow Otto for Creator AI capabilities are embedded across platform tools—including ServiceNow Studio, Workflow Studio, UI Builder, and the ServiceNow IDE—so developers can generate apps, flows, code, and UI components without leaving the tools they use every day. ServiceNow Otto for Creator also includes Build Agent, an autonomous AI agent that can develop full-stack applications through natural language conversations in the ServiceNow IDE.

ServiceNow Otto for Creator benefits

Benefit	Feature	Users
Create a complete full-stack application, including core logic and UI components, through a conversation with an autonomous AI agent.	Build Agent	Pro-code developer
Create multi-step flows from natural language descriptions and generate recommendations for the next step of an existing flow.	Flow generation, Flow recommendations	Citizen developer, Pro-code developer
Write scripts quickly with AI-generated code based on text or code prompts.	ServiceNow Otto for Code	Citizen developer, Pro-code developer
Create UI pages and experiences in UI Builder by describing what you want in natural language.	UI generation, Using UI Builder agent	Citizen developer, Pro-code developer
Generate catalog items from text instructions.	AI Authoring for Catalog Builder	Citizen developer, Pro-code developer
Generate release notes and update set descriptions.	Exploring release lifecycle documentation AI agent	Developer

What to explore next

To learn more about configuring and using ServiceNow Otto for Creator, see:

- Configuring ServiceNow Otto for Creator
- Using generative AI with ServiceNow Otto for Creator
- Using agentic AI with ServiceNow Otto for Creator

Configuring ServiceNow Otto for Creator

To get started using ServiceNow Otto for Creator, install ServiceNow Otto for Creator. Then turn on the skills, AI agents, or agentic workflows that you want to use.

Installing ServiceNow Otto for Creator

Check your company's entitlements to verify that you have access to ServiceNow Otto for Creator. To start using ServiceNow Otto for Creator, you must request it from the ServiceNow®

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Store. Once approval has been granted, you can install it on your instance. See Install ServiceNow Otto for Creator for more information.

Activating AI capabilities

After installing ServiceNow Otto for Creator, you must turn on the AI capabilities that you want to use. Some AI capabilities are turned on by default, such as the ServiceNow Otto for Code skills, including code assist summarization, code assist edit, and code assist generation, and code assist autocomplete. Capabilities that aren't on by default must be turned on or activated in order to use them.

You can turn on ServiceNow Otto for Creator skills in the AI Admin Hub. When you turn on a skill, you might have to complete additional configuration steps, such as assigning roles and restrictions for who can access the skill. See the product documentation for the skill that you want to activate for additional configuration steps.

AI agents and agentic workflows are activated in AI Agent Studio. When you activate an AI agent or agentic workflow, you might have to complete additional configuration steps, such as selecting a large language model (LLM) for the AI agent or agentic workflow and assigning roles and restrictions. See the product documentation for the AI agent or agentic workflow that you want to activate for additional configuration steps.

Required roles

To use ServiceNow Otto for Creator, you might need specific roles. Some roles are specific to the tool that the capability is embedded in, such as catalog_builder_editor to use the catalog item generation skill in Catalog Builder, or flow_designer to use the flow generation skill.

Other roles might be required for ServiceNow Otto® experiences, such as the now_assist_panel_user role for creating applications in ServiceNow Studio with the app generation skill.

ServiceNow Otto for Creator has its own role, now.assist.creator, which might be required for using some ServiceNow Otto for Creator capabilities. See ServiceNow Otto for Creator [now.assist.creator] role for more information.

To learn what roles are required for the ServiceNow Otto for Creator AI capability you want to use, see the product documentation for the capability.

Install ServiceNow Otto for Creator

Install the ServiceNow Otto for Creator application.

Before you begin

- Role required: admin
- Review the ServiceNow Otto for Creator application listing in the ServiceNow Store for information on dependencies, licensing or subscription requirements, and release compatibility.
- Enable Next Experience. For information about activating Next Experience, see Considerations for activating Next Experience.

Procedure

1. Navigate to the ServiceNow Otto for Creator application on the ServiceNow Store .

Important: ServiceNow Otto for Creator requires a separate subscription.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

2. From the ServiceNow Otto for Creator application page, select **Request App**.
3. After approval has been granted, on your instance, navigate to **All** > **System Applications** > **All Available Applications**.
4. Find the ServiceNow Otto for Creator application (sn_now_creator) using the filter criteria and search bar.
5. Select **Install**.

What to do next

Turn on the ServiceNow Otto for Creator skills, AI agents, and agentic workflows that you want to use.

ServiceNow Otto for Creator [now.assist.creator] role

This role grants users access to ServiceNow Otto for Creator skills.

Important: Starting with version 28.4.3 of ServiceNow Otto for Creator, most ServiceNow Otto for Creator AI capabilities don't require the now.assist.creator role. Some AI capabilities might have additional role requirements. See the product documentation for the specific skill, AI agent, or agentic workflow to learn what roles are required.

Contains roles

This role contains no roles.

Groups

List of groups this role is assigned to by default.

None.

Special considerations

None.

Using generative AI with ServiceNow Otto for Creator

Use the generative AI skills available with ServiceNow Otto for Creator to help you build faster and smarter on the ServiceNow AI Platform.

ServiceNow Otto for Creator contains generative AI skills that empower you to build more efficiently on the ServiceNow AI Platform. The skills available with ServiceNow Otto for Creator are embedded in applications across the ServiceNow AI Platform, enabling you to leverage generative AI at every stage of your development process. For example, using ServiceNow Otto for Creator, you can summarize applications in ServiceNow Studio, create UI pages in UI Builder, and generate flows in Workflow Studio. The following table highlights the generative AI skills available with ServiceNow Otto for Creator and the applications the skills work in.

ServiceNow Otto for Creator skills and the applications that they work in

Generative AI skill	Application
App summary generation	ServiceNow Studio
Catalog generation	Catalog Builder
Code generation	ServiceNow IDE
Flow generation	Workflow Studio

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ServiceNow Otto for Creator skills and the applications that they work in (continued)

Generative AI skill	Application
Flow recommendations	Workflow Studio
Flow summarization	Workflow Studio
Playbook generation	Workflow Studio
Playbook recommendations	Workflow Studio
Process Mining	Process Mining
RPA bot generation	RPA Hub
UI generation	UI Builder

Process Mining

Learn about the Process Mining skill available with ServiceNow Otto for Creator.

With Process Mining, you can identify the work notes and generate insights about inefficiencies, challenges, and other reasons that cause reassignment of work. To learn more about Process Mining, see ServiceNow Otto for Process Mining.

RPA bot generation

Learn about the RPA bot generation skill available with ServiceNow Otto for Creator.

With RPA bot generation, you can create automations, activities, and add automation logic from text instructions and preview options. To learn more about RPA bot generation, see Robotic Process Automation (RPA) bot generation skill.

UI generation

UI generation in ServiceNow Otto uses AI-powered skills to help app developers, UI Builder admins, and low-code developers build experiences on the ServiceNow AI Platform® using natural language.

Get started

Explore UI generation



Learn how you can create UI experiences with the power of AI.

Configure UI generation



Configure UI generation.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Using UI generation	Reference
	
Quickly generate a UI with natural language.	Get details about properties, roles, and more.

Important:

- Not all model providers are available for customers with in-country SKUs, and some AI products/features are currently unavailable for in-country customers. For more information, see the KB1584492 article in the Now Support Knowledge Base. Be sure to check for model provider availability updates in future releases.
- Some AI products/features are currently unavailable for customers in the FedRAMP, NSC DOD IL5, or Australia IRAP-Protected data centers, self-hosted customers, or in other restricted environments. For more information, see the KB0743854 article in the Now Support Knowledge Base. Be sure to check for availability updates in future releases.
- Some AI products/features are currently available only for customers in some regions. Be sure to check for availability updates in future releases.
- Some AI products and skills are not available in Regulated Markets. For more information, see KB2593939: Regulated Markets AI Products/Skills Not Available. Be sure to check for availability updates in future releases.

Troubleshoot and get help

- ServiceNow Community
- Search the Known Error Portal for known error articles
- Contact Customer Service and Support

AI limitations

This application uses artificial intelligence (AI) and machine learning, which are rapidly evolving fields of study that generate predictions based on patterns in data. As a result, this application may not always produce accurate, complete, or appropriate information. Furthermore, there is no guarantee that this application has been fully trained or tested for your use case. To mitigate these issues, it is your responsibility to test and evaluate your use of this application for accuracy, harm, and appropriateness for your use case, employ human oversight of output, and refrain from relying solely on AI-generated outputs for decision-making purposes. This is especially important if you choose to deploy this application in areas with consequential impacts such as healthcare, finance, legal, employment, security, or infrastructure. You agree to abide by ServiceNow's AI Acceptable Use Policy, which may be updated by ServiceNow.

Data processing

This application requires data to be transferred from ServiceNow customers' individual instances to a centralized ServiceNow environment, which may be located in a different data center region from the one where your instance is, and potentially to a third-party cloud provider, such as Microsoft Azure. This data is handled per ServiceNow's internal policies and procedures, including our policies available through our CORE Compliance Portal.

Data collection

ServiceNow collects and uses the inputs, outputs, and edits to outputs of this application to develop and improve ServiceNow technologies including ServiceNow models and AI products. In addition, this application will collect information about scripts (and associated script records) in which ServiceNow Otto for code generation is called. Customers can opt out of future data collection at any time, as described in the Now Assist Opt-Out page.

For more information, see the Now Assist documentation.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Exploring UI generation

Learn how UI generation uses AI-powered skills to help app developers, UI Builder admins, and low-code developers build experiences on the ServiceNow AI Platform® using natural language.

UI generation overview

ServiceNow Otto for Creator enables the UI generation feature. With this feature, you can provide a description of the UI experience, and AI-powered ServiceNow Otto will help you get started. Users of all skill levels can benefit from UI generation, making it easier and more efficient to create engaging experiences.

UI generation benefits

The following table summarises the key benefits of UI generation features and how they help different users work more efficiently.

Feature	Benefits	Users
Experience Generation	Improves UI quality and consistency by automating experience creation, reducing manual effort, and clearly identifying AI-generated experiences.	Developers, Administrators
UI Builder Agent	Reduces time spent searching for help by providing instant answers about UI Builder features and documentation.	Developers
Client script summarization	Saves time by automatically explaining what a client script does, making code easier to understand, review, and maintain.	Developers
Data binding generation	Reduces user errors, removes the need to memorize syntax, and speeds up data binding configuration.	Developers
Event handler generation	Simplifies event handler configuration by helping you use plain language instead of filling out forms manually. It manages routing, binds parameters, and sets destinations, so you can focus on creating your experience rather than the technical details.	Developers
Module generation	Speeds up application development by enabling module creation through natural-language descriptions instead of manual configuration.	Developers

Explore the ServiceNow Otto capabilities available in UI Builder, from generating full experiences and modules to configuring data bindings and event handlers using natural language.

Experience Generation

Experience Generation is a ServiceNow Otto capability in UI Generation that uses AI to generate fully functional, multi-page experiences from plain-English descriptions.

Experience Generation removes the need to manually configure navigation, page layouts, data bindings, and chart components when building a new experience. You describe what you want in natural language, and ServiceNow Otto interprets your input, identifies the correct ServiceNow® table, and creates a working experience automatically.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Internally, your description—called an utterance—is submitted to ServiceNow®'s large language model (LLM) alongside purpose-built prompts. The LLM dissects the utterance into key parameters such as experience name, navigation style, page types, and chart configuration. For any details you do not specify, the LLM applies smart defaults to make sure that the result is a valid experience.

Benefits

Fast time to value

A complete multi-page experience is generated from a single plain-English prompt, replacing a process that previously required significant manual effort.

Plain-English input

No deep knowledge of UI Generation configuration is required. You can provide as little or as much detail as you want.

Intelligent inference

The LLM identifies the correct data table, navigation type, page structure, and chart configuration based on your description.

Smart defaulting

When details are omitted from your description, the feature applies smart defaults so that every generation attempt produces a successfully created experience.

Preview before committing

A read-only experience preview lets you verify the generated structure, including navigation, page types, chart parameters, and a live data sample, before saving.

Connected to real data

The generated experience is connected to the identified ServiceNow table from the start, with navigation, record links, and chart data all functional without additional configuration.

Significantly reduced manual effort

Manually building an equivalent experience, with connected pages, data bindings, and chart components, would require considerably more effort and expertise. Experience Generation handles all of this automatically.

Generated experience structure

When ServiceNow Otto generates an experience, it creates the following components:

Experience

A named web interface at a specific URL, with application scope and app shell configured automatically. The experience is tagged as generated by Otto for easy identification.

Navigation

A navigation style that matches your request. For example, tabbed navigation supports working across multiple tasks for short periods of time.

Pages

A common workspace pattern consisting of three interconnected pages:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Landing page: The first page you see when opening the experience. It displays summary widgets such as a total record count and a configurable chart (for example, a pie chart grouped by state).
- Filtered list: Displays records from the selected table with filtered views, sortable columns, and export and create actions.
- Simple record: A detail view that fulfillers use to view, edit, and act on individual records.

Related topics

Create an AI-generated experience

Use case: Create an AI-generated experience

UI Builder Agent

The UI Builder Agent is a ServiceNow Otto AI agent that assists low-code developers working in UI Builder by responding to questions, information requests, and page editing instructions.

The UI Builder Agent is pre-configured and available in AI Agent Studio as part of the UI Generation application. It supports developers as they build and modify pages in UI Builder by providing contextual AI assistance.

The ServiceNow Otto panel displays the UI Builder agent's responses. You can access the ServiceNow Otto panel by selecting the Otto icon in the Next Experience Unified Navigation.

Administrators install the required applications, configure the agent in AI Agent Studio, enable the ServiceNow Otto panel, and activate the default ServiceNow Otto Panel - Platform assistant before developers can interact with the agent.

Required applications

Install the following applications from Application Manager to support the UI Builder Agent:

- ServiceNow Otto for Creator (`sn_now_creator`) — Provides ServiceNow Otto AI skills for creators, including UI Builder support.
- **UI Generation** (`sn_ui_generation`) — Contains the UI Builder Agent definition and its associated tools. Use the Zurich version or later.
- **Conversational Studio** (`sn_convo_studio`) — Provides the assistant infrastructure used by the ServiceNow Otto panel.

Key features of UI Builder agent

The UI Builder agent provides the following features:

Learn about UI Builder

This feature answers your questions about the UI Builder. It provides detailed information about the features and functionalities of the UI Builder, along with links to relevant product documentation.

You can ask about specific features or tasks, such as "What is a data resource?" or "How can I add a component to my page?" You will receive instant responses based on the latest ServiceNow documentation.

Understand page configuration

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

This feature offers insights into how a page is built, including its design and structure. It describes the configuration of various elements on the page, helping developers quickly anticipate how future changes might affect functionality.

You can ask about your page's configuration. For example, you might ask: "Is dynamic theming implemented on my page?" or "Are any client scripts being used on my page?"

Accelerate page configuration

This feature enables you to modify and enhance your page. You can apply different layouts, such as a single row or a three-column format. You can add and configure components, such as buttons with labels and links that direct users to external websites. You can also update the styling of your page, such as changing the background color.

To configure the features, see Configure UI Builder Agent. To use the features, see Using UI Builder agent.

Activation overview

Enabling the UI Builder Agent involves the following high-level steps:

1. Install the required applications from Application Manager.
2. Enable the UI Builder Agent setup wizard in AI Agent Studio.
3. Turn on the ServiceNow Otto panel in AI Admin Hub.
4. Add a display experience to the ServiceNow Otto Panel - Platform (default) assistant, request AI Search activation, and activate the assistant in Assistant Designer.

Related topics

Using UI Builder agent

Use case: Using UI Builder Agent

Client script summarization

Client script summarization uses ServiceNow Otto to create easy-to-understand explanations of client scripts within the UI Builder editor. This feature helps creators to learn what a script does without needing to read the complex code.

How client script summarization works

Client scripts in UI Builder control the behavior of pages and components. Because these scripts can be complex, creators who are unfamiliar with JavaScript or the specific script logic may find it difficult to understand what a script does or how it affects the experience. Client script summarization addresses this by generating an AI-powered explanation on demand, directly within the client script editor panel.

The feature is powered by a ServiceNow Otto skill and is accessed through a dedicated right-hand panel in the **Edit client script** dialog. When a creator selects a client script and requests an explanation, the system sends the script to the skill endpoint and returns a structured summary.

Structure of the generated summary

The summary returned by Otto is organized into two sections:

Summary

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

A concise, plain-language description of what the script does. For example, a script that resets a modal form would be described in terms of its effect on the user experience, without requiring the reader to parse individual API calls.

Code explanation

A more detailed, line-by-line or block-by-block walkthrough of how the script achieves its purpose. This section describes which APIs are used, what each state variable does, and which parameters are or are not used by the script.

Benefits

Client script summarization provides the following benefits:

Faster onboarding

Creators who are new to a project or to JavaScript can quickly learn about the existing scripts.

Reduced context-switching

Explanations appear inside the editor panel, so creators don't have to leave UI Builder to look up documentation.

Improved script literacy

Creators with limited coding experience gain visibility into what scripts do, which helps them make informed decisions about whether to modify or replace a script.

Related topics

Summarize a client script using ServiceNow Otto

Use case: Summarize a client script using ServiceNow Otto

Data binding generation

ServiceNow Otto generates data bindings for component properties using natural language instructions, reducing the need to manually navigate data trees or construct formulas.

When you configure a component property in UI Builder, the **Bind data** dialog includes a **Generate data binding with** *`ServiceNow Otto`* option. You describe the binding you want in plain language, and ServiceNow Otto produces the corresponding formula or data reference, which you can review before applying it.

How data binding generation works

When you submit a natural language instruction, ServiceNow Otto interprets your intent and maps it to the available data sources for the component. The generator has access to the following data source categories shown in the **Data types** panel:

Page properties

The properties defined at the experience or page level.

Data resource

The data resources added to the page, such as a list of records returned from a table.

Client states

The client-side state parameters defined for the experience.

Event payload

The data carried by a component event, available when configuring an event handler.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The **Pill view** and **JSON** tabs let you inspect the data structure of each source before or after generating a binding. The **Formulas** tab shows available formula functions, such as `CONCAT` and `WHERE_GT`, that ServiceNow Otto may use to satisfy your instruction.

Types of generated bindings

Depending on the component property and your instruction, ServiceNow Otto generates one of the following binding types:

Concatenation formula

Combines static text with a data pill. For example, the instruction `Show "Choose your ride" and the user's full name` produces a `CONCAT` formula that joins a string literal with the session user name pill.

Filter formula

Applies a condition to a data resource. For example, the instruction `Return from fleet whose year of manufacture is greater than 2020` produces a `WHERE_GT` formula on the fleet data resource, filtering the result set to records whose `year` field exceeds 2020.

Direct data pill reference

References a single field from a data source without transformation. For example, the instruction `Use the pillId from event payload` resolves to the `pillId` pill from the event payload data source.

Suggested prompts

The **Generate data binding** panel displays suggested prompts based on the component's context and the data sources available on the page. You can select a suggestion to populate the instruction field, then submit it as-is or modify it before generating. The panel also includes a **Show more** option to display additional suggestions.

Reviewing generated bindings

After ServiceNow Otto generates a binding, a preview appears in the component's property panel before the binding is applied. You can inspect the formula or data pill reference and close the dialog without applying it if the result doesn't match your intent. You can also stop an in-progress generation by selecting **Stop generating**.

Related topics

Generate a data binding and formula

Use case: Generate a data binding and formula

Event handler generation

ServiceNow Otto event handler generation enables you to configure event handlers for UI components using natural language, eliminating the need for manual configuration of complex routing parameters and binding expressions.

In UI Builder, event handlers define how a component responds to user interactions, such as a button select or a reference link selection. Configuring these handlers manually requires knowledge of specific parameters, page routing structures, and binding syntax — all of which can be time-consuming and error-prone, particularly for developers who are new to the platform.

With the event handler generation, you describe the desired behavior in plain language. ServiceNow Otto interprets the prompt, identifies the correct destination page or URL, maps any required parameters, and generates a complete configuration. You review the result before applying it, and you can edit any field using the standard form controls before saving.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Benefits

The event handler generation provides the following benefits:

Faster configuration

Natural language input replaces manual form-filling, significantly reducing the time needed to set up event handlers.

Reduced complexity

ServiceNow Otto interprets intent and populates routing parameters, destination pages, and bindings automatically.

Intelligent parameter binding

ServiceNow Otto can bind page parameters dynamically from state parameters or event payloads, enabling context-aware navigation without manual binding expressions.

Contextual sample prompts

The ServiceNow Otto panel displays relevant sample prompts based on the pages and components available in the current experience, so you can start quickly without composing a prompt from scratch.

Human review before applying

Generated configurations are presented as a read-only preview before being committed. Developers can accept and edit, or reject and re-prompt, ensuring accuracy at every step.

How ServiceNow Otto generates event handler configurations

When you add an event handler to a component in UI Builder and open the ServiceNow Otto configuration panel, the system presents a text field where you describe the behavior you want. You can enter your own prompt or select from the sample prompts displayed.

ServiceNow Otto processes the prompt against the available pages, parameters, and event payloads in the current experience and generates a configuration. The result is displayed as a read-only preview. You review the destination, parameter bindings, and trigger settings, then choose to accept and edit, or reject the result.

If you accept, the configuration is applied to the component. You can continue editing individual fields using the standard form controls before selecting **Add** to save the handler.

Supported event handlers

The event handler generation supports the three most commonly used event handlers in UI Builder:

Open page or URL

The Open page or URL event handler navigates the user to a page when a component event triggers. The destination can be a page within the current experience or an external URL.

For pages in the current experience, you select the target page and provide any required parameters, such as a table name or sys_id. For external URLs, you enter the full URL. Both destination types support an **Open in new tab** option.

Parameters can be bound statically or dynamically from state parameters or event payload fields. Advanced mode exposes additional binding fields for route, fields, and params.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Open and close modal dialog

The Open and close modal dialog event handler opens or closes a modal dialog in the current experience when a component event triggers. ServiceNow Otto can configure this handler based on a natural language description of which dialog to open or close and under what conditions.

Viewport load requested

The Viewport load requested event handler triggers the loading of a specific viewport within the experience when a specified event triggers. ServiceNow Otto can configure this handler based on a natural language description of the target viewport and the triggering event.

Event payloads and parameter binding

Some component events dispatch a payload — a set of data values emitted by the source component when the event triggers. When a payload is available, ServiceNow Otto can bind destination parameters directly to payload fields, enabling fully dynamic navigation without any static values.

For example, the Reference link clicked event on a list component dispatches the following payload fields: `listTable`, `nativeEvent`, `record_id`, `row`, `sys_id`, and `table`. ServiceNow Otto can map these fields to the corresponding parameters of a record page destination automatically, so that selecting any reference link in the list opens the correct record.

When an event has no payload — such as the Button clicked event — parameters must be bound from state parameters or entered as static values.

Related topics

Configure an event handler with ServiceNow Otto

Use case: Configure an event handler with ServiceNow Otto

Module generation

ServiceNow Otto generates application modules in UI Builder workspaces using natural-language prompts.

Module generation overview

With the introduction of the ServiceNow Otto module generation feature, you can describe the modules you need through a prompt instead of manually creating and configuring each module in UI Builder.

The AI agent processes your prompts and generates various modules, including lists, records, URLs, scripts, dashboards, and folders.

Module types

The AI agent supports creating the following module types:

Module types

Type	Description
List	Displays records from a table with optional filters and views.
Single record	Opens a specific record form.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Module types (continued)

Type	Description
New record	Creates a record with optional role restrictions and views.
URL	Opens external or internal links with arguments and window configuration.
Script	Executes defined scripts with arguments and window settings.
Dashboard	Displays dashboard components.
Folder	Organizes modules hierarchically.

For more information about the module types, see Application Navigator category.

Related topics

Generate modules using ServiceNow Otto

Use case: Generate modules using ServiceNow Otto

General guidelines UI generation

Use these general guidelines for UI generation to get better experiences.

Writing prompts

Write clear and specific but concise prompts

Describe the table and navigation your users will be working with. For best results, include as much detail as possible.

Experiment with different prompts

As you refine and experiment, the Now LLM Service uses this feedback to learn and improve.

- Try adding a specific title for your desired experience.
- Try including navigation details.
- Keep track of your prompts, including any modifications, and instructions for generating prompts to meet your specifications. This tracking enables easy regeneration of past results for comparative analysis.

Example prompts for UI generation

Strong prompt	Weak prompt	Notes
Create a Tabbed experience called, "Incident Management" to help track Incidents by Priority.	Track incidents.	Includes experience title details.
Create a Breadcrumb style experience called, "Change Request Tracking" that uses a Pie chart to group Change Requests by State.	Change request pie chart.	Includes navigation details.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example prompts for UI generation (continued)

Strong prompt	Weak prompt	Notes
Create a Tabbed experience to allow user to see Requested Items trending by Created Date in a Line Chart.	Requested items.	Includes chart details.
Create a Breadcrumb navigation experience called, "Task Tracker" with a Bar chart showing Tasks in a Bar Chart Grouped By Assigned to.	Task tracker bar chart.	Includes Grouped by details.

Reviewing generated experience

Review experience

Implement strict and detailed reviews of the AI-generated experience to determine its accuracy, efficiency, and how well it adheres to your standards.

Test experience

Validate the experience by running it against test cases in controlled environments to ensure that it functions according to your requirements.

Configuring UI generation

Configure and install UI generation.

Install UI generation

Install the ServiceNow Otto for Creator application from the ServiceNow® Store to enable UI generation.

Before you begin

Review the ServiceNow Otto for Creator application listing in the ServiceNow Store for information on dependencies, licensing or subscription requirements, and release compatibility. ServiceNow Otto for Creator installs the ServiceNow Otto for UI generation application.

Role required: admin

Procedure

1. From the ServiceNow Otto for Creator application page on the ServiceNow Store, select **Buy**.
2. After approval is granted, on your instance, navigate to **All** > **System Applications** > **All Available Applications** > **All**.
3. Using the search bar, search for the ServiceNow Otto for Creator application (sn_now_creator).
4. Select **Install**.
5. Confirm that ServiceNow Otto for Creator is installed:
   a. Navigate to **All** > *AI Admin Hub* > **Skills**.
   b. In the workflow list, select **Creator**.
   c. On the UI Generation card, confirm that the Experience Generation skill is active.
   For more information about AI Admin Hub, see AI Admin Hub.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

What to do next
Grant the ui_builder_admin role to users who need UI generation access. For more information, see Grant UI Builder admin role.

Note:

You can use Now LLM Service, Azure OpenAI, Google Gemini or Anthropic Claude on AWS as the AI model provider for all generative AI skills and AI agents. Use the Configuration Controls in AI Control Tower to define which options are available, then set the skill-level preferences in the AI Admin Hub console. For more information, see Large language models on the ServiceNow AI Platform.

Related topics

Install ServiceNow Otto for Creator

Grant UI Builder admin role

Assign the UI Builder administrator role to a user by editing the user record and adding the ui_builder_admin role.

Before you begin
Make sure that you install UI generation and that you have the ui_builder_admin role. For more information, see Install UI generation and Grant UI Builder admin role.

Configure UI Builder agent. For more information, see Configure UI Builder Agent.

Role required: admin

Procedure

1. In the application navigator, navigate to **All** > **User Administration** > **User**.
2. Open the user record for the user you want to grant the role to.
3. In the **Roles related** list, select **Edit**.
4. From the **Collection** list, select ui_builder_admin, and then select  .
5. Select **Save**.

Result
The user is granted the `ui_builder_admin` role and can access UI generation features.

To grant this role to additional users, repeat this procedure for each user.

Configure UI Builder Agent

The UI Builder Agent is a AI agent that assists low-code developers working in UI Builder by responding to questions, information requests, and page editing instructions.

Before you begin
Role required: ui_builder_admin

About this task
Administrators must install the required applications, configure the agent in AI Agent Studio, enable the ServiceNow Otto panel, and activate the default ServiceNow Otto Panel- Platform assistant before developers can interact with the agent.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Install required applications for UI Builder Agent
2. Enable UI Builder Agent in AI Agent Studio
3. Enable the ServiceNow Otto panel
4. Activate the ServiceNow Otto Panel- Platform (default) assistant

Install required applications for UI Builder Agent

Install ServiceNow Otto for Creator, UI Generation, and Conversational Studio from Application Manager before configuring the UI Builder Agent.

Before you begin
You must have administrator access to Application Manager on your ServiceNow instance.

Role required: ui_builder_admin

About this task
Install and update three applications to support the UI Builder Agent before configuring it in AI Agent Studio.

Procedure

1. Navigate to **All** > **Application Manager**.
   Application Manager opens and displays the search interface for licensed applications and plugins.
2. In the search field, type *`ServiceNow Otto for Creator`* and press Enter.
3. Select the **Installed** tab and verify that ServiceNow Otto for Creator (`sn_now_creator`) is listed and shows **Up to date**.
   If the application is not installed, locate it under the **Available for you** tab and install it.
4. Clear the search field, type `ui generation`, and press Enter.
5. Select the **Installed** tab and verify that UI Generation (`sn_ui_generation`) is installed and shows **Up to date**.
   The UI Generation application must be the Zurich version or later.
6. Clear the search field, type `Conversational Studio`, and press Enter.
7. Select the **Installed** tab and verify that Conversational Studio (`sn_convo_studio`) is installed and shows **Up to date**.
   If the application is not installed, locate it under the **Available for you** tab and install it.

Result
All three applications — ServiceNow Otto for Creator, UI Generation, and Conversational Studio — are installed and up to date on your instance.

Enable UI Builder Agent in AI Agent Studio

Follow the UI Builder Agent setup wizard in AI Agent Studio to review and confirm your agent configuration.

Before you begin
Verify that you have installed all the required applications. See Install required applications for UI Builder Agent.

Role required: ui_builder_admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

About this task

The UI Builder Agent comes pre-configured with the UI Generation application. The setup wizard in AI Agent Studio guides you through the process of defining the agent, covering aspects such as its specialty, tools, security controls, triggers, and channels.

> **Important:** The UI Builder Agent record is read-only in the Global application scope. When viewing the agent in AI Agent Studio, you will see a banner indicating that the record belongs to the UI Generation application. You can proceed through the wizard without switching scope.

Procedure

1. Navigate to **All** > **AI Agent Studio** > **Overview**.
   The AI Agent Studio overview page opens, showing available AI agents and recent agentic workflow activity.
2. Locate the **UI Builder Agent** card and select it.
   The UI Builder Agent setup wizard opens on the **Define the specialty** step.
3. Review the agent specialty definition and select **Continue**.
   You don't need to modify the specialty or subsequent configuration steps. The agent comes fully configured.
4. On the **Add tools and information** step, select **Continue**.
5. On the **Define user access** step, select **Continue**.
6. On the **Define data access** step, review the user identity settings and select **Continue**.
   The agent is pre-configured with a **Dynamic user** identity type and the `ui_builder_admin` approved role. The dynamic user setting means the agent derives its data access permissions from the user who initiates the interaction.
7. On the **Add triggers** step, select **Continue**.
8. On the **Select channels and status** step, scroll down to **Activation status**, enable **This AI agent is active**, and then select**Test**.
   The system navigates to the Testing tab in AI Agent Studio. An alert banner appears indicating that you must enable the ServiceNow Otto panel before testing.

Result

The UI Builder Agent setup wizard is complete. The agent configuration is confirmed, and you can enable the ServiceNow Otto panel to proceed with testing and activation.

Enable the ServiceNow Otto panel

Enable the ServiceNow Otto panel in AI Admin Hub.

Before you begin

Role required: ui_builder_admin

About this task

Enable the panel before testing or using the UI Builder Agent.

Procedure

1. On the AI Agent Studio Testing tab alert banner, select the **Edit this setting** link.
   Alternatively, navigate directly to **All** > *AI Admin Hub* > **Experiences**.
   The ServiceNow Otto panel page opens in AI Admin Hub, showing a summary of the panel and a **Turn on** button.
2. In the **Turn on** section, select **Turn on**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

A success dialog appears confirming that the ServiceNow Otto panel is turned on. The dialog shows that generative AI skills in the ServiceNow Otto panel are available. It also provides a link to the Assistants page in CI Admin console for further configuration.

3. In the success dialog, select the **Assistants page in CI Admin console** link.
   The CI Admin Experience Assistants page opens.

Result
The ServiceNow Otto panel is enabled on your instance and is available to users based on their designated permissions.

Activate AI search

Activate the AI Search.

Before you begin
Role required: ui_builder_admin

About this task
Activating AI Search turns on ServiceNow Otto in Virtual Agent.

Procedure

1. Go to **All** > **AI Search** > **AI Search Status**.
2. Select **Request AI Search**.

   AI Search activation is a background process that takes about 15 minutes to complete. A confirmation page titled **AI Search is requested** appears when the request is submitted successfully.

   A message confirms that the AI Search is activated.

Result
You can now access AI-powered search capabilities in Virtual Agent.

Activate the ServiceNow Otto Panel- Platform (default) assistant

Add a display experience to the ServiceNow Otto Panel- Platform (default) assistant, request AI Search activation, and activate the assistant in Assistant Designer.

Before you begin
Role required: ui_builder_admin

About this task
The ServiceNow Otto Panel - Platform (default) assistant is the main system assistant in the Unified Navigation app shell. To use the UI Builder Agent through the ServiceNow Otto panel, you must assign at least one display experience to the assistant and turn it on.

Procedure

1. On the CI Admin Experience Assistants page, select **Manage assistants**.
   Assistant Designer opens, showing all available assistants including ServiceNow Otto Panel-Developer (default), ServiceNow Otto in Catalog Builder, and ServiceNow Otto Panel- Platform (default).
2. On the ServiceNow Otto Panel- Platform (default) card, select **Edit**.
   A banner on the assistant overview page shows that the assistant is inactive and requires at least one display experience to be activated.
   The ServiceNow Otto Panel- Platform (default) assistant opens in Assistant Designer.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

3. Select **Go to display experiences**.
   The **Display experiences** settings page opens.
4. Select **Add ServiceNow platform** and select **Unified Navigation app shell**.
5. Select **Add**.
   The Unified Navigation app shell entry appears in the ServiceNow AI Platform® list with the **Enhanced chat** experience type.
6. Select **Save**.

   A success banner confirms that the ServiceNow Otto Panel- Platform (default) assistant is saved successfully.

   The **Activate** button is enabled.
7. Select **Activate**.
   If you haven't activated AI Search, an alert banner appears indicating that AI Search must be activated to turn on ServiceNow Otto in Virtual Agent. To activate AI Search, see Activate AI search.

   A success banner confirms that the ServiceNow Otto Panel- Platform (default) assistant is activated.

   The status of ServiceNow Otto Panel- Platform (default) assistant in Assistant Designer changes from **Inactive** to **Active**.

Result

The ServiceNow Otto Panel- Platform (default) assistant is active.

The UI Builder Agent is available to users through the ServiceNow Otto panel in UI Builder and across the platform.

Using UI generation

UI generation in ServiceNow Otto for Creator enables developers to build ServiceNow portal experiences faster using natural language. Instead of manually configuring each page, module, or data binding, you describe what you need, and ServiceNow Otto for Creator it for you.

When UI generation is enabled, the option to create an AI-Generated experience (✨) appears in the UI Builder dialog.

Learn how to use ServiceNow Otto for Creator in UI Builder to generate experiences, configure components, and accelerate development using natural language.

Create an AI-generated experience

Use ServiceNow Otto in UI Builder to generate a multi-page experience from a natural language description of what you want to build.

Before you begin

Make sure that you install UI generation and that you have the ui_builder_admin role. For more information, see Install UI generation and Grant UI Builder admin role.

Role required: ui_builder_admin

About this task

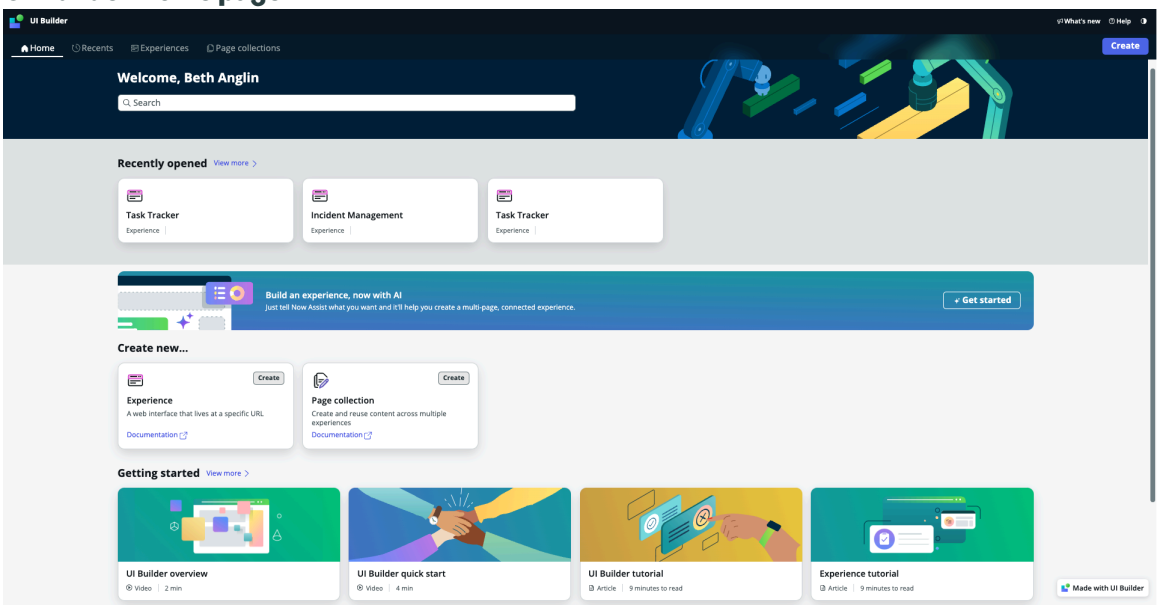
You can access the experience generation feature from the UI Builder homepage. You describe the experience you want in natural language—including the name, table, navigation style, and any chart preferences. ServiceNow Otto generates a fully connected experience for you to review and save.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

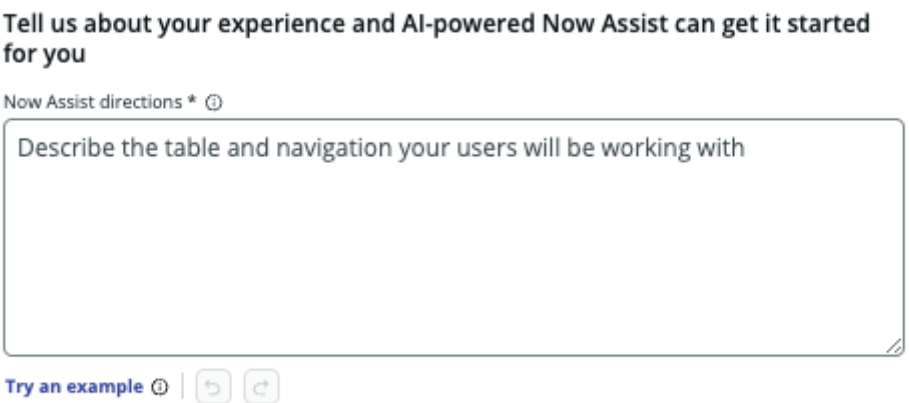
1. In the application navigator, navigate to **All** > **Now Experience Framework** > **UI Builder**.

UI Builder Home page



2. Select **Create** in the upper-right corner of the page.
3. Select **Experience** from the menu that appears.
   A dialog titled *`Create experience with Now Assist?`* appears, presenting two options: **AI-Generated experience** and **Continue with Now Assist**.
4. Select **AI-Generated experience**.
   The Experience Generation interface opens. The interface is split into two panels: a left panel containing the **Now Assist directions** input field, and a right panel that displays guidance on what information to provide.

Create an experience form



5. In the **Now Assist directions** field, type a description of the experience you want to create.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Write your description in plain language. Your description can include any combination of the following details:

- The name of the experience
- The ServiceNow table your users will work on
- The navigation style (for example, tabbed navigation)
- Chart type and data grouping preferences (for example, a pie chart grouped by state)

You can provide as little or as much detail as you want. The LLM infers any details you omit and applies smart defaults to produce a complete, valid experience.

Tip: If you're unsure what to write, select **Try an example** to populate the field with a sample description.

Example description:

```
Create an experience called "Change Request Tracker" for
 tracking Change
Requests, that uses a tabbed navigation, and that shows a Pie
 chart that
groups the data by "state".
```

6. Select **Generate preview**.
   Your description is submitted to ServiceNow's LLM. A *Generating preview* indicator appears in the right panel while the experience configuration is being processed. The right panel header changes to **Experience preview (Read-only)** and shows a **Generated by Now Assist** label when generation is complete.

Result

The experience preview appears in the right panel. It displays the proposed experience name, navigation type, page structure, chart parameters, and a live data sample from the identified ServiceNow table. The **Regenerate preview** and **Save and edit experience** buttons are set to active.

Related topics

Experience Generation

Use case: Create an AI-generated experience

Review and save an experience preview

After ServiceNow Otto generates an experience, review the read-only preview to verify the proposed configuration before saving and opening the experience in UI Builder.

Before you begin

Role required: ui_builder_admin

About this task

The experience preview is a structured summary of what ServiceNow Otto proposes to create. It shows the LLM's interpretation of your natural language description. It shows the experience name, navigation style, page types, chart parameters, and a live data sample from the identified ServiceNow table. You can save the experience as-is or update your description and regenerate the preview.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. In the **Experience preview** panel, review the experience name at the top of the panel.
   The experience name is derived directly from your description. Verify that it matches your intent before saving.
2. Review the **Navigation** section to confirm the navigation style is correct.
   For example, **Navigation: Tabs** indicates that the experience uses a tabbed navigation style, which supports working across multiple tasks for short periods of time.
3. Review the **Describe the parameters for your pages** section to verify the page types and chart parameters.
   a. Check the parameter tags displayed at the top of the section (for example, **Chart type: pie**, **Chart title**, and **Group by: state**) to confirm the chart configuration is correct.
   b. Verify that the three page type cards—**Landing page**, **Filtered list**, and **Simple record**—are shown, confirming that a complete multi-page workspace structure will be created.
4. Review the **Table** section to confirm that ServiceNow Otto identified the correct ServiceNow table.
   The table name and its system name (for example, **Change Request (change_request)**) are displayed along with a live sample of records from that table. Use the data sample to verify that the correct table has been selected.
5. Do one of the following based on your review:
   - If the preview matches your intent, select **Save and edit experience** to create the experience and open it in UI Builder.
   - If the preview does not match your intent, update the text in the *Now Assist* **directions** field on the left panel and select **Regenerate preview** to generate a new preview with your updated description.

Result

After selecting **Save and edit experience**, UI Builder creates the experience and opens the page list view. The page list shows the experience name, URL path, application scope, app shell, and three generated pages: Home (landing page), List, and Record. A banner at the top confirms that ServiceNow Otto generated the experience and prompts you to check it for accuracy.

Verify the generated experience

After saving an AI-generated experience, open it in a browser to confirm that all pages, navigation, charts, and record links are functioning correctly.

Before you begin

Role required: ui_builder_admin

About this task

When ServiceNow Otto saves a generated experience, it creates three connected pages and wires them to the selected ServiceNow table. Before distributing the experience to users, verify that each page loads correctly and that navigation between pages works as expected. A banner in UI Builder reminds you that the experience was generated by ServiceNow Otto and may require review before use.

Procedure

1. From the experience page list in UI Builder, select the URL path link displayed under the experience name to open the experience in a new browser tab.
   The landing page of the experience opens. It displays the experience title, a record count widget, a pie chart (or other configured chart) grouped by the requested field, and a list of records from the selected table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

2. On the landing page, verify the following:
   a. Confirm that the experience title displayed at the top of the page matches the name you requested.
   b. Confirm that the chart is present and displays data grouped by the correct field (for example, grouped by state).
   c. Select a slice on the pie chart and confirm that the list only shows the records matching that slice's value.
   d. Confirm that the record list displays columns and data from the correct table.
3. In the left navigation sidebar, select the link to the list page to navigate to the filtered list view. The list page opens, showing all records from the selected table with sortable columns, filter controls, and action buttons including **Edit**, **Export**, and **New**.
4. On the list page, select a record number link to open a specific record.
   The record page opens, displaying the full detail view for that record, including all field sections, tabs, and the activity panel. A breadcrumb tab showing the record number appears in the tab bar, confirming that tabbed navigation is working.
5. On the record page, select the plus sign (**+**) in the tab bar to verify that the create form for a new record opens correctly.

Result

If all pages load correctly, display the expected data, and navigation between pages works as expected, the generated experience is ready for further customization or distribution. If any element is missing or incorrect, open the experience in UI Builder to make edits, or return to the Experience Generation interface to regenerate with a revised description.

Trouble?

Troubleshooting Experience generation issues

Symptom	Cause	Resolution
AI-Generated experience option not visible in UI Builder.	The `ui_builder_admin` role is not assigned, or the Experience Generation skill is not active.	Assign the ui_builder_admin role to the user. Navigate to *AI Admin Hub* > **Features** > **Creator** and verify that the Experience generation skill is active.
Generated experience does not match the prompt	Prompt lacks sufficient detail about table, chart type, or navigation structure.	Include experience name, chart type, navigation style, and table name in the prompt. Use **Try an example** to understand expected prompt structure.

Using UI Builder agent

Use the UI Builder agent to learn about features and functionality, understand your page setup, and accelerate page configuration.

The UI Builder agent introduces key features to enhance user experience. It offers an informative resource to answer questions about the features and functionality of UI Builder and provides links to relevant documentation. Additionally, it gives an overview of page configuration, helping developers understand the structure and interconnections of page elements. It also streamlines

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

the process of modifying and enhancing pages by enabling you to apply different layouts, configure components, and update styles.

Agentic workflows and their AI agents use role masking to determine which users can access them. Ones installed with your applications have specific roles that come included with the application. If you select *`Users with specific roles`* for user access, you must configure the security controls to include these roles. For the instructions to change the security controls, see Define security controls for an agentic workflow.

Learn about UI Builder

Use this feature to learn about UI Builder.

Before you begin

Install ServiceNow Otto for UI generation application. For more information, see Install UI generation.

Role required: ui_builder_admin

About this task

This feature answers your questions about the UI Builder. It provides information about the features and functionalities of the UI Builder. Additionally, it includes links to relevant product documentation.

Procedure

1. Navigate to **All** > **UI Builder**
2. Create a new experience or navigate to an existing one.
   For more information about creating an experience, see Create an AI-generated experience.
3. Create a new page or navigate to an existing one.
4. From the page editor, open the ServiceNow Otto panel.
5. In the chat panel, ask a question about UI Builder.

   Tip: You can ask about specific features or tasks, such as `What is a data resource?`, or `How can I add a component to my page?`

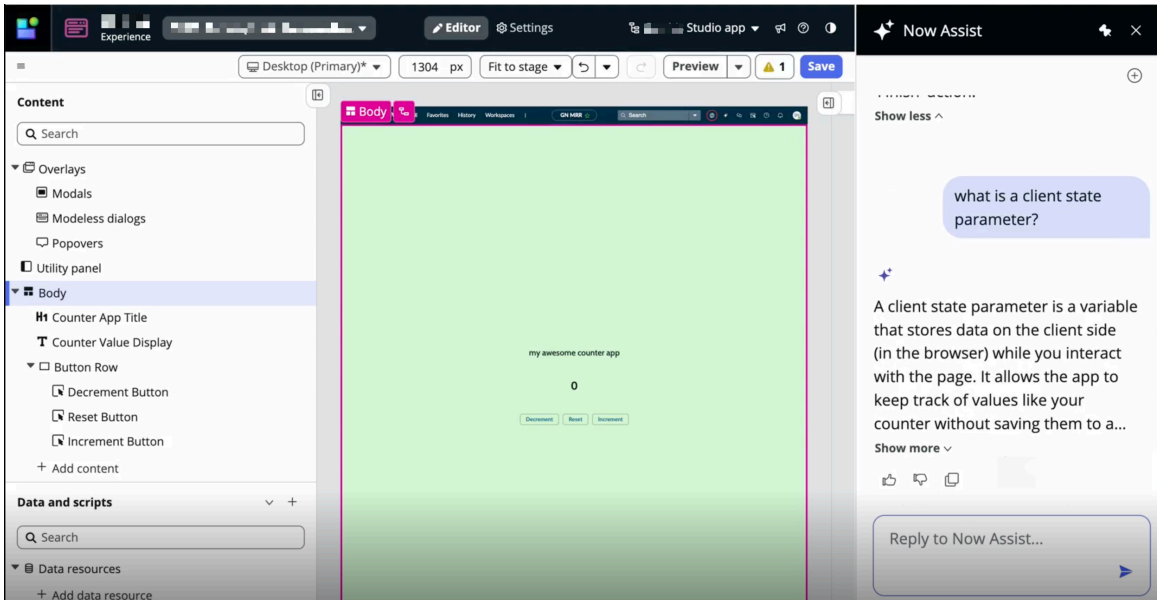
   You receive instant responses based on the latest ServiceNow documentation.

Result

The UI Builder agent processes your question and provides answers, along with relevant documentation links.

For example:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Note: The UI Builder agent operates only within UI Builder.

Understand page configuration

Use this feature to understand the architecture and functionality of a page and its elements.

Before you begin

Install ServiceNow Otto for UI generation application. For more information, see Install UI generation.

Role required: ui_builder_admin

About this task

This feature provides insights into how a page is constructed, including its design and structure. It can describe how various elements on the page are configured. This feature helps developers to quickly anticipate how future changes might impact functionality.

Procedure

1. Navigate to **All** > **UI Builder**
2. Create a new experience or navigate to an existing one.
   For more information about creating an experience, see Create an AI-generated experience.
3. Create a new page or navigate to an existing one.
4. From the page editor, open the ServiceNow Otto panel.
5. In the chat panel, enter your question.

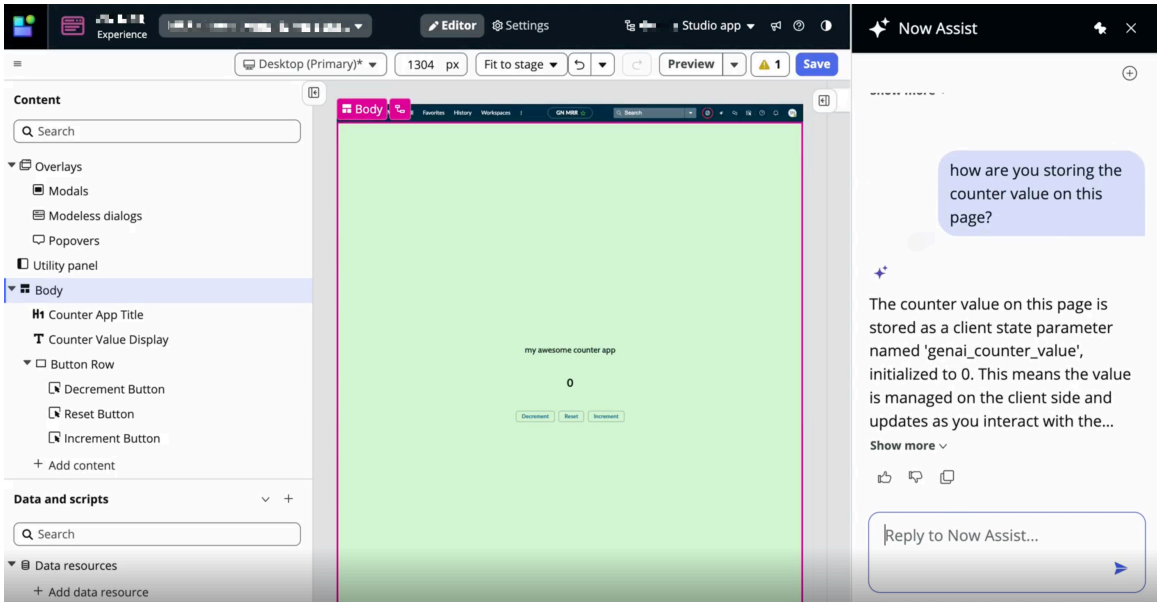
Tip: You can inquire about the configuration of your page. For instance, you might ask, `Is dynamic theming implemented on my page?`, or `Are any client scripts utilized on my page?`.

Result

Based on your request, the UI Builder agent provides a detailed explanation of the page configuration.

For example:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Note:

- The UI Builder agent operates only within UI Builder.
- This feature is available only when you're viewing a page in the page editor.

Accelerate page configuration

Use this feature to build and modify your page.

Before you begin

Install ServiceNow Otto for UI generation application. For more information, see Install UI generation.

Role required: ui_builder_admin

About this task

This feature enables you to easily modify and enhance your page. You can apply different layouts, such as a single row or a three-column format. Additionally, you can add and configure components, such as buttons with labels and links that direct users to external websites. You can also update the styling of your page, such as changing the background color.

Procedure

1. Navigate to **All** > **UI Builder**
2. Create a new experience or navigate to an existing one.
   For more information about creating an experience, see Create an AI-generated experience.
3. Create a new page or navigate to an existing one.
4. From the page editor, open the ServiceNow Otto panel.
5. In the chat panel, request a page modification, and then select **Send**.

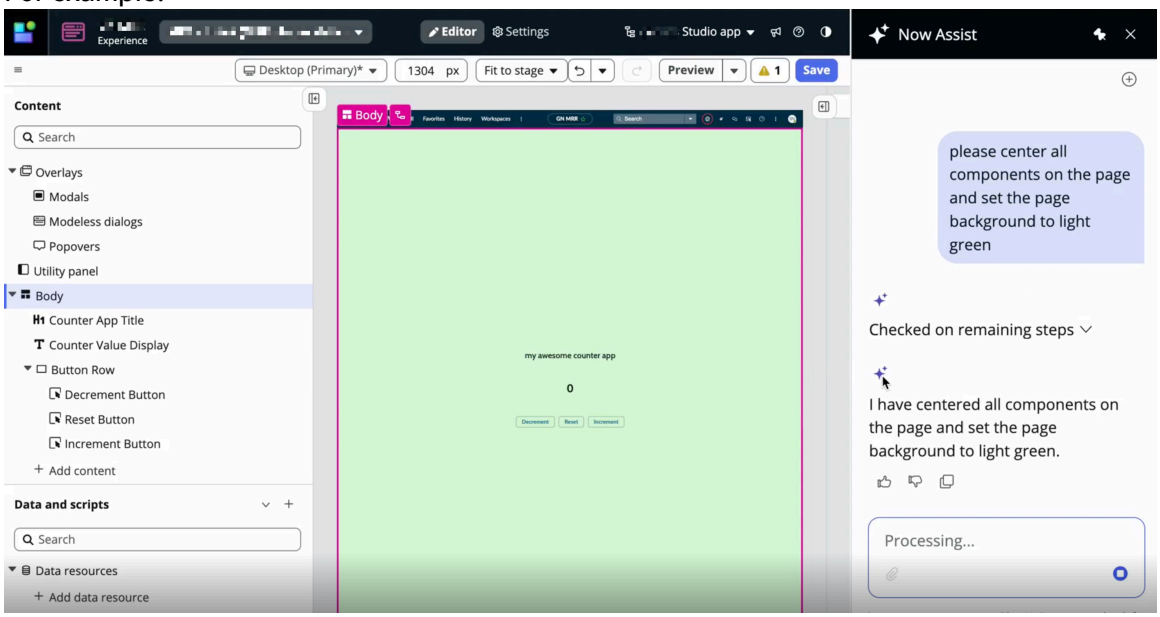
Tip: You can request to apply layouts (such as single-row or three-column), add components (such as buttons with labels and links to external websites), and update the page styles (for example, changing the background color).

Result

UI Builder agent implements the modifications and enhancements to the design and functionality of your page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For example:



Note:

- The UI Builder agent operates only within UI Builder.
- This feature is available only when you're viewing a page in the page editor.

Trouble?

Troubleshooting UI Builder Agent issues

Symptom	Cause	Resolution
UI Builder Agent does not respond	The UI Builder Agent skill is disabled, or the user is outside of UI Builder.	Enable the UI Builder Agent in AI Admin Hub. The agent operates only within UI Builder and is not available in other areas of the platform.

Related topics

UI Builder Agent

Use case: Using UI Builder Agent

Summarize a client script using ServiceNow Otto

Use the client script summarization skill to generate a plain-language summary of a client script directly in the UI Builder editor.

Before you begin

Make sure that you install UI generation and that you have the ui_builder_admin role. For more information, see Install UI generation and Grant UI Builder admin role.

Turn on the Client script summarization skill.

Role required: ui_builder_admin

About this task

Client scripts can be long and complex. The Explain code with ServiceNow Otto panel appears on the right side of the Edit client script dialog and generates a structured explanation of the

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

selected script on demand. You can request a new explanation if the first response does not meet your needs.

Procedure

1. In the application navigator, navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience with a client script.
3. Select a page to open the page editor.
4. From the Data and scripts panel, open a client script.
   The Explain Code panel appears if you have the ui_builder_admin role and have enabled the Client script summarization skill.
5. Select **Explain this code**.

   ServiceNow Otto sends the script to the skill endpoint. When processing is complete, the panel displays two sections:

   - Summary: A concise description of what the script does.
   - Code explanation: A detailed explanation of how the script works, including which APIs are used and what each state variable does.

   Note: If you change the script, a message in the **Explain code panel** notifies you of the update and asks whether the code should be explained again. Select **Explain again** to generate a new summary.

6. **Optional:** If the generated summary does not provide sufficient clarity, select the regenerate icon to request a new response from ServiceNow Otto.
   The model generates an alternative explanation for the same script.
7. Review the summary and code explanation, then select **Cancel** or **Apply** as appropriate to close the dialog.

Result

A concise summary, information about where the client script is triggered, and a detailed explanation of the script's functionality are displayed. Use this information to decide whether to keep, modify, or replace the script.

Trouble?

TroubleshootingClient script summarization issues

Symptom	Likely Cause	Resolution
Explain Code panel does not appear	Client script summarization skill is disabled, or user lacks the ui_builder_admin role.	Activate Client script summarization skill in AI Admin Hub. Confirm the user has the ui_builder_admin role.

Related topics

Client scripts

Client script summarization

Use case: Summarize a client script using ServiceNow Otto

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Generate a data binding and formula

Automatically generate a data binding and formula using natural language with correct syntax.

Before you begin

1. Install UI Generation application (28.1 – Zurich or later). For more information, see Install UI generation.
2. Make sure that you have the ui_builder_admin role. For more information, see Grant UI Builder admin role.
3. Navigate to **All** > *AI Admin Hub* > **Creator** and activate the Data Binding Generation skill.
4. Select a large language model (LLM) provider.

Role required: ui_builder_admin

About this task

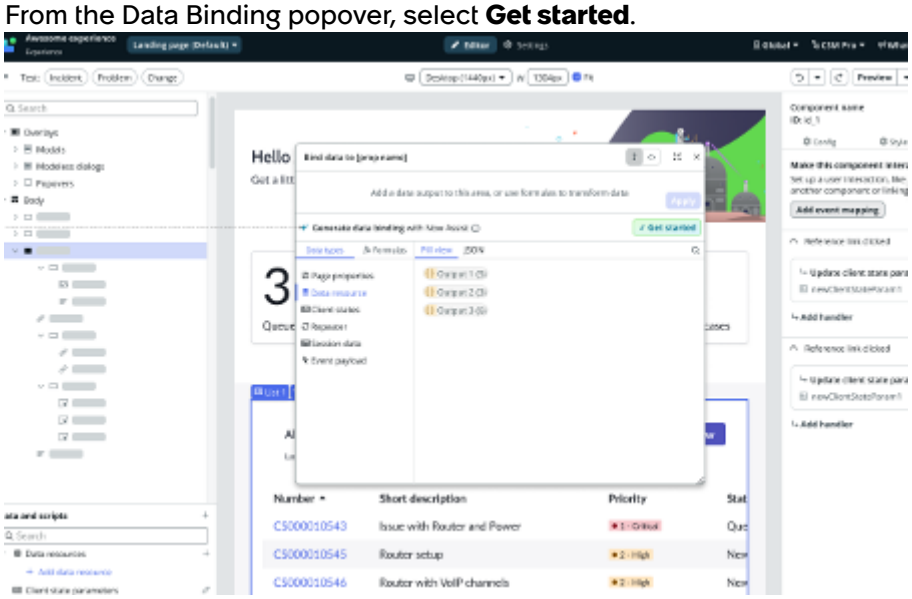
Learn to use Data Binding Generation in ServiceNow® UI Builder with ServiceNow Otto to automatically generate bindings and formulas. These bindings can be applied to any component property, such as labels, text, lists, or buttons, and can include logic, calculations, or data transformations.

Procedure

1. Open UI Builder.
2. Create an experience or open an existing one.
   For more information, see Create an experience for UI Builder.
3. Create a page or open an existing one.
   For more information, see Create a page in UI Builder.
4. Create a page variant or open an existing one.
   For more information, see Create a page variant.
5. Open the Page Editor.
6. Add a data resource or open an existing one.

   For more information, see Add and configure data resources to a page.

7. From the Data Binding popover, select **Get started**.

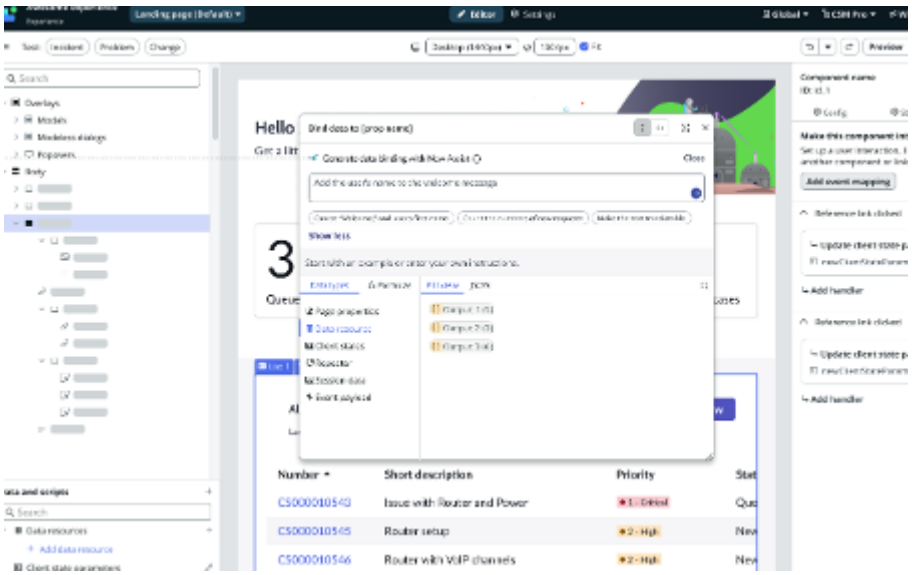


© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

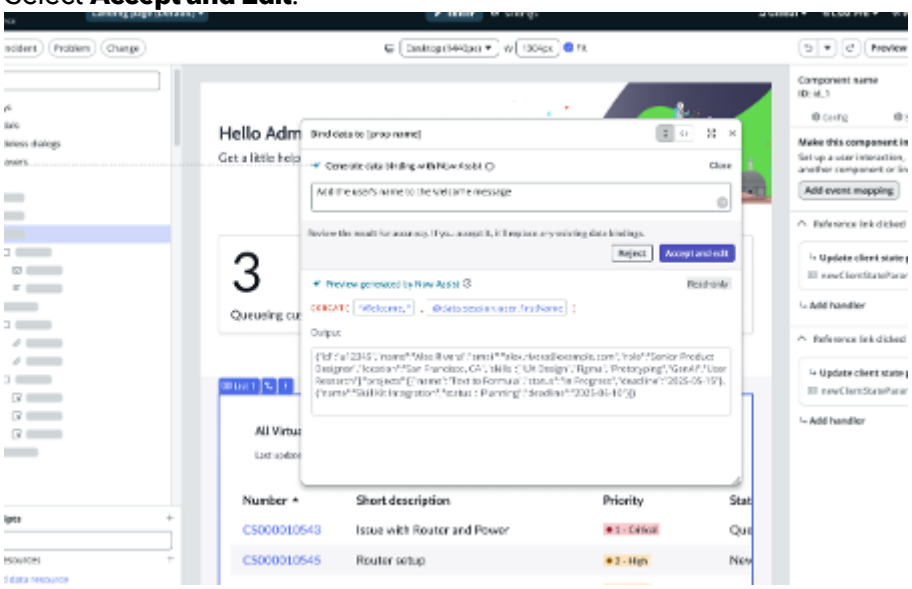
8. You can enter your prompt using natural language or select one from the examples.

   Note: For example:

   - Instead of writing `IF(@context.propers.bare,"bare page","not bare page")`, you can enter `If the page is bare, show 'bare page'; otherwise, show 'not bare page'.`
   - Instead of `LEN([1,2,3]), "")`, you can enter `How many items are in this list?`



9. Select the arrow to generate a binding or formula.
10. Review the syntax and output.
11. Select **Accept and Edit**.



12. Apply or reject the binding.
    - If you're happy with the output, select **Apply**.
    - If you're unhappy with the output, select **Reject** and start over by entering a new prompt.
13. Preview and test the output.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Result
Data binding and formula are generated.

Trouble?

Troubleshooting Data binding generation issues

Symptom	Cause	Resolution
Data binding generation returns incorrect syntax	The prompt is ambiguous or uses terms not recognized as synonyms.	Refine the prompt using more specific language. Refer to the Formula Operators table for supported synonyms. Try selecting an example from the Data Binding popover.

Related topics

Natural language synonyms for formula operators

Configure an event handler with ServiceNow Otto

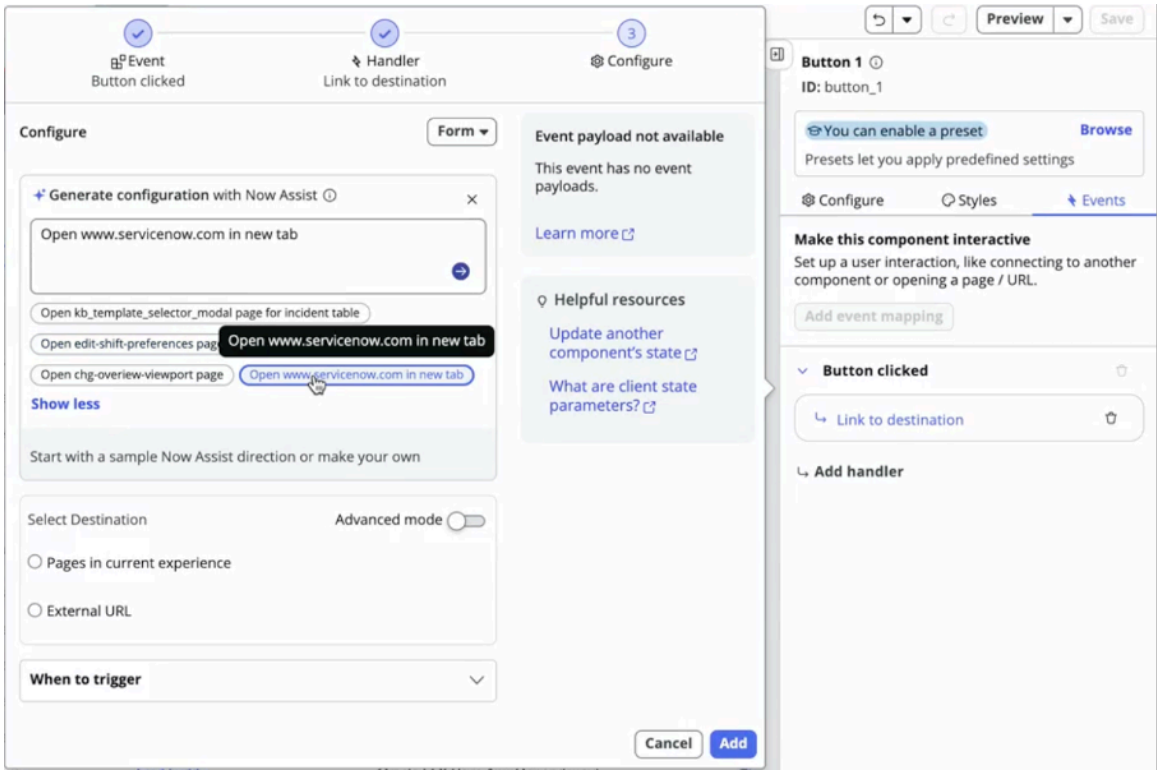
Use ServiceNow Otto in UI Builder to configure event handlers. At present, you can configure Open page or URL, Open or close modal, and View load requested event handlers.

Before you begin
Make sure that you install UI generation and that you have the ui_builder_admin role. For more information, see Install UI generation and Grant UI Builder admin role.

Role required: ui_builder_admin

Procedure
1. In the application navigator, navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   For more information, see Configure how users interact with your applications in UI Builder.
3. Open or create a page.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. Add a component to your page, such as a button.
   For more information about adding components to a page, see Add and configure components.
5. To add an event handler to your component's event, go to the configuration panel and select **Events**.
   An event handler lets you assign an event to a component. For example, if you add a button component to your page, you want it to perform an action when a user selects it.
6. In the event handler, select **Open page or URL** and select **Continue**.
7. In the Configure section, select **Get Started**.
8. In the **Generate configuration with Now Assist** field:
   - Type the action you want to perform (For example, open www.servicenow.com) or select an option from the dynamic prompts.
   - You can reference data from page context.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



9. Select the right arrow icon.
   The configurations are set.
10. You can either reject, or accept and edit the configurations.
11. After you review the changes, select **Add**.
12. Test the event by selecting **Preview** in the header and trigger the action.

Result
The event handler configurations are updated.

Generate modules using ServiceNow Otto

Use ServiceNow Otto to generate modules for your application by entering natural language prompts instead of manually configuring them.

Before you begin
Make sure that you install UI generation and that you have the ui_builder_admin role. For more information, see Install UI generation and Grant UI Builder admin role.

Role required: admin

About this task
Use ServiceNow Otto to generate modules by providing prompts in natural language. The AI agent analyzes your prompt to identify the module type and key parameters, such as table names, views, roles, filters, and arguments. It then generates the requested modules.

Procedure
1. From the application navigator, select **All**, and then select **Create menu**.

   The module creation page opens.
2. In the **Title** field, enter a name for the application menu.
3. **Optional:** In the **Roles** field, enter a specific role to limit access to certain users.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

4. **Optional:** In the **Description** field, provide a description of the application menu to help clarify its purpose.
5. Select **Create**.
   The Add modules to your application menu page opens.
6. Select **Generate multiple modules at once using Now Assist**.
7. In the **Tell Now Assist what you want** field, enter a prompt that describes the modules you want to create.

   For example:

   - Create a list of active incidents assigned to me, and call it 'My Incidents'.
   - Make a Workspace list page for all change requests scheduled this week.
   - Create three record modules for the list, incident, and task tables.

   Tip: You can also select a module type from the available options.

8. Select **Generate**.
   ServiceNow Otto generates the modules.

   An AI indicator appears next to the generated module, showing that it hasn't been confirmed.

9. Review the generated modules and do one of the following:
   - To confirm the module and modify its fields, select **Accept and Edit**.
   - To discard the generated module and try a different prompt, select **Reject**.

   After you select **Accept and Edit**, the AI indicator disappears.
10. **Optional:** After accepting the module, edit the configuration fields as needed.
    You can modify fields such as module type, table, role, and other properties specific to the module type.
11. **Optional:** Add or rearrange modules as needed to suit your specific needs.
12. Select **Done**.

Result

The modules are added to your application menu and are ready for use.

Trouble?

Module generation issues

Troubleshooting Module generation issues

Symptom	Cause	Resolution
Module generation creates the wrong module type	The prompt did not clearly indicate the intended module type.	Specify the module type explicitly in the prompt (for example, "Create a list module" or "Create a URL module"). Alternatively, select a module type from the available options before entering the prompt.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example: Create five modules

1. In the **Tell Now Assist what you want** field, enter the following prompt:

```
Add:
- a single record module called "Test Record",
- an URL with arguments "url=https://example.com" and window
 name "External",
- a list called "My Tasks",
- a script called "Data Cleanup Script" with arguments "var
 gr = new GlideRecord('task'); gr.query();" and window name
 "Cleanup"
- a dashboard module
```

2. Select **Generate**.

   The system creates five modules:

Module types

Module	Type	Description
Test Record	Record	A basic configuration without any additional parameters.
URL	Direct	Contains parsed arguments: `url=https://example.com` and a window name: "External."
My Tasks	List	A basic list configuration.
Data Cleanup Script	Script	Includes parsed script arguments and a window name: "Cleanup."
Dashboard	Dashboard	An empty dashboard module.

3. Select **Accept and edit**.
4. (Optional) If needed, add, or rearrange modules.
5. Select **Done** to finish.

UI Generation use cases

Learn how UI Generation capabilities in ServiceNow Otto for Creator support developers in creating and configuring UI Builder experiences using natural language.

UI generation is a set of AI-assisted features available in ServiceNow Otto for Creator. These features reduce the manual effort involved in building and maintaining ServiceNow UI experiences by enabling developers describe what they need in natural language and having ServiceNow Otto generate ready-to-use output.

Use case: Create an AI-generated experience

Learn how to generate a multi-page UI experience from a natural language description.

Scenario

A government agency is rolling out a new citizen-facing service request portal on ServiceNow. A team of developers with varying levels of UI Builder experience needs to deliver multiple portal experiences quickly, including a dashboard for tracking request status by category, a tabbed experience for managing open cases by priority, and a breadcrumb-style experience for change

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

requests. The team is under a tight deadline and cannot afford to build each experience from scratch.

Problem

Manually creating a multi-page experience in UI Builder involves several steps, including configuring the experience structure, navigation style, chart types, page layouts, and data sources. This process can be particularly challenging for developers who are less familiar with UI Builder, as it presents a steep learning curve. Furthermore, even experienced developers may find the manual creation of multiple experiences to be repetitive and time-consuming, increasing the risk of inconsistencies throughout the portal.

Solution

Experience generation empowers developers to create seamless experiences using natural language directly within the UI Builder. With this feature, developers can articulate their desired experience in a straightforward manner.

ServiceNow Otto automatically generates a fully structured multi-page experience preview, allowing developers to review the output. They can choose to save the generated experience or refine their prompt to regenerate it for further adjustments.

This functionality ensures that all team members, regardless of their level of expertise, can quickly produce a working experience within minutes. Subsequently, any necessary fine-tuning can be performed manually, streamlining the development process and enhancing overall productivity.

Before you begin

Make sure that you install UI generation and that you have the ui_builder_admin role. For more information, see Install UI generation and Grant UI Builder admin role.

Configuration steps

Perform the following steps to use Expereice generation feature in UI Builder:

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. From the UI Builder Home page, select **Create**, then select **AI-Generated experience**.
3. In the **Create an experience** form, enter a detailed description.

   For example:

   ```
   Create a Tabbed experience called Service Request Tracker to
    help track open cases by priority using a bar chart.
   ```

   Tip: The quality of the generated experience largely depends on the clarity of your prompt. Include the following details: the name of the experience, the navigation style (such as Tabbed or Breadcrumb), the type of chart, and the table that users interact with. To see examples of effective prompts, you use the **Try an example** option in the **Create an Experience** form before writing your own.

4. Select **Generate preview**.

   ServiceNow Otto generates and displays a preview of the experience.

5. Review the generated experience. If satisfied, select **Save and edit experience**.

   If not, refine the description and select **Regenerate preview**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

6. After saving, review the main experience page to confirm the URL, app shell, roles, and page variants are correctly configured.
7. Repeat the steps to create additional experience (for example, Change Request Tracking and Request Status Dashboard experiences).
8. Open each saved experience in the page editor to fine-tune layouts, data sources, and styling as needed.

Outcome

The development team creates three experiences in a few minutes, which is significantly faster than the hours it would take to build each one manually. Developers of all experience levels contribute, ensuring that the generated experiences have a consistent structure throughout the portal. As a result, the team can focus most of its time on meaningful customization rather than basic configuration.

Related topics

Experience Generation

Create an AI-generated experience

Use case: Using UI Builder Agent

Learn UI Builder features and accelerate page configuration using conversational AI.

Scenario

A citizen developer at a retail company is creating an employee self-service portal using UI Builder. They are relatively new to the platform and are working on a page that requires a three-column layout, along with a button that links to an external HR system. Additionally, they would like to verify whether any client scripts are running on the page before making any layout changes.

Problem

The developer is not yet familiar with UI Builder 's configuration options. Looking up documentation for each task interrupts their workflow and slows progress. They are unsure how to apply a multi-column layout, add a button with an external link, or check whether existing scripts might be affected by their layout changes.

Solution

The UI Builder agent is available directly within the UI Builder page editor as a chat-based assistant. The developer can ask questions about features, request page modifications, and get configuration insights — all without leaving the editor. The agent provides instant answers backed by the latest ServiceNow documentation and can directly apply changes to the page on the developer's behalf.

Before you begin

Make sure that you install UI generation and that you have the ui_builder_admin role. For more information, see Install UI generation and Grant UI Builder admin role.

Configuration steps

Perform the following steps to use UI Builder agent:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open the employee self-service portal experience.
3. Open the page to edit and open the ServiceNow Otto panel from the page editor.
4. In the chat panel, do the following:
   - Ask the UI Builder agent if there are any client scripts on the page: `Are any client scripts used on my page?`

     The agent provides a detailed breakdown of the page's current scripts and configurations.
   - Ask the UI Builder agent how you can apply a three-column layout: `How can I apply a three-column layout to this page?`

     The agent explains the available layout options and relevant steps.
   - Ask the UI Builder agent to add a three-column layout to the page: `Apply a three-column layout to this page.`

     The agent implements the layout change.
   - Ask the UI Builder agent to add a button with label and navigation: `Add a button with the label 'Visit HR Portal' that links to https://hr.example.com in a new window.`

     The agent adds and configures the button component.
5. Review the changes in the page editor and make any necessary adjustments to the styling.

Outcome

The developer sets up a three-column layout and adds a linked button component in a single session, with guidance from the UI Builder agent. They also gain confidence in their page's script configuration before making any changes. Tasks that would have typically required multiple searches through documentation are now completed without leaving the editor.

Related topics

UI Builder Agent

Using UI Builder agent

Use case: Summarize a client script using ServiceNow Otto

Understand and edit client scripts.

Scenario

A developer at a financial services company has joined a team responsible for managing an existing IT operations portal created with UI Builder. This portal consists of multiple pages that contain client scripts written by a previous developer who is no longer with the organization. Before making any modifications, the new developer needs to understand the functionality of each client script and the specific events that trigger them.

Problem

The client scripts are lengthy and undocumented, making it time-consuming for the developer to read through each one to understand its purpose. Misinterpreting a script before making changes could introduce bugs or break key portal functionality. Manually reviewing and documenting each script would take hours and slow down the team's ability to deliver updates.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Solution

With Client script summarization, the developer can select any client script in the **Data and scripts** panel and request an instant AI-generated explanation. ServiceNow Otto provides both a high-level summary and a detailed breakdown of the script's logic and trigger conditions, enabling the developer to quickly assess the script's purpose and decide whether changes are safe to make.

Before you begin

Make sure that you install UI generation and that you have the ui_builder_admin role. For more information, see Install UI generation and Grant UI Builder admin role.

Configuration steps

Perform the following steps to use the Client script summarization skill:

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open the IT operations portal experience.
3. Select the page that includes the client script for review.
4. In the **Data and scripts** panel, open the relevant client script.

   The **Explain Code** panel appears.

   Note: The **Explain Code** panel appears only if you're assigned the ui_builder_admin role and if the Client script summarization skill is enabled.

5. Select **Explain this code**.
6. Review the concise summary and detailed explanation, including trigger conditions.
7. If you make changes to the script, select **Explain again** to regenerate the summary for the updated version.

   Note: If you update a script after reviewing the explanation, the **Explain Code** panel will notify you of the code changes. Always select **Explain again** to understand the latest version of the script.

Outcome

The developer quickly comprehends the purpose and functionality of the client scripts, whereas manually reading and interpreting the code would take several hours. This knowledge enables the developer to confidently plan and implement updates without the risk of introducing unintended regressions.

Related topics

Client script summarization

Summarize a client script using ServiceNow Otto

Use case: Generate a data binding and formula

Automate data binding configuration without memorizing formula syntax.

Scenario

A developer at a healthcare organization is creating a self-service patient intake portal using UI Builder. The portal features a form that shows dynamic labels based on whether the user is

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

viewing a standard page or a simplified "bare" page layout. The developer needs to configure a label component that changes its text accordingly. Additionally, the developer wants to display the total number of active intake requests submitted that day.

Problem

The developer is not familiar with UI Builder's formula syntax. Writing expressions like IF(@context.props.bare,"bare page","not bare page") or LEN([list]) requires memorizing the exact syntax, which slows down development and increases the risk of errors. Every time the developer encounters a new binding requirement, they have to look up documentation or ask a colleague.

Solution

With Data binding generation, the developer describes what they need in plain language directly in the Formula Builder. ServiceNow Otto converts the natural language input into the correct formula or data binding syntax automatically. The developer reviews the generated syntax, accepts it, and applies it to the component property in one streamlined flow — without writing a single line of formula code manually.

Before you begin

Make sure that you install UI generation and that you have the ui_builder_admin role. For more information, see Install UI generation and Grant UI Builder admin role.

Configuration steps

Perform the following steps to use Data binding generation:

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Navigate to the patient intake experience.
3. Open the page variant in the Page Editor.
4. Select the label component and open the Data binding popover.
5. Select **Get started**.
6. Enter the prompt: `If the page is bare, show 'Simplified View'; otherwise, show 'Standard View'.`

   Tip: Refer to the examples in the Data Binding popover to understand which types of natural language prompts are most effective. Prompts that clearly describe the condition and possible outcomes typically yield the most accurate results.

7. Select the arrow to generate the binding.

   ServiceNow Otto generates: `IF(@context.props.bare,"Simplified View","Standard View")`.

8. Review the syntax, select **Accept and Edit**, then select **Apply**.
9. Enter a prompt to request count label: `Enter how many active intake requests are there today?`

   ServiceNow Otto generates: `LEN([@data.activeIntakeRequests])`.

10. Preview and test the page to confirm both bindings render correctly.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Outcome

The developer sets up a dynamic binding quickly, without referring to the documentation. This eliminates syntax errors, enabling the developer to focus on experience design rather than writing formulas.

Related topics

- Data binding generation
- Generate a data binding and formula
- Natural language synonyms for formula operators

Use case: Configure an event handler with ServiceNow Otto

Automate event handler configuration without writing manual binding expressions.

Scenario

A UI developer is creating a Service Operations Workspace homepage for a service desk team. The homepage features two components: a button that directs agents to a list of open incidents, and a task list that displays open change requests. For the task list, when an agent selects a reference link in any row, the developer wants the associated change request record to open in a new tab, without requiring any additional navigation steps.

Problem

Without ServiceNow Otto, the developer must manually add an Open page or URL handler to the button component and configure both the destination page and the associated table parameter. For the task list, they need to add a second Open page or URL handler to the Reference link clicked event. Additionally, they must write binding expressions to map the event payload fields table and sys_id to the record page parameters. This process requires familiarity with the names of the payload fields, the binding syntax, and the parameter names expected by the record page, making it a complex configuration task for each handler.

Solution

With Event handler generation, the developer uses natural language prompts in the ServiceNow Otto panel to configure an event handler. ServiceNow Otto identifies the correct destination page, selects the appropriate event payload fields, and generates accurate binding expressions. The developer then reviews the preview and accepts the configuration without needing to look up payload field names or manually write any binding expressions.

Before you begin

Make sure that you install UI generation and that you have the ui_builder_admin role. For more information, see Install UI generation and Grant UI Builder admin role.

Configuration steps

Perform the following steps to use the Event handler generation skill:

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open the Service Operations Workspace experience and navigate to the home page.
3. Select the button component and open the Button clicked event.
4. Add an Open page or URL handler. In the ServiceNow Otto input panel, enter the prompt:
   `Open simple list page with table from state parameter.`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

> **Tip:** When writing prompts for Event handler generation, reference the specific destination page type and the fields you want to map by name. For reference link events, ServiceNow Otto displays the available payload fields in the panel — use these field names in your prompt for the most accurate results.

ServiceNow Otto identifies the simple list page, selects it as the destination, and binds the table parameter to the appropriate state parameter.

5. Review the preview, select **Accept and edit**, and save the handler.
6. Select the task list component and open the Reference link clicked event.
7. Add a Open page or URL handler. ServiceNow Otto displays the available event payload fields. Enter the prompt: `Open record page in a new tab with fields table and sysid from event payload`.

   ServiceNow Otto sets the destination to the record page, binds the table parameter to the payload's table field, binds sysid to the payload's sys_id field, and enables the Open in new tab option.
8. Review the preview, confirm the bindings are correct, and accept the configuration.

Outcome

Both event handlers can be configured with natural language prompts. The developer could set them up quickly without looking up payload field names or writing binding expressions manually. During runtime, selecting the button navigates the agent to the incident list, with the table parameter automatically populated from the workspace's state. Additionally, selecting any reference link in the task list opens the corresponding change request record in a new tab, pulling the table name and sys_id directly from the selected row. The generated configurations are ready for immediate use. The developer retains full control and can edit any field using standard form controls before saving.

Related topics

Event handler generation

Configure an event handler with ServiceNow Otto

Use case: Generate modules using ServiceNow Otto

Create and configure application menu modules using natural language

Scenario

An admin at a logistics company is building a new custom application for managing warehouse operations. The application menu should include multiple modules:

- A list of active tasks
- A form for logging new incidents
- A dashboard for warehouse KPIs
- A link to an external supplier portal
- A folder for organizing reporting modules

Configuring each module individually through the standard UI would require navigating multiple forms and configuration screens.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Problem

Creating and configuring each module in the UI Builder manually can be time-consuming and repetitive, particularly when multiple modules are required simultaneously. Each module type includes its own set of configuration fields, including table names, filters, roles, views, and window settings. As a result, the administrator must fill in these fields individually. When dealing with an application with five or more modules, this process can take considerable time and is prone to configuration errors.

Solution

Using the Module generation feature, the admin provides a single natural-language prompt that describes all required modules. The AI agent analyzes the prompt, identifies the appropriate module types and key parameters, and generates all modules in a single step. The admin then reviews each generated module, makes any necessary adjustments, and saves the complete application menu.

Before you begin

Make sure that you install UI generation and that you have the ui_builder_admin role. For more information, see Install UI generation and Grant UI Builder admin role.

Configuration steps

Perform the following steps to generate modules:

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. From the application navigator, select **All**, then select **Create menu**.
3. Enter a name for the application menu (for example, Warehouse Operations) and select Create.
4. On the Add modules page, select **Generate multiple modules at once using** *Otto*.
5. Enter the following prompt: `Create a list of active warehouse tasks, a new record form for logging incidents, a KPI dashboard called Warehouse Overview, a URL module linking to https://supplier-portal.example.com with window name Supplier Portal, and a folder called Reports.`
6. Select **Generate**.

   ServiceNow Otto creates five modules and displays an AI indicator next to each.
7. Review each module. Select **Accept and Edit** for each module to confirm and adjust configuration fields such as table name, role restrictions, or view settings.
8. Rearrange modules if needed, then select Done.

Outcome

The administrator creates and configures a five-module application menu within a single workflow. The resulting modules are immediately ready for use in the application navigator.

Related topics

Module generation

Generate modules using ServiceNow Otto

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

UI generation reference

Reference topics provide additional information about configuration properties, roles, and more.

UI generation roles

The following roles are installed for use with UI generation.

To learn more about managing per-user subscriptions, see Managing per-user subscriptions in Subscription Management and contact your account representative.

ServiceNow Otto for Creator [ui_builder_admin]

Create experiences using AI-powered UI generation.

Groups

This role is assigned to no groups by default.

Contains Roles

This role contains ui_interaction_admin, canvas_user, and workspace_admin.

Elevated

This role is not an elevated role.

Special considerations

None

Roles and permissions matrix

Learn about the roles required to access each UI Generation feature.

UI Generation roles and permissions

The ui_builder_admin role is the primary role for using ServiceNow Otto UI Generation skills.

UI Generation roles and permissions

Feature	ui_builder_admin	workspace_admin	admin
Experience Generation (AI-Generated)	#	–	–
Client Script Summarization	#	–	–
Data Binding Generation	#	–	–
UI Builder Agent (Learn / Understand / Accelerate)	#	–	–
Module Generation	#	#	–
View AI-Generated Indicator Badge	#	#	#
Manage UI Generation System Properties	–	–	#
Assign ui_builder_admin Role	–	–	#

Role descriptions

ui_builder_admin

Primary role for UI Generation. Contains ui_interaction_admin, canvas_user, and workspace_admin as sub-roles.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For more information, see UI generation roles and Grant UI Builder admin role.

workspace_admin

Allows workspace-level configuration and management within UI Builder.

admin

Full instance administration, including system properties management and role assignment.

Natural language synonyms for formula operators

Data Binding Generation enables ServiceNow Otto to recognize property names by using a set of frequently used synonyms.

Formula operators

With Data Binding Generation, ServiceNow Otto can recognize property names through a library of common synonyms. This feature reduces failed field matching and eliminates the need for builders to understand the exact syntax. For instance, the LEN formula operator is associated with various common terms and phrases such as count, how many, length, number of items, and size. This enhancement improves accuracy and speeds up the binding generation process.

Here's a list of additional examples of operators and their synonyms to guide effective prompt writing:

Formula Operators and Natural Language Synonyms

Formula Operator	Category	Common Synonyms
`LEN`	Math / String	count, how many, length, number of items, size
`SUM`	Math	addition, add up, calculate, total
`MIN`	Math	least, lowest, minimum, smallest
`MAX`	Math	greatest, highest, largest, maximum
`AVG`	Math	average, mean, middle value
`ROUND`	Math	round off, round to, to decimal places
`ABS`	Math	absolute, magnitude, positive value
`CONCAT`	String	append, combine, join, merge strings
`TRIM`	String	clean up, remove spaces, strip whitespace
`UPPER`	String	capitalize all, make uppercase, to caps
`LOWER`	String	all lowercase, make lowercase, to small letters
`CONTAINS`	String	has, includes, is part of
`STARTSWITH`	String	begins with, leads with, starts
`ENDSWITH`	String	closes with, ends in, finishes with
`IF`	Logic	case, check, condition, decision, when
`AND`	Logic	all of, both, every condition, together with
`OR`	Logic	any of, at least one, either
`NOT`	Logic	does not, exclude, negate, opposite of
`ISBLANK`	Logic	empty, is empty, missing, no value

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Formula Operators and Natural Language Synonyms (continued)

Formula Operator	Category	Common Synonyms
`NOW`	Date / Time	current datetime, right now, today's time
`TODAY`	Date / Time	current date, present day, this date
`DATEDIFF`	Date / Time	days between, elapsed time, time gap
`DATEADD`	Date / Time	add days, days from now, future date
`PARSE`	Data	break down, decode, interpret, read format
`LOOKUP`	Data	find value, get field, reference table

Note: There are more than 70 operators, each with multiple synonyms mapped together. This table lists the most commonly used operators. For a complete list, open Formula Builder within UI Builder and select **Browse operators**.

Property labels

With Data Binding Generation, ServiceNow Otto is able to recognize properties based on keywords found in the data property label. If a prompt uses a synonym that doesn't match an exact property (such as, surname), ServiceNow Otto finds the closest related keyword, such as "name". When a prompt includes either an exact property label or part of the label, such as a first name or family name, it matches directly and improves accuracy.

Using agentic AI with ServiceNow Otto for Creator

Use the AI agents and agentic workflows available with ServiceNow Otto for Creator to help you build on the ServiceNow AI Platform more efficiently.

ServiceNow Otto for Creator contains agentic AI assets, such as AI agents and agentic workflows, that empower you to build more efficiently on the ServiceNow AI Platform. The agentic AI assets available with ServiceNow Otto for Creator are embedded in applications across the ServiceNow AI Platform, enabling you to leverage agentic AI at every stage of your development process. For example, with ServiceNow Otto for Creator, you can develop entire full-stuck applications in conversation with Build Agent, or troubleshoot ATF test failures with the ATF troubleshooting agent. The following table highlights the agentic AI assets available with ServiceNow Otto for Creator and the applications that the AI assets work in.

ServiceNow Otto for Creator skills and the applications that they work in

Agentic AI asset	Application
ATF troubleshooting agent	Automated Test Framework (ATF)
Build Agent	• ServiceNow IDE • ServiceNow Studio
Release lifecycle documentation AI agent	• Core UI • ReleaseOps • App Engine Management Center (AEMC) for regenerating release notes

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ServiceNow Otto for Creator skills and the applications that they work in (continued)

Agentic AI asset	Application
Theme generation agentic workflow	Theme Builder
UI Builder agent	UI Builder

ATF troubleshooting agent

Use ATF troubleshooting agent to simplify test failure diagnosis on specific metadata.

Get started

Explore  Learn about ATF troubleshooting agent.	Configure  Install ATF troubleshooting agent to utilize its features.
Use  Use ATF troubleshooting agent to diagnose test failure.	Reference  Learn miscellaneous information about ATF troubleshooting agent.

Exploring ATF troubleshooting agent

ATF troubleshooting agent streamlines the process of figuring out why tests are failing on specific metadata.

ATF troubleshooting agent overview

Use ATF troubleshooting agent to simplify test failure diagnosis on specific metadata.

ATF troubleshooting agent users

AI agents use role masking to determine which users can access them. Ones installed with your applications have specific roles that come included with the application. If you select *`Users with specific roles`* for user access, you must configure the security controls to include these roles. For the instructions to change the security controls, see Define security controls for an AI agent.

ATF troubleshooting agent has the following users.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Users	Description
• system_admin • atf_test_admin • atf_test_designer • atf_ws_designer	Either of these 4 roles is required to access the ATF module and test results tables.
atf_triager + now_assist_panel_user	This role is required to access metadata tables to accurately troubleshoot test failure reasons. This role allows you to have the following access: • Read-only access to test/step result tables • Read-only access to any sys_metadata • Read-only access to syslog • Read/write access to sn_atf_assist_triage_output **Note:** You can access the ServiceNow Otto panel using the now_assist_panel_user role.

ATF troubleshooting agent benefits

Benefit	Feature	Users
Simplify test failure diagnosis	Troubleshoot test failures	atf_triager + now_assist_panel_user

What to explore next

To learn more about configuring and using ATF troubleshooting agent, see:

- Install ATF troubleshooting agent
- Use

Install ATF troubleshooting agent

Install the ServiceNow Otto for Creator application from the ServiceNow Store to get the ATF troubleshooting agent application.

Before you begin

- Review the ServiceNow Otto for Creator application listing in the ServiceNow Store for information on dependencies, licensing or subscription requirements, and release compatibility.
- The default model provider for ATF troubleshooting agent is Anthropic Claude on AWS.

Ensure that the following applications are installed:

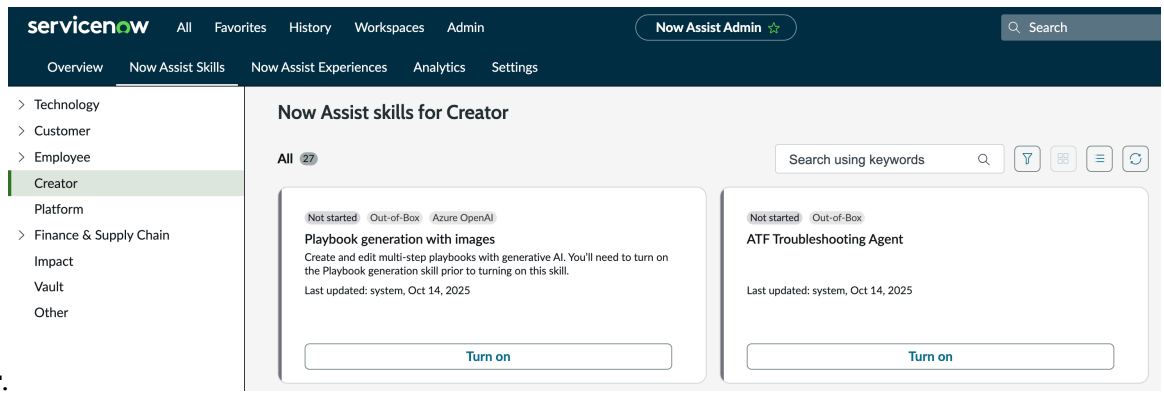
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Build Agent: See Install Build Agent for more information.
- ATF Test Generator and Cloud Runner: Install the ATF Test Generator and Cloud Runner store application from ServiceNow store. You also need to set up the cloud user for seamless execution from the Build Agent interface. See ATF Test Generator and Cloud Runner and Set up cloud user for more information.

Role required: admin

Procedure

1. From the ServiceNow Otto for Creator application page on the ServiceNow Store, select **Buy**.
2. After approval has been granted, on your instance, navigate to **All** > **System Applications** > **All Available Applications** > **All**.
3. Using the search bar, search for the ServiceNow Otto for Creator application (sn_now_creator).
4. Select **Install**.
5. Enable the ATF troubleshooting agent skill:
   a. Navigate to **All** > **AI Admin Hub**.
   b. Go to the **AI Skills** tab and select **Creator**.



   c. Select **Turn on** to enable the skill.
   The skill is enabled for all users.

Troubleshoot test failures

Use ATF troubleshooting agent to significantly reduce the skills and resources needed to troubleshoot test failures on covered metadata.

Before you begin

Note: You must have either of these roles to access the ATF module and test results tables.

- system_admin
- atf_test_admin
- atf_test_designer
- atf_ws_designer

Role required: atf_triager + now_assist_panel_user

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **Automated Test Framework (ATF)** > **Tests**.
2. Select the test you want to run.

   Note: You can also select **New** if you want to create a new test.

3. **Optional:** On the Test new record form, enter all the details to create the test.

   Note: This step is applicable only if you have created a new test in the previous step.

4. **Optional:** Add new test steps to the existing or newly created test.
5. Select **Run Test**.
   The Run Test modal shows up.
6. Select **Go to Result** to view the information on the test execution.
   If you instead select the Test Results related list after the test execution is completed, a list of step summary shows up. You are required to select a step summary record with a Failure status to know more about the reasons of its failure.
   The Test Results form shows up.

   Note: You can access the Test Results form by selecting **Go to Result** on the Run Test modal or from the Test Results related list.

7. Select **Triage with Otto** on the Test Results form.
   The ServiceNow IDE or ServiceNow Studio interface shows up.
8. Review the prompt that is pre-populated depending on the test you're running and the selected summary record with a Failure status.
9. Follow the automatic triaging process for the failed record.
   When a test fails, the agent automatically starts the troubleshooting process. It identifies the step where the failure occurred and analyzes the root cause. The agent then recommends possible fixes. After you select a recommended option, the agent applies the fix, updates the test code, and reruns the test automatically. When the test passes, it provides a summary of the changes made to resolve the failure.

   Note: See Author, execute, and troubleshoot tests and test suites with Test Agent if you want to execute a new application or a test.

ATF troubleshooting agent references

The reference topics provide additional information about ATF troubleshooting agent.

ATF troubleshooting agent design considerations

Leverage the full potential of ATF troubleshooting agent by following these design considerations.

Supported metadata

ATF troubleshooting agent supports the following metadata:

- sys_ui_action
- sys_ui_policy
- sys_ui_policy_action
- sys_ui_policy_rl_action

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- sys_script_client
- sys_ui_script
- sys_data_policy2
- sys_ui_action_role
- sys_ui_action_view
- sys_ui_base_theme
- sys_ui_form
- sys_ui_formatter
- sys_ui_macro
- sys_ui_message
- sys_ui_related_list
- sys_ui_section
- sys_ui_view
- sys_ui_title
- sys_ui_style
- sys_ui_theme
- sys_ui_page
- sys_ui_context_menu
- sys_ui_list
- sys_ui_list_control
- sys_ui_list_control_embedded
- sys_ui_list_script_client
- sys_ui_list_script_server
- sys_ui_related_list
- sys_user_role
- sys_app_application
- sp_widget
- sp_instance
- sys_security_acl_role
- sys_script
- sys_app_module
- sysevent_email_action
- sys_report
- sys_security_acl
- catalog_ui_policy
- catalog_ui_policy_action
- sp_theme
- sp_widget
- sys_sg_universal_link_path_segment

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- sys_sg_sections_screen_m2m_section
- sys_sg_location_tracking_details
- sys_highlighted_value_condition
- sys_sg_card_size
- sys_sg_custom_map_config
- sys_sg_empty_state
- sys_aw_form_header_secondary_values
- sys_aw_registered_scripting_modal
- sys_template
- sys_sg_alert
- sys_aw_form_header
- sys_script_include
- sys_scriplet_function
- sys_ui_extension_instance
- sys_playbook_experience_action_assignment_map
- sys_df_data_filtration
- sys_email_address_filt_domain
- sys_atf_step
- sys_atf_test
- sys_atf_step_config
- atf_input_variable

Exceptions

ATF troubleshooting agent doesn't support the following metadata:

- Flows
- Configurable workspace steps
- Custom UI test steps
- Custom test steps

ServiceNow Studio

Use ServiceNow Studio to build, manage, and deploy custom, base system, or global applications and global metadata records from a single development environment.

What is ServiceNow Studio?

https://player.vimeo.com/video/1143236599?h=613ebecdba&badge=0&autopause=0&player_id=0&app_id=58479

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Get started

Explore



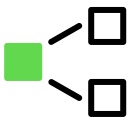
Learn about ServiceNow Studio concepts and features.

Configure



Configure environments, tools, and user access.

Build



Build applications, flows, and more using ServiceNow Studio.

Reference



Get details about ServiceNow Studio components such as fields, tables, and properties.

Which app builder should I use?

Use the following table to choose the app building experience that matches your role and technical background.

App builders

Tool	Users	Features
Build Agent: Build an app using natural language prompts.	Developers of all experience levels (from low-code to Platform and source code)	Build full-stack applications via a conversational interface. You can use Build Agent in ServiceNow Studio and the ServiceNow IDE to generate a complete ServiceNow scoped or global application. For more information, see Exploring Build Agent.
ServiceNow Studio: Build and deliver apps in a unified environment.	Citizen developers, Platform developers	Build apps in a unified development environment. ServiceNow Studio provides streamlined navigation, integrated low-code tools, and built-in tracking and packaging so you can develop and ship apps faster. Use Build Agent in ServiceNow

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

App builders (continued)

Tool	Users	Features
		Studio to create and update apps with a conversational interaction. For more information, see Exploring ServiceNow Studio.

Give ServiceNow Studio a try

Ready to give ServiceNow Studio a try? You can test it out using your own Personal Development Instance (PDI), which requires you signing in to the Developer Site. Find out more on PDIs in the Personal developer instance guide.

Select this button to try ServiceNow Studio on a PDI

Try ServiceNow Studio for free

Try installing ServiceNow Studio now on a PDI! Login required.

Helpful resources

Use these ServiceNow resources to find helpful information about app development.

ServiceNow Community

The fastest way to learn Build Agent prompting? Ask Build Agent.

Video

Watch App Engine Academy: Platform capabilities and data modeling for custom applications.

Video

Watch App Engine Academy: Using ServiceNow Studio for Multi-Level Table Inheritance.

ServiceNow Community

Find Instance help for your personal developer instance (PDI)

Video

Watch App Engine Academy #1: Custom App Development Best Practices

Customer Admin Support

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Contact your company's Customer Admin to unlock or add user accounts, perform restores or zBoots, and more.

Known Error Portal

Search the Known Error Portal for known error articles

Customer Service

Contact Customer Service and Support

Exploring ServiceNow Studio

ServiceNow Studio is a unified development environment where platform developers create, modify, and deploy apps. Use ServiceNow Studio to manage app files, edit metadata, and package changes for deployment without switching between tools.

What can you do in ServiceNow Studio?

ServiceNow Studio provides a single environment for app development, from creation to deployment. Use it to:

- Build, manage, and deploy custom, base system, or global applications and global metadata records, all without switching between tools. You can also customize store apps.
- Search for and edit any type of metadata record using the Navigator panel taxonomy.
- Package app changes for deployment using update sets, pipelines, or the Application Repository.

ServiceNow Studio features and benefits

Use ServiceNow Studio to build, manage, and deploy apps faster in a single development environment. It provides the following features for admins and delegated developers.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



- AI-native design: Use the generative and agentic AI capabilities in ServiceNow Studio to build apps faster and extend custom applications with AI. For more information, see AI tools and files in ServiceNow Studio.
- Build Agent: Use Build Agent, an autonomous AI agent, to create and update applications in ServiceNow Studio. For more information, see Build Agent in ServiceNow Studio.
- Integrated builders: Access low-code builders available in the ServiceNow AI Platform, including Table Builder and flows in Workflow Studio, alongside app development tools. For more information, see Integrated development tools for ServiceNow Studio.
- AI and code search: Search for any application or metadata record in your instance using AI Search and standard code search. For more information, see Find an app or app file using Search and Find an app or app file using the Navigator panel.
- Accelerated deployment: Create and package apps for deployment using update sets, pipelines, or the Application Repository without leaving ServiceNow Studio. For more information, see Update sets in ServiceNow Studio and Application Repository in ServiceNow Studio.
- Choose your development experience: Switch between development environments using the experience switcher in ServiceNow Studio to use the best tool for each stage of your project. For more information, see Change your development experience in ServiceNow Studio.
- Create apps and files: Create apps and files in any scope that you have access to edit, including the global scope. For more information, see Create an application in ServiceNow Studio and Create an app file in ServiceNow Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Global and custom scope: Open and edit files of any scope you have access to, including global and custom files. The scope switches automatically when you switch between open files in different applications. For more information, see Open apps and app files across scopes in ServiceNow Studio.

Helpful resources

Use these ServiceNow resources to learn more about ServiceNow Studio and app development:

ServiceNow Community

The fastest way to learn Build Agent prompting? Ask Build Agent.

Video

Watch App Engine Academy: Platform capabilities and data modeling for custom applications.

Video

Watch App Engine Academy: Using ServiceNow Studio for Multi-Level Table Inheritance.

Video

Watch App Engine Academy #1: Custom App Development Best Practices

ServiceNow Community

Learn Instance help for your personal developer instance (PDI)

ServiceNow Social Ambassadors Program

ServiceNow Social Ambassadors Program training

ServiceNow Studio and Build Agent tutorial

Use Build Agent in ServiceNow Studio to develop reusable server-side logic and create an application on the ServiceNow AI Platform, taking it from initial data modeling through testing and validation.

Complete this tutorial to learn how to use Build Agent for common development tasks and to apply practical patterns for data modeling, testing, and validation. This tutorial covers the following items:

- Application creation, custom data modeling, and table design
- Server-side business logic with script includes
- Validation and testing using background scripts
- Reusable utility patterns for real-world use cases

The full ServiceNow Studio and Build Agent tutorial is hosted on Gitbook, and may require a login.

Use App Engine instead of customizations

Use App Engine development tools to extend the ServiceNow AI Platform without customizing existing applications. Building a new app preserves upgrade compatibility and keeps your changes isolated and easier to maintain.

When your company needs to add new functionality to the ServiceNow AI Platform, you can customize existing applications, such as IT Service Management (ITSM), or create a new

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

application using App Engine developer products, such as App Engine Studio, Creator Studio, or ServiceNow Studio. A simple guideline for which path to choose is:

- If the customization extends the intended purpose of the application, it works better to customize. For example, you can add IT functionality to ITSM.
- If the customization doesn't extend the intended purpose of the application, it works better to create a new application using App Engine developer products. For example, do not repurpose the ITSM workflow to add a travel request workflow.

Examples of when to use App Engine

ServiceNow products work best when they're used as they were intended. If you find yourself heavily customizing an application to repurpose it, a better plan is to create a new application using App Engine developer products.

The following scenarios demonstrate where creating a new application works better than heavily customizing an existing ServiceNow application:

- Your company has a business process that augments existing product functionality but doesn't follow the same workflow exactly.
- You have a novel use case for an app that doesn't align with any product workflow.
- You have a use case that could be built by heavily customizing an out-of-the-box application, but it doesn't align with what the existing application was intended to do.

Let's dive deeper into the last use case.

Issues with repurposing existing products

ServiceNow applications come with roles, processes, and flows that are specially tailored to their use case. For example, ITSM apps help with IT users, IT issues, IT reports, and IT cases.

You might have an idea for an app that's similar to but doesn't exactly align with ITSM. Because ITSM gives you a starting point, you might be tempted to customize ITSM to add the new functionality. For example, ITSM tracks IT issues, and a travel app you want to create might track travel requests. While the workflows sound similar, in fact, they use very different data, different user interfaces, and the details of each workflow vary greatly. Rather than heavily customize ITSM to repurpose it, a better plan is to use App Engine developer products for the following reasons:

- Combining two workflows creates conflicts.
- Customizing applications has implications.

Combining two workflows creates conflicts

In the ITSM example, the repurposing of ITSM to include a travel workflow uses different data, different tables, different roles, and different workflows than ITSM. As ITSM, ITSM customizations, and the travel workflow grow over time:

- Their features will continue to diverge.
- Adding new functionality or fixing problems in one workflow might adversely impact the other.
- The performance of ITSM may suffer.
- The code base will grow and the two purposes of ITSM will make troubleshooting more difficult.
- Quality engineers will require two different testing frameworks.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

All these issues may cause unnecessary complications, poorer performance, upgrade delays, and software problems.

Customizing applications has implications

The ServiceNow AI Platform is built to embrace customization and configuration. The ServiceNow AI Platform is flexible enough to fit your company's business needs. How you customize ServiceNow applications, however, can have significant impacts on ServiceNow support, upgrading to future ServiceNow AI Platform versions, and the functionality of the platform.

Let's start by differentiating customization and configuration:

- Customization is any change made to the code that is part of the baseline installation of a ServiceNow instance. You use code to customize applications.
- Configuration is any change you make to the behavior of a product that does not touch the code in the baseline installation of a ServiceNow instance. You can use system properties, ServiceNow products, or code to configure an application.

The following are some of the implications that result from customizing applications:

- If you add code to an application, you own it whether or not it modifies the code in the baseline installation on a ServiceNow instance.
- The platform marks all customizations and skips them when you update to a new version of the platform. That means you are responsible for manually updating the customizations. This can have a significant impact on the time and resources required to update to new platform versions.
- The ServiceNow AI Platform uses a framework that supports applications in how they process tasks, how forms are rendered in multiple browsers, and the overall user experience. Introducing customizations can have unintended consequences on this framework.
- You own the burden of testing custom code and determining if it impacts platform functionality.
- ServiceNow Customer Support cannot troubleshoot custom code or issues caused by custom code.

Customization is one of the key features of the ServiceNow AI Platform. However, over-customizing an application to repurpose it is likely to generate technical debt, lengthen your upgrade cycle, and complicate future platform upgrades because the custom code may not easily migrate to new platform versions.

Conclusion

Customization and configuration are hallmarks of the ServiceNow AI Platform that enable your company to customize workflows to fit its specific needs. Proceed with these tasks in the following order:

1. Configure ServiceNow applications as much as you can before customizing them.
2. Customize an application only when it extends the intent of the application.
3. Use App Engine developer products, such as App Engine Studio, Creator Studio, and ServiceNow Studio, to create applications rather than customizing an application to create functionality that doesn't align with its original purpose.

For more information, see Customization vs configuration with ServiceNow Studio.

Customization vs configuration with ServiceNow Studio

There are important differences between customizing and configuring ServiceNow applications. The ServiceNow platform is built to embrace customization and configuration, but how you do so

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

can have significant impacts on ServiceNow support, upgrading to future ServiceNow platform versions, and the functionality of the ServiceNow platform.

The general rules around customization are:

- Customize an application only if it extends the original intent of the application. For example, add IT functionality to ITSM but don't add a travel workflow. Instead of over-customizing an application, create a new application using App Engine products, such as Creator Studio or ServiceNow Studio.
- Configure as much as you can before customizing an application.
- If you add code or make other modifications to out-of-the-box functionality, you own them.

What is configuration

Configuration is the process of using ServiceNow built-in tools and features to modify an application's behavior without making changes to flows or the code that is part of the baseline installation on a ServiceNow instance.

Configuration can take the form of using ServiceNow built-in tools to add tables and more, setting instance-wide parameters, as well as using code to extend an application's functionality to meet business needs as long as the code does not modify the baseline code installation. The entire platform is designed for you to add configuration code.

If you add code, such as workflow scripts, you own it, even if it doesn't alter the baseline code installation. That includes owning the impact it has on the entire ServiceNow platform. Issues that arise from added code are beyond the scope of ServiceNow support to debug.

Reverting a configuration should not require any change to the baseline code.

Configuration examples include:

- Forms: Configure tables, fields, data types, default values, and field dependencies to configure the data you capture and display.
- UI elements: Modify layouts, add related lists, add buttons, and change field names.
- Service Catalog: Configure portals where your customers can request catalog items such as service and product offerings.
- ACLs: Restrict unauthorized users from accessing forms and data.
- System property values: Modify the application's experience for all users.

What is customization

Customization is any change made to the flows or the code that is part of the baseline installation on a ServiceNow instance. You use ServiceNow products or code to customize applications.

If you add code, you own it, even if it doesn't alter the baseline installation. That includes owning the impact it has on the entire ServiceNow platform.

Customization examples include:

- Scripting: Customize ServiceNow through scripting using JavaScript. This includes creating client scripts, server-side scripts, and business rules with intricate logic that modify the baseline code.
- Custom tables: Develop custom tables to accommodate specialized data that doesn't fit within standard tables.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Integration: Customize integration with external systems, such as APIs and web services, for seamless data exchange.
- Widgets and portals: Create custom widgets and portals to provide unique features and user experiences.
- Workflows: Create and modify workflows using Workflow Studio. Create and manage playbooks, flows, actions, decision tables, and integrations from one design environment to automate tasks. Upgrading to a new version of a flow requires reapplying your customizations.

Tools for customization and configuration

ServiceNow offers many tools and features, such as business rules, to modify the out-of-the-box behavior of ServiceNow applications. Whether they customize or configure an application depends on how they're used. Using these tools to modify the installed code base constitutes customization. Using these tools to add code that does not modify the flows, or the installed code base constitutes configuration. In both cases, you own the code you add as well as the impact it has on the ServiceNow platform.

ServiceNow tools include:

- UI Policies: Dynamically modify the visibility of fields and attributes on a form according to user inputs.
- Business rules: Automatically trigger actions based on specified conditions.
- UI Actions: Extend and customize forms and lists by adding buttons, context menu items, or other UI elements that perform specific actions when clicked.
- Client-side scripts: Scripts that execute within the user's browser when certain actions occur on a form or a UI page.
- Server-side scripts: Scripts that execute on the ServiceNow server or database, for example, to update record fields when a database query runs.

What is personalization

Personalization is when users use out-of-the-box application tools to modify an application's look and feel only for themselves. Admins can change the look and feel for all users and that is considered configuration. Personalization examples include a user choosing to use dark mode or choosing which table columns to display.

Personalization does not change the baseline code installation on a ServiceNow instance. So, personalization does not impact customer support or interfere with upgrades to new ServiceNow versions.

Ramifications of customizing ServiceNow products

The ServiceNow platform is extremely flexible and built to embrace customization and configuration to fulfill a wide range of business requirements. How you customize ServiceNow applications, however, can have significant impacts on ServiceNow support, upgrading to future ServiceNow platform versions, and the functionality of the platform. Instead of customizing ServiceNow applications, consider using App Engine development products, such as Creator Studio and ServiceNow Studio to create new applications.

The ServiceNow platform uses a framework that supports applications in how they process tasks, how forms are rendered in multiple browsers, and the overall user experience. ServiceNow relies on the framework's integrity to develop and provide support in a consistent manner. Customizations can impair this framework, change platform functionality, and impair workflows and upgradeability.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Customizations trigger the platform to create sys_update_xml records, which are stored in the Customer Update table. The platform marks all customizations and skips the customized records when you update to a new version of the ServiceNow platform. That means you are responsible for manually updating the customizations. This can have a significant impact on the time and resources required to update to new platform versions.

Note: The Customer Update table also contains modifications or additions to configuration metadata, for example, creating a new catalog item or a new workflow.

For more information, see the Customer Updates table. Note that the complexity of maintaining customizations dramatically increases as the number of customizations increases.

Customizing the installed code base can be costly, generate technical debt, lengthen your upgrade cycle, and complicate future platform upgrades because the custom code may not easily migrate to new platform versions. Custom code can change the standard functionality of the ServiceNow platform in unintended ways. Evaluate demands for customization carefully and only resort to customization where there is clear business value and no alternative. Wherever possible, avoid customization by using configuration instead.

If you customize a product:

- You are responsible for maintaining the customization going forward.
- Customer Service and Support will not support problems caused by custom code. If it is the cause of the problems, the support team will likely advise you to revert to the out-of-the-box code.

What is the Customer Service and Support stance on supporting customization

The ServiceNow Customer Service and Support stance on customization is if you add code, you own it and its consequences. Why? Customer Support is not privy to your custom business logic, doesn't know what the expected behavior should be, is unable to reproduce the issue on an out-of-the-box instance, and customer support engineers are not certified implementation specialists, so they are not certified to review customized code logic.

Alternatives to customization

If you have requirements and ideas for enhancements, instead of customizing the installed code base, you can:

- Use configuration instead of customization.
- Submit an Enhancement Request to the ServiceNow development team. Each request is evaluated and, if approved, will be incorporated into a future release.
- Create an app using App Engine developer products to handle desired functionality.

When to use App Engine developer products instead of customizing

When your company needs to add new functionality to the ServiceNow platform, you can customize existing applications, such as ITSM, or create a new application using App Engine developer products, such as Creator Studio or ServiceNow Studio. A simple guideline for which path to choose is:

- If the customization extends the intended purpose of the application, it works better to customize. For example, you can add IT functionality to ITSM.
- If the customization doesn't extend the intended purpose of the application, it works better to create a new application using App Engine developer products. For example, do not repurpose the ITSM workflow to create a travel request workflow.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For example, ITSM is designed to handle IT issues. To customize it to handle travel requests goes beyond the original intention of ITSM. Because IT and travel requests have different workflows, it's better to create a travel request app using App Engine developer tools, such as Creator Studio and ServiceNow Studio, instead of customizing ITSM.

For more information, see Use App Engine instead of customizations.

Examples of when to use App Engine developer products

ServiceNow products work best when they're used as they were intended. If you find yourself heavily customizing an application to repurpose it, a better plan is to create a new application using App Engine developer products.

The following scenarios demonstrate where creating a new application works better than heavily customizing an existing ServiceNow application:

- You have a novel use case for an app that doesn't align with any product workflow.
- You have a use case that could be built by heavily customizing an out-of-the-box application, but it doesn't align with what the existing application was intended to do.
- Your company has a user group or business process that should be separate from the OOTB product workflow.

Guidelines for customizing ServiceNow products

If you must make a customization, consider the following suggestions:

- Maximize configuration options first.
- Avoid copying objects. Instead, update objects in place wherever possible, except for Service Portal widgets and other items designated to be reused.
- Default to "add before edit." This means that you should, for example, add fields to forms rather than change the type of an existing field. Avoid using the same names as out-of-the-box objects, methods, or classes when adding them.
- Minimize the number of fields you add to a form. The more fields you have on a form, the longer it may take to load.
- Export original records as backups before customizing them. Keep track of their out-of-the-box sys_id's in case you must restore them in the future.
- Use scoped applications as your default for any new custom development.
- Document all customizations. Add comments explaining why you customized (including business justification). Review all comments before upgrading to determine if you can revert to out-of-the-box code.
- Create tests for all customizations. Write Automated Test Framework (ATF) tests for all customizations where possible.
- Use HealthScan regularly to identify unnecessary customizations.
- Customizations should be made to baseline objects where necessary so that conflict resolution and decision-making can be appropriately recorded in the updates. Hidden customizations may cause administrators to overlook updates in future assessments in case reverts or merges are necessary.
- Test your customization for all use cases. Include performance testing and the introduction of unintended consequences.
- Administrators are responsible for verifying that their customizations work after a ServiceNow platform upgrade, and for keeping track of what customizations are made.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Handling customizations when you upgrade

Customizations trigger the platform to create sys_update_xml records, which are stored in the Customer Updates table. These records are not updated during platform version upgrades. ServiceNow marks them as skipped records in the ServiceNow Upgrade Monitor. To make sure they're successfully ported to the upgraded instance, you must manually process the skipped changes. For more information, see the Customer Updates table.

Assuming you've documented all your customizations—including the business justification—take your documented inventory and compare it with the skipped records identified in the Upgrade Monitor. After filtering out low-risk changes that have resulted in skipped records (for example, field labels or form layouts), you'll need to decide whether to:

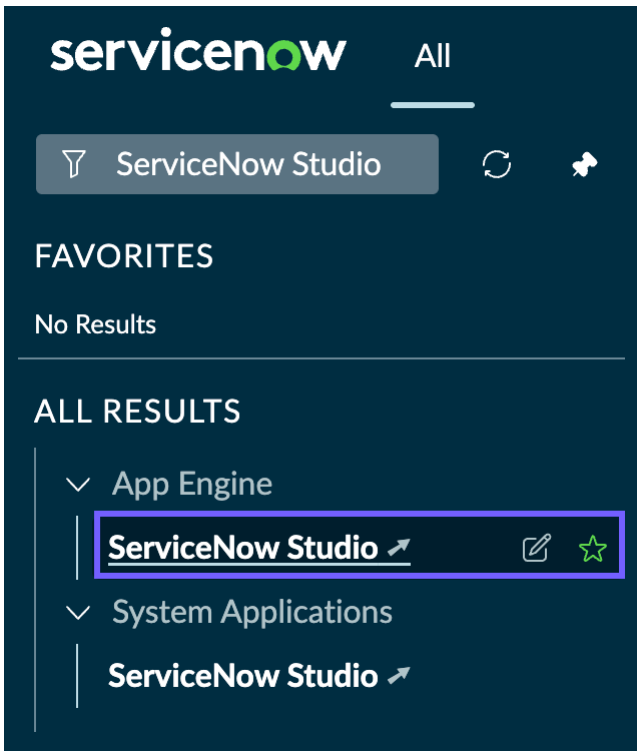
- Retain each customization
- Revert to out-of-the-box
- Merge your customization with the base system to resolve conflicts

Access ServiceNow Studio

Open ServiceNow Studio from the ServiceNow AI Platform All menu or directly from App Engine Studio.

How do I open ServiceNow Studio from the All menu?

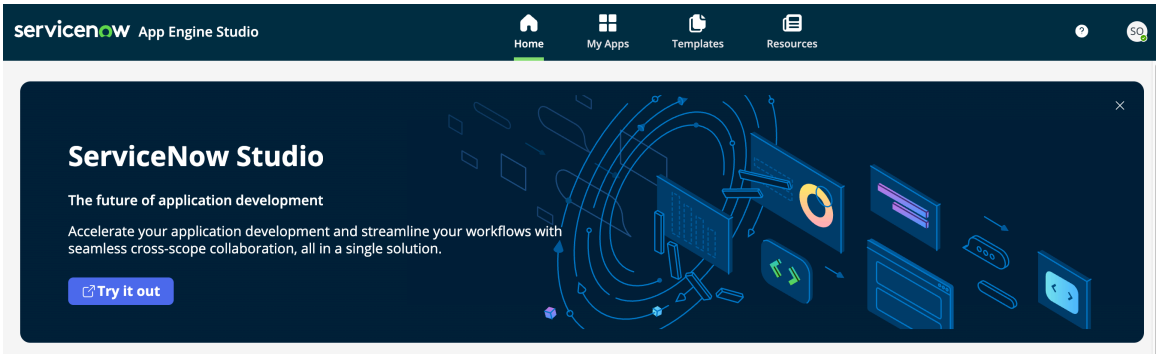
Navigate to **All** > **App Engine** > **ServiceNow Studio**, or search for `ServiceNow Studio` in the filter navigator and open it from the search results. To access ServiceNow Studio faster, add it to your **Favorites** tab.



How do I open ServiceNow Studio from App Engine Studio?

When you need a more advanced development environment for an app you are building in App Engine Studio (AES), open the app directly in ServiceNow Studio. A banner appears in AES with information about ServiceNow Studio. Select **Try it out** to open your app in ServiceNow Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Note: Admins can control whether this banner appears by changing the value of the **show_servicenow_studio_banner** system property.

ServiceNow Studio quick start

Complete these tasks to learn the core workflows in ServiceNow Studio, from finding apps and files to packaging and deploying changes.

Where do I start in ServiceNow Studio?

Complete these tasks to learn the core workflows in ServiceNow Studio. For a step-by-step app creation tutorial, see ServiceNow Studio and Build Agent tutorial.

ServiceNow Studio quick start tasks

Task	Description
ServiceNow Studio for ServiceNow IDE developers	Learn about ServiceNow IDE features and capabilities in ServiceNow Studio.
Getting started with Build Agent in ServiceNow Studio	Learn how to start building apps using a conversational interaction with Build Agent.
Personalizing the ServiceNow Studio homepage	Update the theme and development style used in ServiceNow Studio.
Find an app or app file using Search	Search from any page in ServiceNow Studio to find and open metadata records quickly.
Find an app or app file using the Navigator panel	Browse the Navigator panel taxonomy to locate files and metadata records by type.
Create a collection of files in ServiceNow Studio	Curate a collection of your favorite files and apps, and organize them to fit your needs.
Add files to a collection in ServiceNow Studio	Add apps, files, and lists to one or more collections.
Bookmark apps, files, and lists in ServiceNow Studio	Bookmark frequently used apps and app files so you can access them from one location.
Open apps and app files across scopes in ServiceNow Studio	Learn how ServiceNow Studio switches automatically between scope when editing metadata records.
Find info about an app in ServiceNow Studio	Open the App details page to view app metadata and create files.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ServiceNow Studio quick start tasks (continued)

Task	Description
Navigate directly to a table in ServiceNow Studio	Open table views directly by entering shortcut commands such as **.list**, **.do**, and **.config** after a table name in the Search bar.
Focus the Navigator panel on an app or app file	Focus the Navigator panel on any open file or application so the panel reflects your current context.
Create an update set in ServiceNow Studio	Create an update set to package your app changes for deployment.

ServiceNow Studio for ServiceNow IDE developers

Use ServiceNow IDE capabilities within ServiceNow Studio to build apps in source code. All IDE features have been consolidated in the Explorer tab in ServiceNow Studio.

To create apps in ServiceNow Studio, you use ServiceNow Fluent, a domain-specific programming language, to define the metadata that makes up applications. ServiceNow Fluent includes APIs for defining the different types of metadata. For more information, see ServiceNow Fluent.

Workspaces

The concept of workspaces is a bit different in ServiceNow Studio than in ServiceNow IDE. In the ServiceNow IDE, you can add any applications that you're working on to a workspace to navigate through them all from one place. You can create multiple workspaces to group different sets of applications. Workspaces are specific to a user, and applications can be added or removed from a workspace at any time.

In ServiceNow Studio, the **Explorer** tab acts as a single, persistent workspace, where you can add any application that has been converted to Fluent. For that reason, workspaces don't need to be created in ServiceNow Studio to create or group applications.

Fluent source control

ServiceNow Fluent source control is available in ServiceNow Studio to track work done on apps built in source code and converted to Fluent.

Important: Only admins can work on Fluent apps. Don't convert the app if you want delegated developers to continue working on the application.

For more information, see Building apps in source code in ServiceNow Studio and Fluent source control in ServiceNow Studio.

Getting started with Build Agent in ServiceNow Studio

Begin developing on the ServiceNow AI Platform quickly with Build Agent, which enables developers to build, manage, and deploy custom, base system, or global applications and global metadata records.

Build Agent is integrated directly into ServiceNow Studio, giving developers access to AI-assisted development without leaving their authoring environment. Open Build Agent from the ServiceNow Studio interface to perform any development task. For example, use Build Agent when writing scripts, generating code, and working through implementation tasks as you build applications and features.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

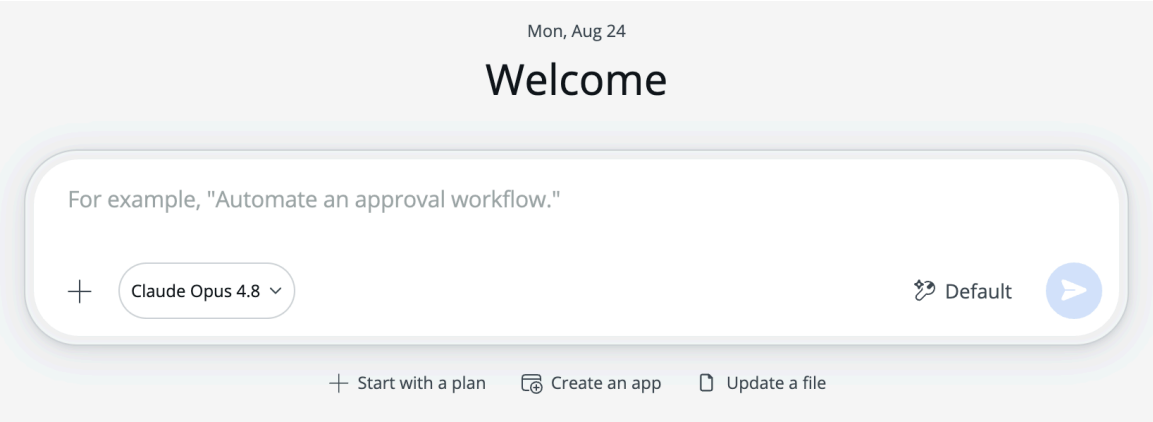
To get started, open ServiceNow Studio and enter a prompt into the main Build Agent chat panel. Build Agent generates code, explains platform concepts, and helps you troubleshoot issues in the context of your current work. You can accept, modify, or discard suggestions as you go. For more information, see Exploring Build Agent.

Note: This topic provides a short overview of Build Agent. For more information and full documentation, see Exploring Build Agent.

How do I access Build Agent?

Access Build Agent from several places.

- The home page of ServiceNow Studio contains the main chat panel for Build Agent.
- The ServiceNow Otto icon in the activity bar opens the list of all current and past conversations with Build Agent.
- From the ServiceNow AI Platform, from the right-click **Configure** menu where you might edit business rules or client scripts, ServiceNow Studio opens with the Build Agent activated with your request.



What can I do in Build Agent?

Build Agent is an AI tool designed for developers within ServiceNow Studio and the ServiceNow IDE. Build Agent acts as an autonomous AI agent capable of independently generating a custom, base system, or global application and global metadata records. For example, a global scope app that uses tables such as incident, problem, and change.

Key Build Agent benefits include:

- Built-in governance to develop securely
- Faster enhancement of existing applications, such as adding new features
- Reduced development backlogs and streamlined deployment of new applications
- Better documentation of existing applications through automated documentation
- Increased developer productivity across skill levels
- Reduction in overall development cost
- Automated testing with Automated Test Framework (ATF) test suite generation and execution

Key capabilities of Build Agent include:

- Creating and updating applications in source code or metadata records.
- Conversational interface.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Support for file uploads.
- Code-related tasks such as rewriting tables, explaining code, validating and enhancing existing applications, and fixing application errors.

Related resources

- Exploring Build Agent
- Build Agent configuration
- Use Build Agent
- Build Agent reference

Personalizing the ServiceNow Studio homepage

Personalize your instance of ServiceNow Studio to match your preferences for development style and theme.

Activity bar

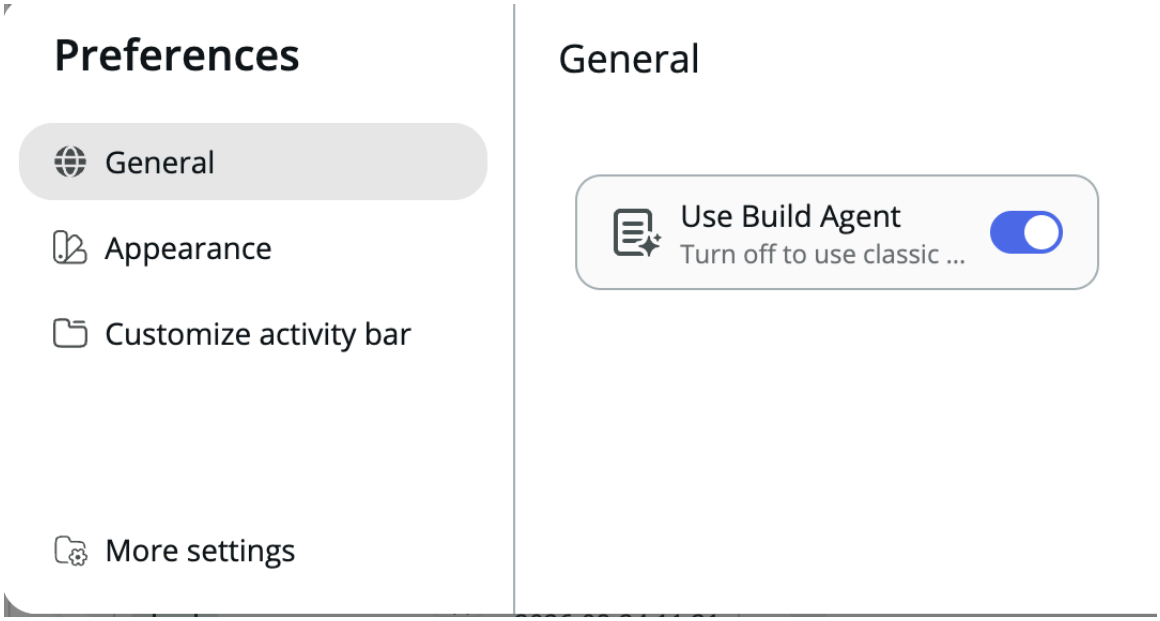
Drag tabs up and down to reorder them to your preference. For example, if you want the **Recent** tab at the top for easy access, drag it to a new location.



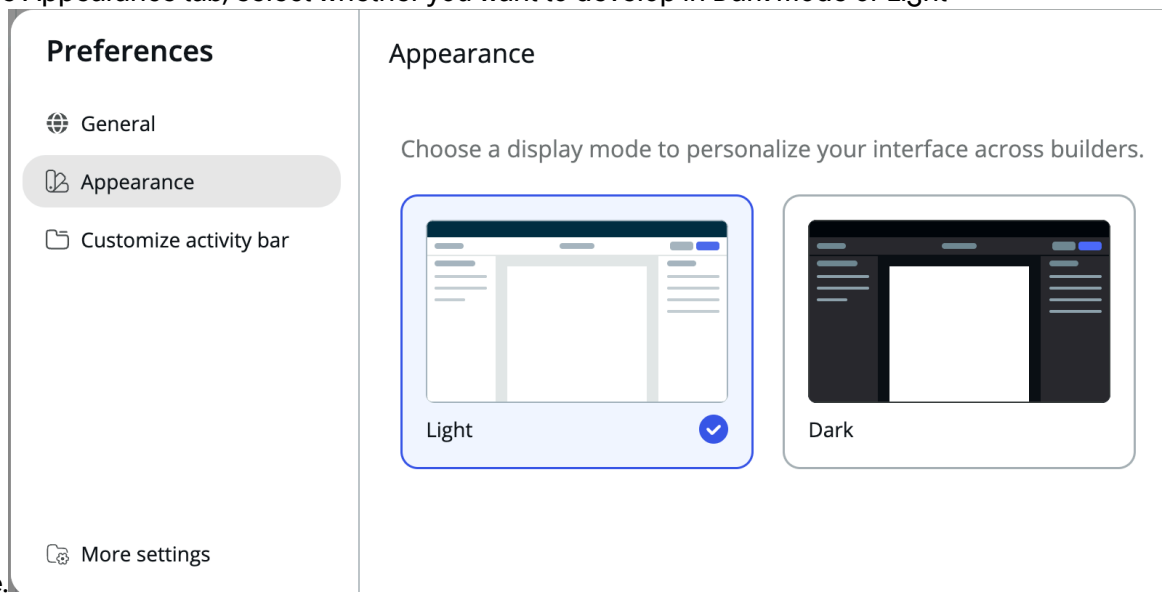
Build Agent and theme preferences

On the General tab, using Build Agent is active by default, but you can turn it off to continue using ServiceNow Otto for app generation.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



On the Appearance tab, select whether you want to develop in Dark mode or Light mode.

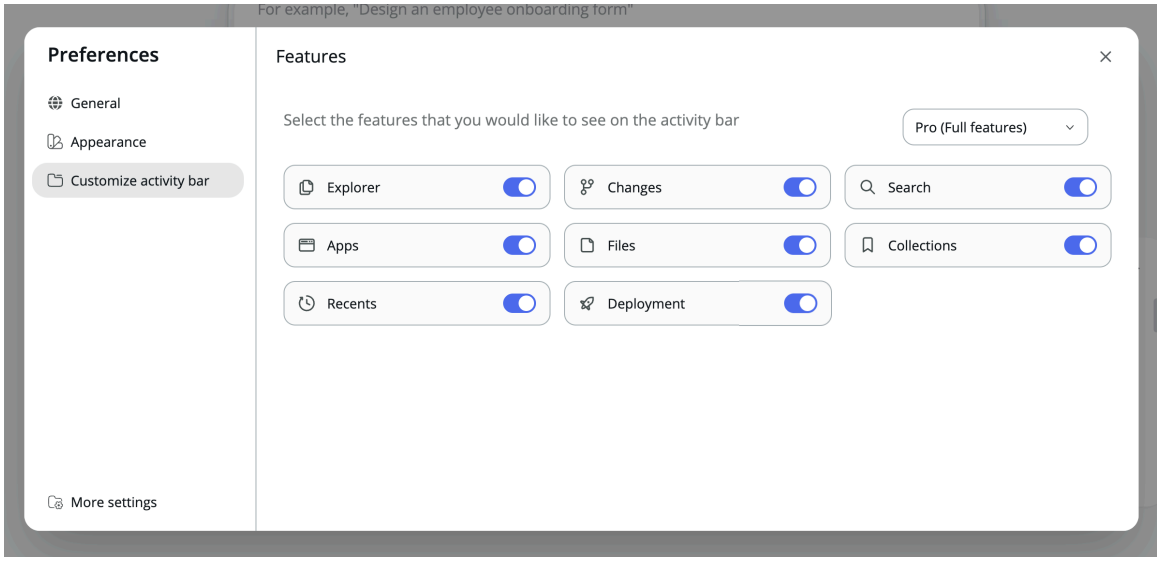


Development mode

Select the tabs you want to see in the Activity bar by choosing a development mode. Access your preferences in the bottom left corner of the screen, select **Preferences**, and select **Customize activity bar**.

If you want minimal tabs showing to keep things simple, choose **Vibe (Minimal features)**, which includes the Apps and Changes tabs. Select **Pro (Full features)** to access all tabs in ServiceNow Studio, or toggle the switch on any feature for a **Custom** view.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Find an app or app file using Search

Use Search in ServiceNow Studio to find and open any app or app file. You can also activate code search to find snippets of code within specific metadata types on your instance.

Before you begin

Role required: admin or delegated_developer

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Select the Search icon in the Navigator panel.

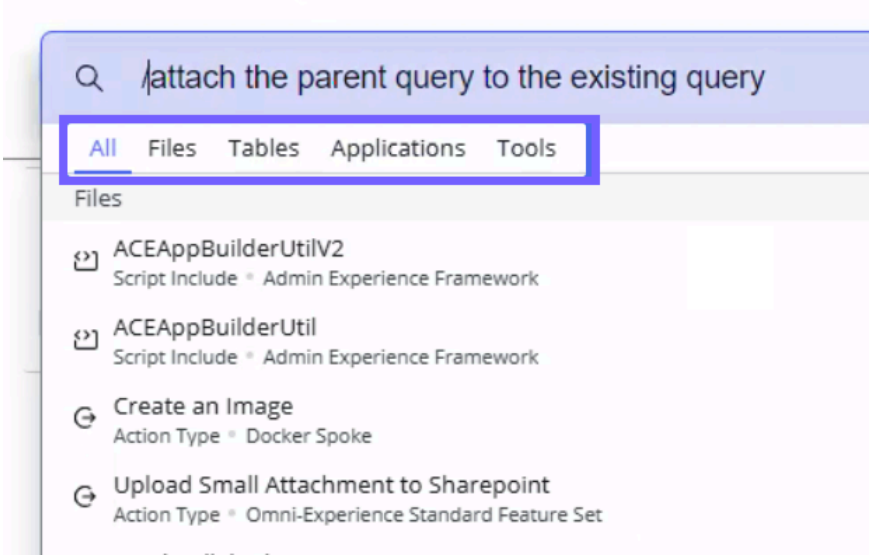
   Note: To access Search from any page using a keyboard shortcut, press `⌘ + /` or `ctrl + /`.

3. **Optional:** To search within code, select the switch next to **Activate code search**.

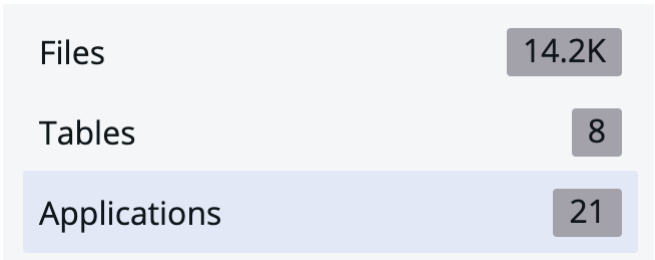
   Note: Code search finds snippets of code within specific application file types. For more information, see Find an app or app file using code search.

4. Enter your app or app file name in the Search bar.
   To navigate directly to a table, follow the instructions in Navigate directly to a table in ServiceNow Studio.
5. Select your app or file from the results list.
6. **Optional:** Filter the results by selecting **Files**, **Tables**, **Applications**, or **Tools** in the search bar.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



7. **Optional:** To expand and filter results further, press `Enter` to open the **Search results** tab, then select a filter to narrow the list.



Find an app or app file using the Navigator panel

Use the Navigator panel in ServiceNow Studio to browse apps and app files by metadata type and open them directly.

Before you begin

Role required: admin or delegated_developer

About this task

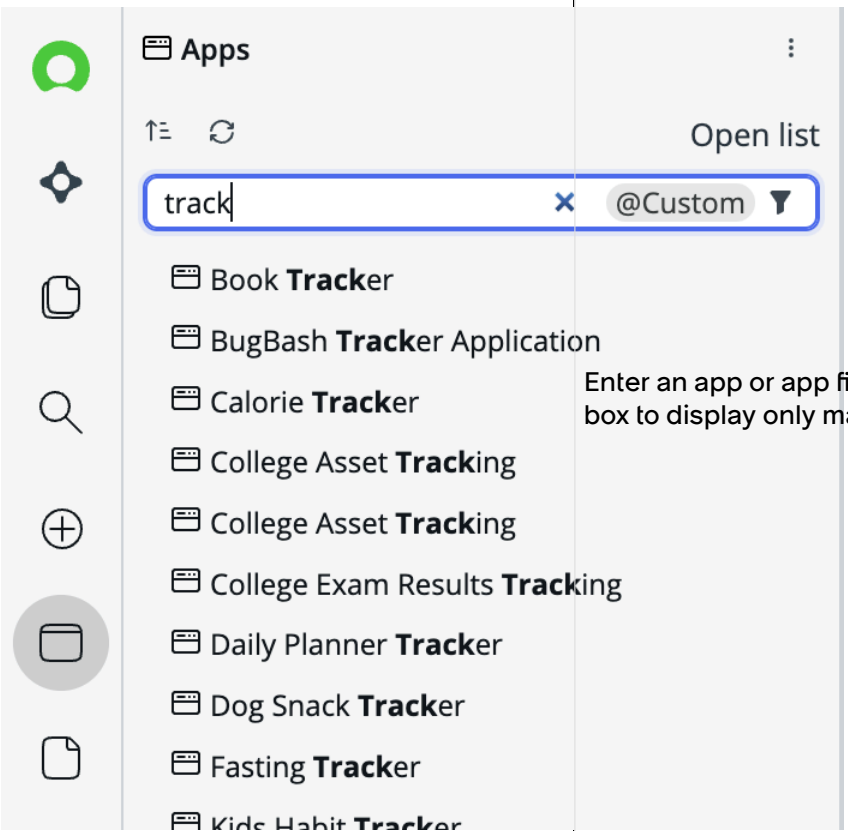
The Navigator panel organizes apps and app files alphabetically by metadata category and type. For example, to find an audio app file, navigate to **File Categories** > **Content** > **Audio**.

For more information about the file types available in the Navigator panel, see ServiceNow Studio Navigator panel taxonomy.

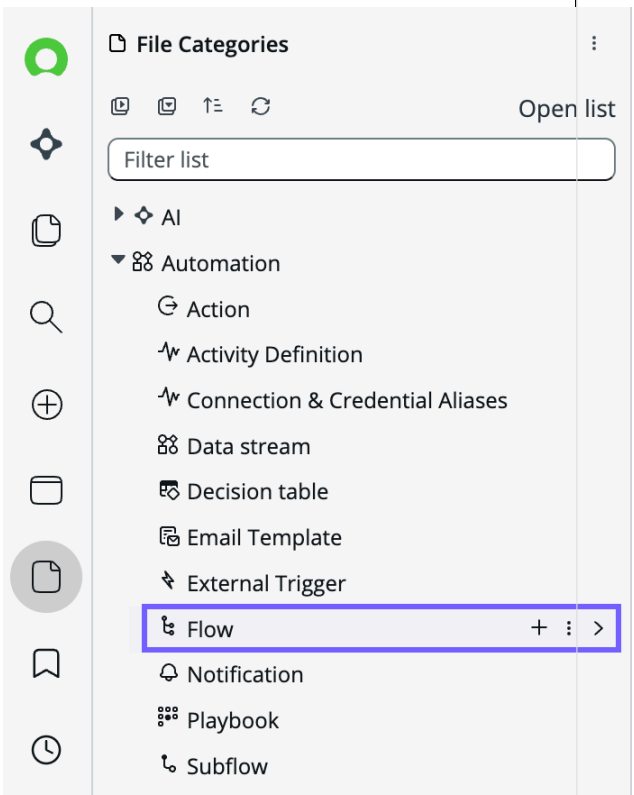
Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Verify that the Navigator panel is open on the left side of ServiceNow Studio.
3. Select the icon that corresponds to the type of file you want to find.
   - To find an app, select the Apps icon .
   - To find an app file, select the File Categories icon .
4. Refine the list of apps or app files using the following options.

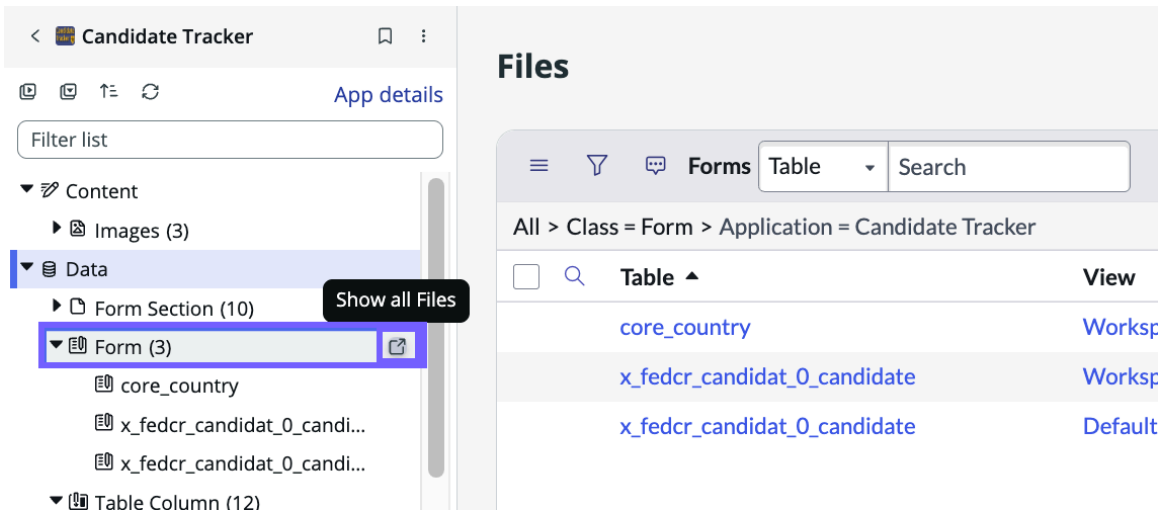
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Choice	Description
Filter icon	**Note:** Only used when finding apps. Select a filter to display applications within a specific scope. Available filters: **All**, **Custom**, **Store**, **Creator Studio**, and **Fluent**. The default filter is **Custom**.
Sort icon	Sort the list of apps or app files in alphabetical or reverse alphabetical order.
Filter list box	 Enter an app or app file name in the Filter list box to display only matching results.
Expand the metadata section	**Note:** Only used when finding app files. For more information, see Metadata record categories in the ServiceNow Studio Navigator. Expand the metadata section that contains the app file you want. For example, to find a flow file, expand **Automation** and select **Flow**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Choice	Description
	

5. **Optional:** Select **Open list** to view a complete list of apps or app files in a new tab. ServiceNow Studio opens a new tab with the complete list of apps or app files.
6. **Optional:** To view all files of a single metadata type, hover over a file type and select **Show all Files**.



7. Select the app or file to open it in a new integrated tab.

Create a collection of files in ServiceNow Studio

Create a collection of files for personal organization in ServiceNow Studio. Create and organize folders, and group apps and files your way.

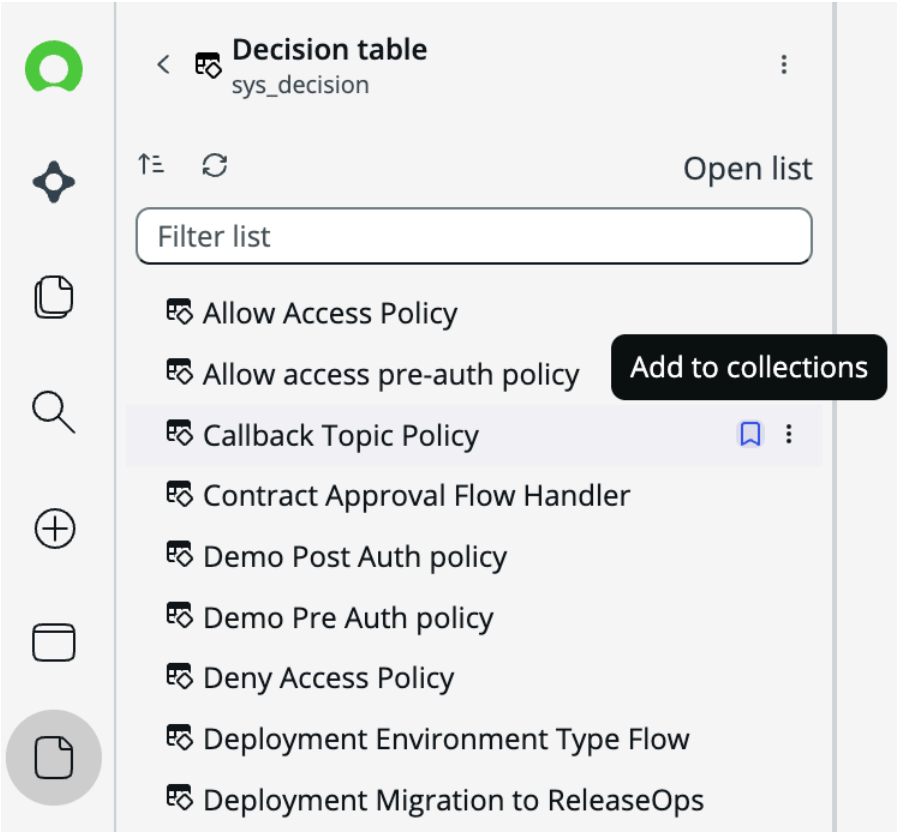
Before you begin

Role required: admin or delegated_developer

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

About this task
Collections extend the current bookmark functionality in ServiceNow Studio by adding the ability to group multiple files and apps in the same folder. Bookmarks are still available as a sub-category within the collections tab in the Navigator panel.

Procedure
1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. In the File Navigator, hover over a file and select the add to collections icon .



3. Select **New Collection**.
4. Enter a name for the collection, and select **Save**.
5. Add files and applications of any scope to the collection.
6. Select **Save**.

Result
The new collection displays in the Navigator panel and is available for use. For more information, see Add files to a collection in ServiceNow Studio.

Add files to a collection in ServiceNow Studio

Group your apps, files and lists into folders and organize your work by stories, epics, or related use cases.

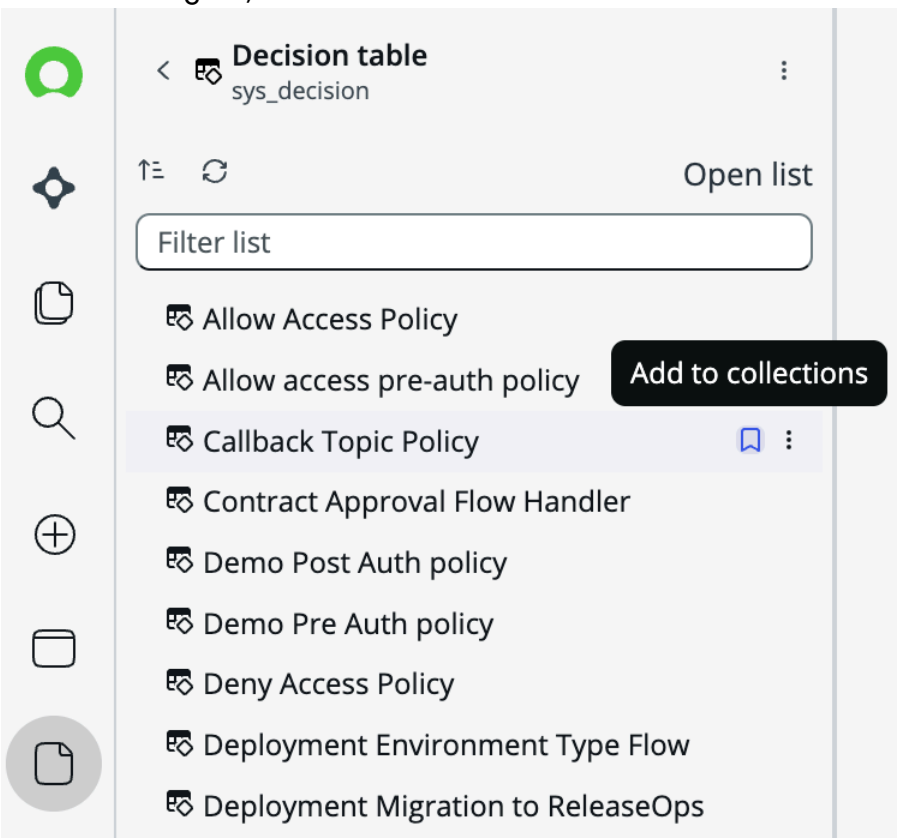
Before you begin
Role required: admin or delegated_developer

About this task
Add files to one or more collections to group them for related work.

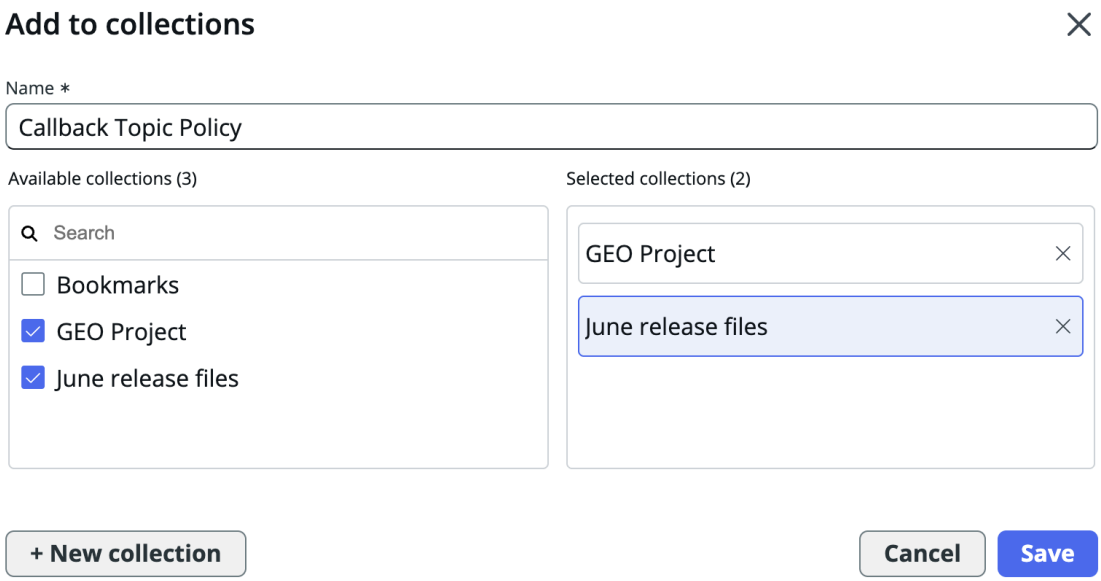
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. In the File Navigator, select a file to add to a collection.



3. Select one or more collections to add the file to.



4. Select **Save**.

Result

The collection contains the file you added and is viewable from the Collections tab in the Navigator panel.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Bookmark apps, files, and lists in ServiceNow Studio

Bookmark apps, metadata files, and lists in ServiceNow Studio so you can access them quickly from the Bookmarks section of the Collections tab in the Navigator panel.

Before you begin

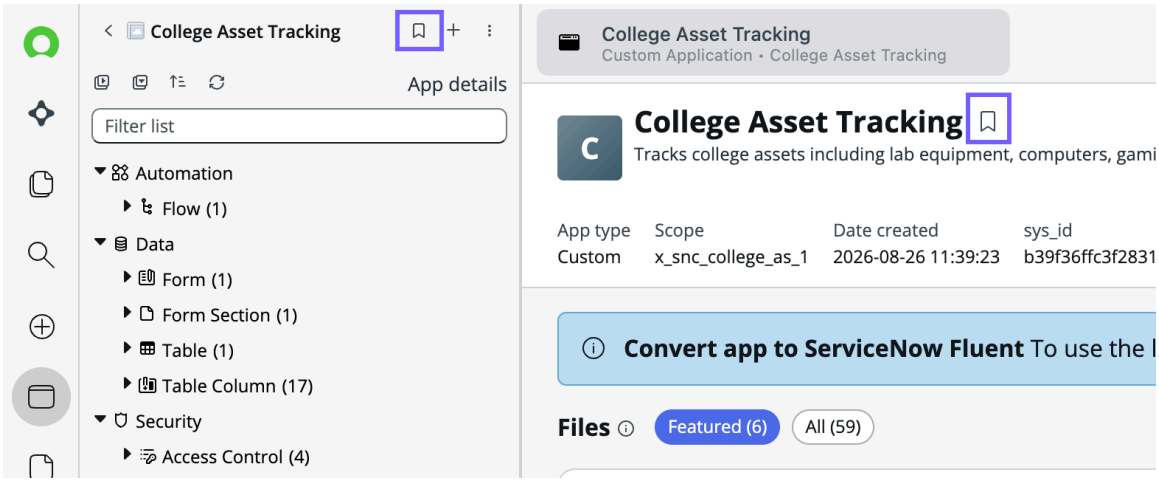
Role required: admin or delegated_developer

About this task

Note: Bookmarks and favorites are distinct ServiceNow capabilities. Bookmarks exist only within ServiceNow Studio, while favorites are accessible throughout your instance. Bookmarks do not automatically become favorites, and favorites do not automatically become bookmarks.

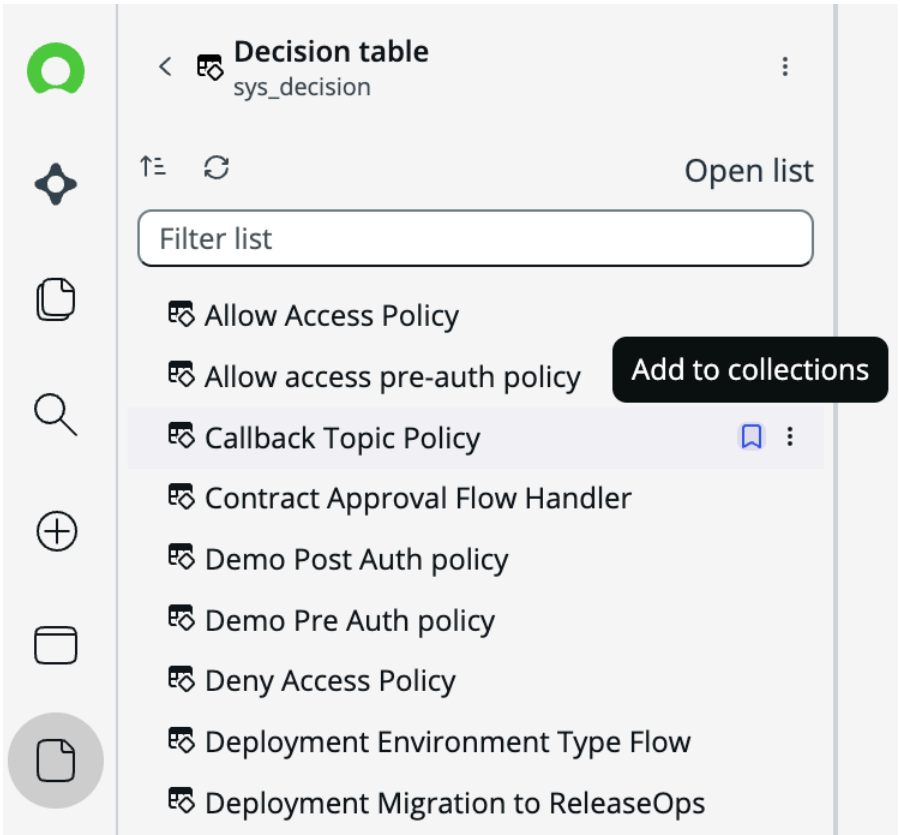
Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Select the Add to collections icon from one of the following locations.
   - To bookmark an open app, select the Add to collections icon next to the app name.

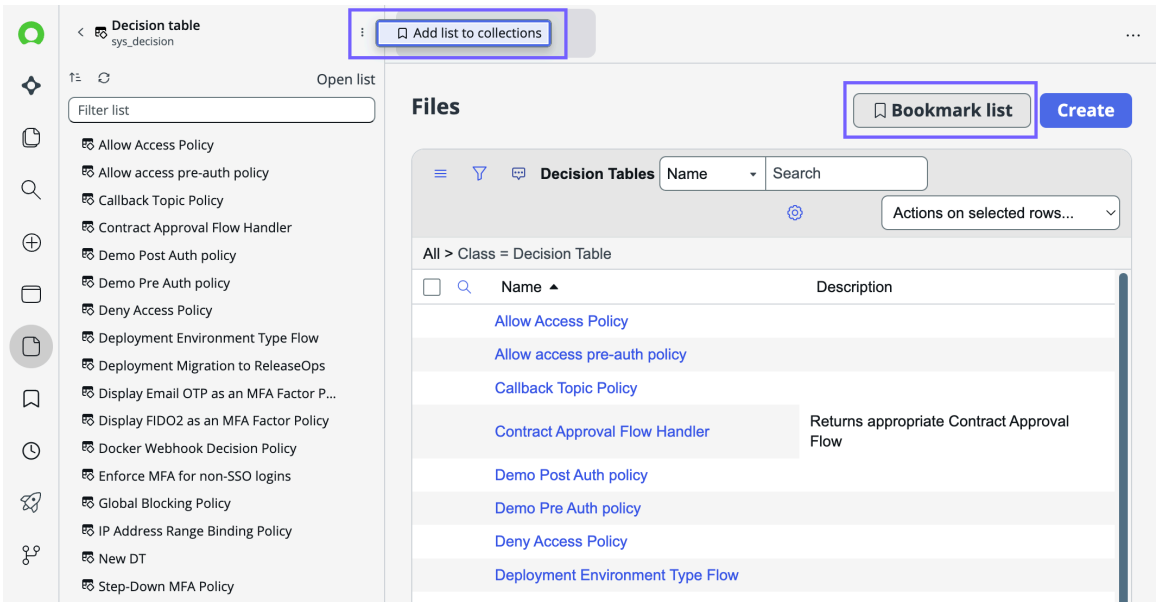


   - To bookmark an app or file in the Navigator panel, hover over it and select the Add to collections icon.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

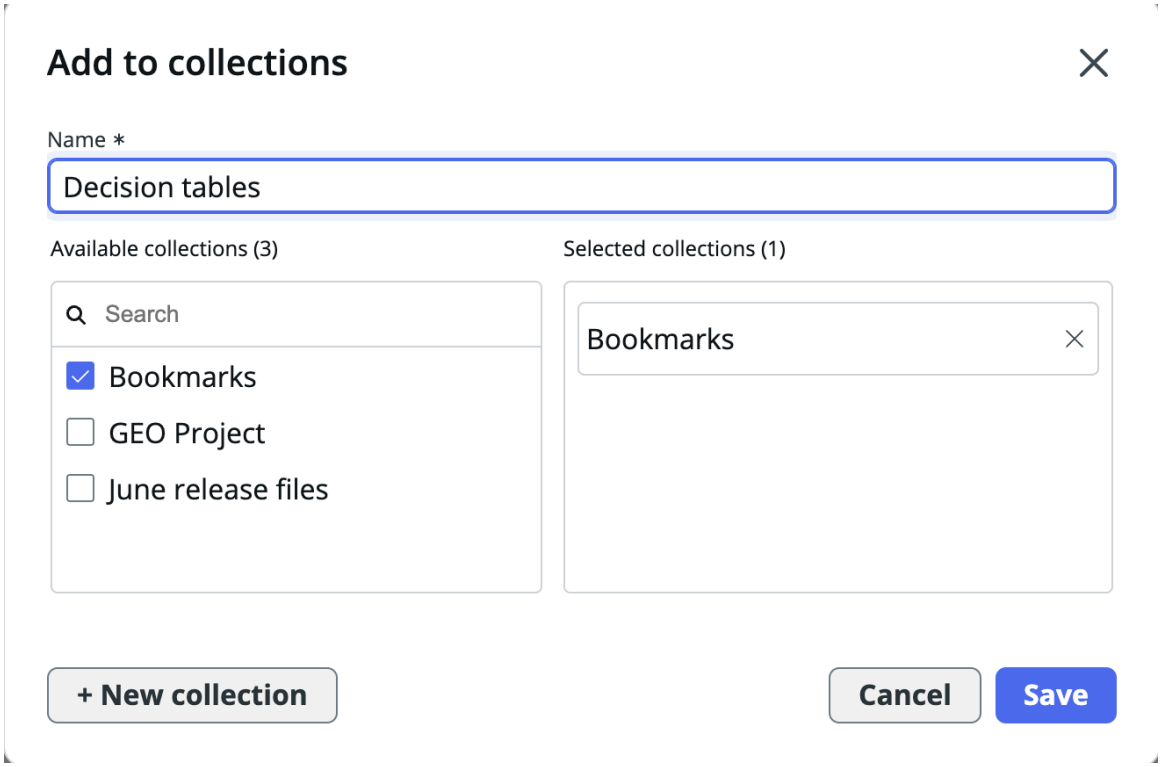


- To bookmark a list, select **Bookmark list** or select the more options icon  , then select **Add list to collections**. For example, open the list of all decision tables from the Automation file category and bookmark the list.



3. Select the **Bookmarks** collection.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



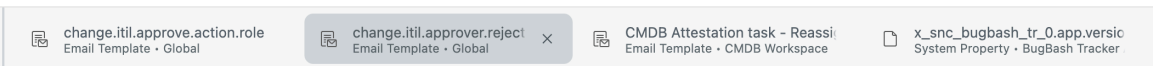
4. **Optional:** Add the app, list, or file to another collection for easy access.
5. Select **Save**.
6. **Optional:** Refresh the Navigator panel to view your new bookmark in the **Bookmarks** collection.
7. **Optional:** To rename the bookmark, select the rename bookmark icon, enter a new name, and select **Save**.

Open apps and app files across scopes in ServiceNow Studio

Open and edit apps and app files in any scope — global or custom — in ServiceNow Studio. The current scope displays at the bottom of the screen and updates automatically when you switch to an app in a different scope.

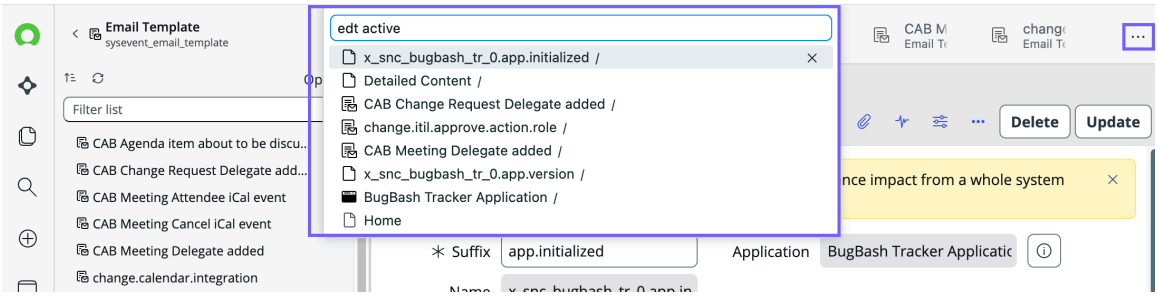
Tabs in ServiceNow Studio

Each app or file you open in ServiceNow Studio opens in its own tab, showing the file name, file type, and scope.



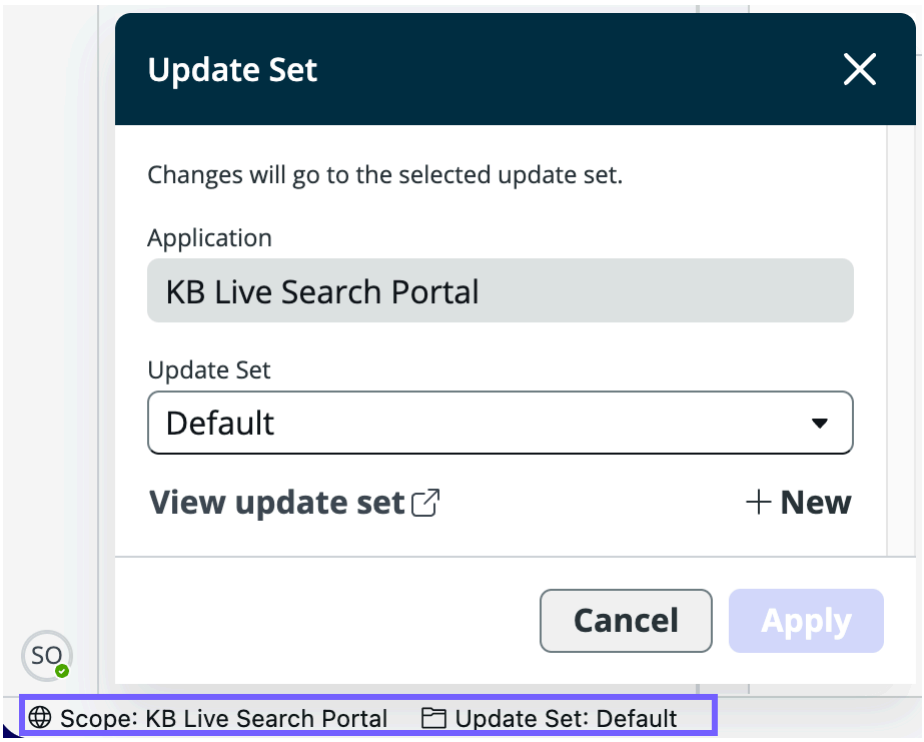
If your browser window is narrow, open tabs may overflow. Select the more actions icon and choose **Show Opened Editors** to see a list of all open files. Select any file to open it in the canvas.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Update sets

The scope and update set associated with the current file appear at the bottom of the screen. Switch update sets or create one to package your changes.



Some builders override automatic scope switching. For example, when you use Table Builder to edit a form or table file, a message indicates that the scope is controlled by the builder.



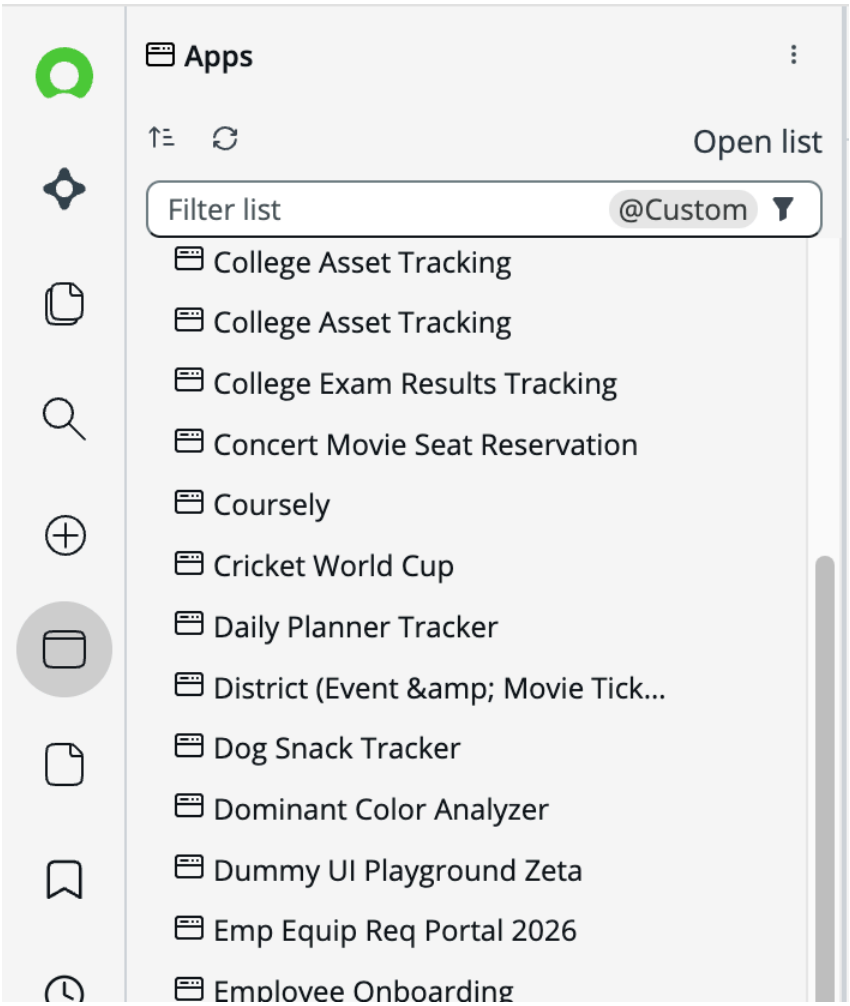
Find info about an app in ServiceNow Studio

Use the App details page in ServiceNow Studio to view metadata about an existing application and take action directly from the page.

Opening an existing application for the first time

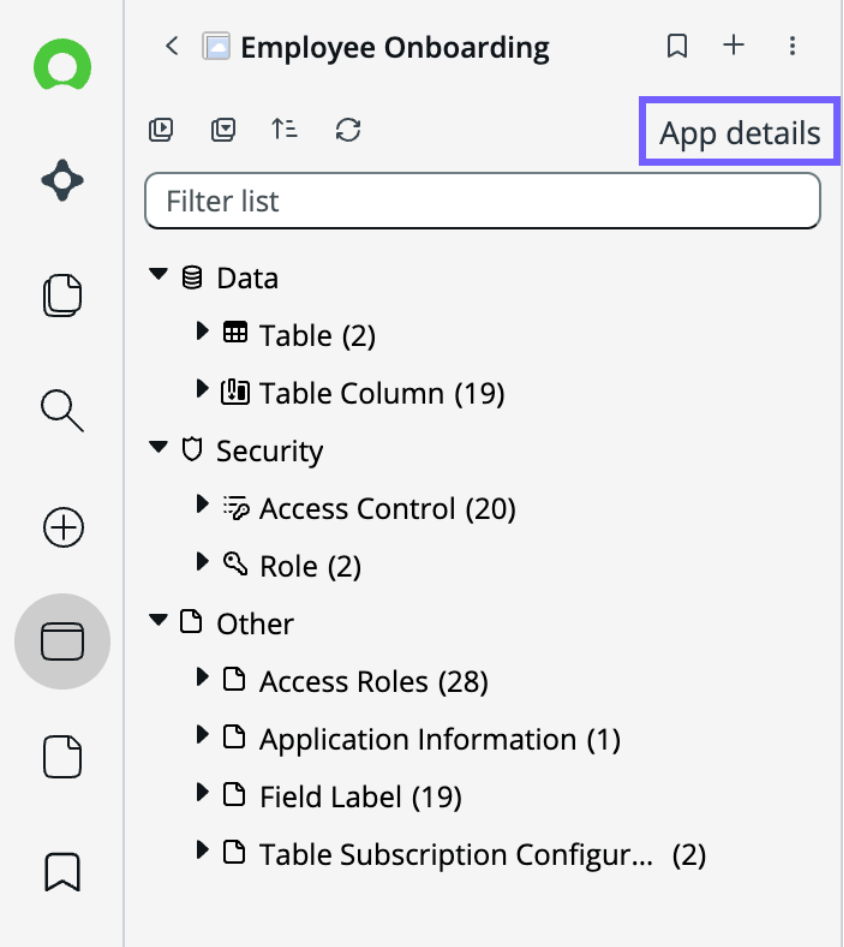
When you open ServiceNow Studio for the first time, open your apps from the Navigator panel on the left side of the screen. Search for the app by name or select it from the list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Selecting an app opens its full contents in the Navigator panel, with all file types organized by category, such as Automation. Select **App details** to open the app in a new tab.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

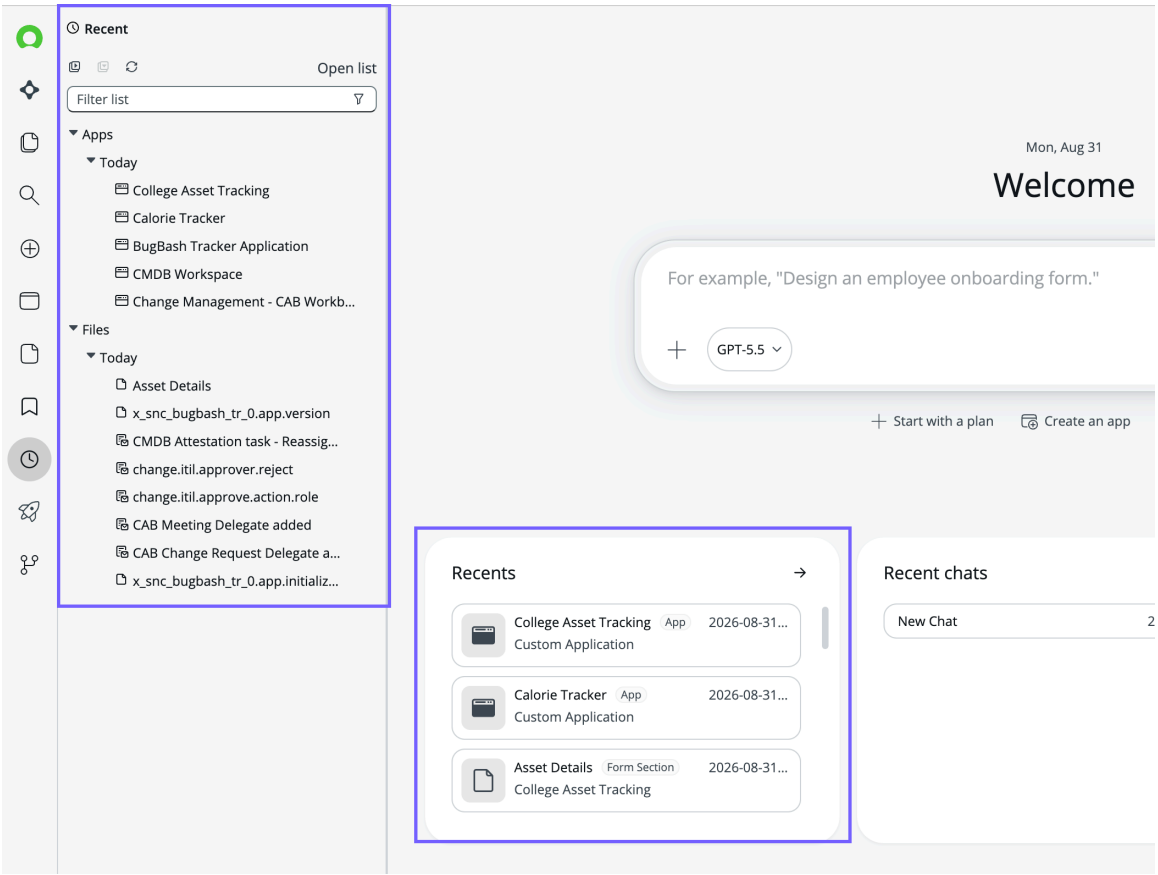


For more information, see Find an app or app file using the Navigator panel.

Opening apps for continued work

After you open an app for the first time, it appears in the Recents section on the ServiceNow Studio home page and the **Recent** section in the Navigator panel.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

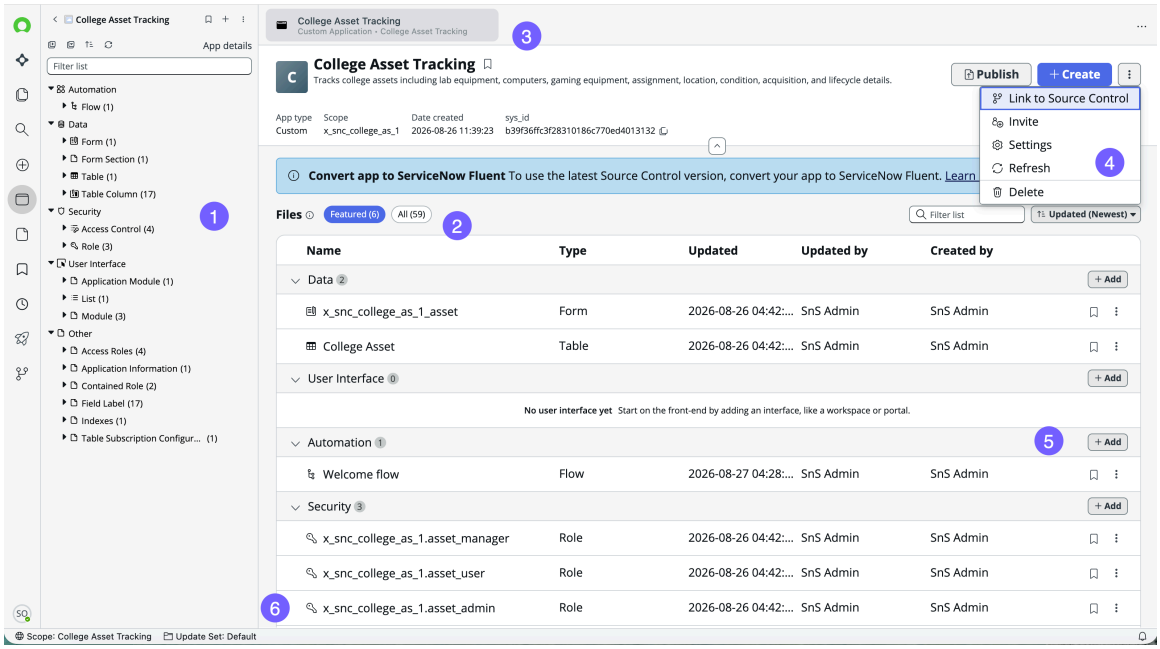


App details features

The App details page shows information about your app and provides access to common app actions.

1. **Navigator panel**: View app files organized by category. Bookmark files or apps, open the full list of any file type, or filter the list to find specific files. For more information, see Bookmark apps and app files in ServiceNow Studio.
2. **Files section**: View featured file types (Data, User Interface, Automation, and Security) or all file types. Filter or sort the list to find the files you want to work with. For more information, see Metadata record categories in the ServiceNow Studio Navigator.
3. **Page header**: View app metadata, invite collaborators, or create files directly using the + Create option. Badges next to the app title — such as Creator Studio, AI, or Fluent — indicate where the app was created and how it must be deployed. Easily convert your app to Fluent by selecting **Convert** on the blue menu. For more information, see App details page and Viewing app origination information in ServiceNow Studio.
4. **App actions**: Select the more options icon ⋮ to link the app to source control, invite collaborators, access settings, or publish, refresh, or delete the app. If the app is linked to source control, available source control operations appear in a menu next to the **Publish** button. For more information, see Applications in ServiceNow Studio.
5. **File records**: Select a file to open it, select **Add** to create files of that type, or bookmark files directly from the record. For more information, see Create an app file in ServiceNow Studio.
6. **Scope and update set**: View the scope and update set associated with the current app at the bottom of the page. The scope switches automatically when you switch between tabs in ServiceNow Studio. Create an update set and associate app changes to it so your changes are ready to deploy. For more information, see Open apps and app files across scopes in ServiceNow Studio and Update sets in ServiceNow Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Navigate directly to a table in ServiceNow Studio

Open any table view directly in ServiceNow Studio using the **Open list** feature in the Navigator panel or search shortcuts in the Search bar.

Before you begin

Role required: admin or delegated_developer

Watch a short video on how to navigate directly to a table.

https://player.vimeo.com/video/1155106448?h=0f72f76aca&badge=0&autopause=0&player_id=0&app_id=58479

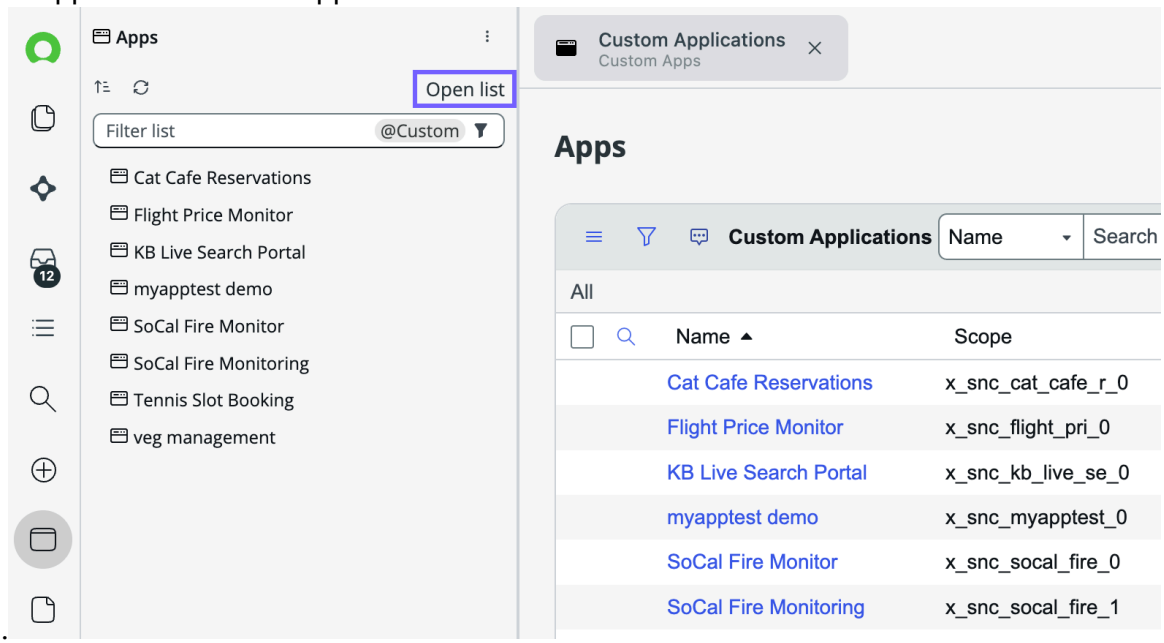
Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Open a table directly using one of the following options.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- In the Navigator panel, with any application or file category selected, select **Open list**.

  The list opens in a new tab. In the example below, this is a list of all apps on the Custom Applications table [sys_app].

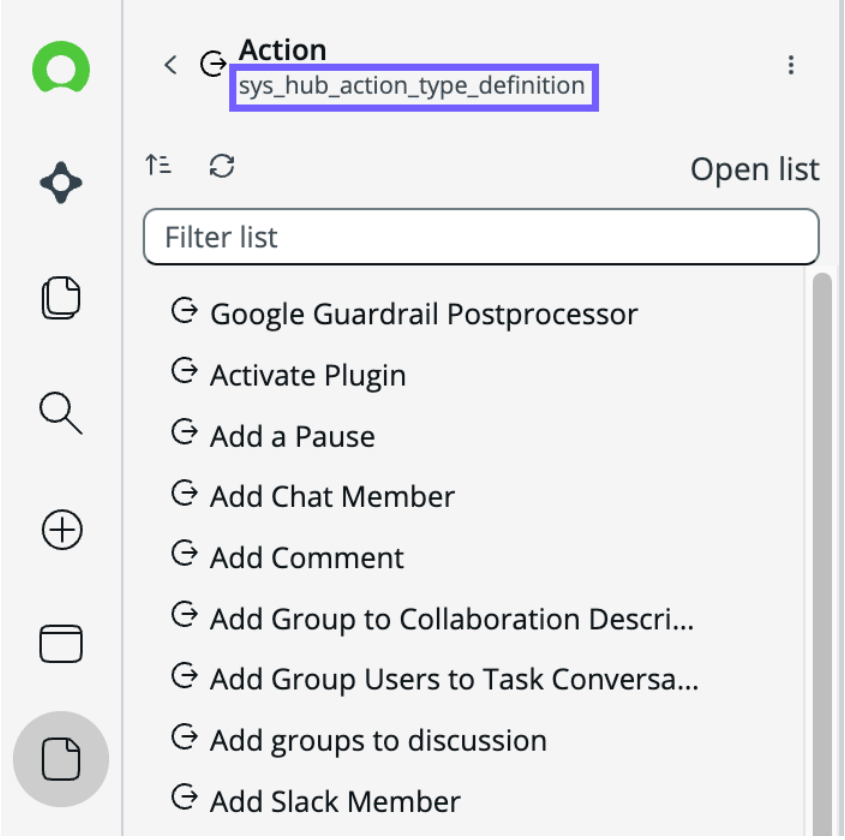


- In the Search bar, enter one of the inputs in the table below and press **Enter**. The table view opens in a new tab.

  Important: The table name must match the name in the dictionary entry for the table. For more information about the primary table associated with each metadata type, see ServiceNow Studio Navigator panel taxonomy.

  To find the primary table name for a metadata type, select the metadata type in the Navigator panel. The primary table name appears below it. In the following example, **sys_hub_action_type_definition** is the table name for the **Action** metadata type.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



When you enter an input, the table view opens in a new tab.

Table opening shortcuts

Input	Result
`<table name>.list`	Opens the list view of the table.
`<table name>.form` or `<table name>.do`	Opens the form view of the table.
`<table name>.config`	Opens the configuration view [personalize_all.do] of the table.
`<table name>.filter`	Opens an empty list view of the table so you can apply filters without loading all records. Use this shortcut for large tables that take a long time to load.

For example, to open the list of all users, search for the Users table [sys_user] using `sys_user.list`. To open a new form view of the Users table, search for `sys_user.do`. To open the Users table configuration, search for `sys_user.config`. To open the Users table without loading any records, search for `sys_user.filter`.

Note: Use upper or lower case to control how the table opens.

- <table_name>.lower case: Opens the table in the content pane.
- <table_name>.UPPER case: Opens the table record or list in a new tab.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Focus the Navigator panel on an app or app file

Focus the Navigator panel on any open app or app file in ServiceNow Studio so the Navigator panel reflects your current work context.

Before you begin

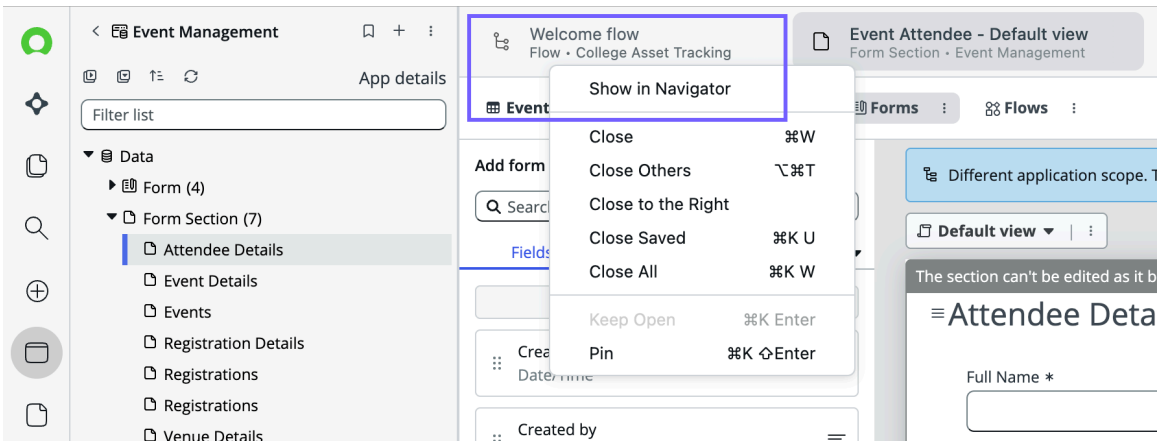
Role required: admin or delegated_developer

Watch a short video on how to focus the Navigator panel on an app or app file.

https://player.vimeo.com/video/1155106522?h=37c7e45cdc&badge=0&autopause=0&player_id=0&app_id=58479

Procedure

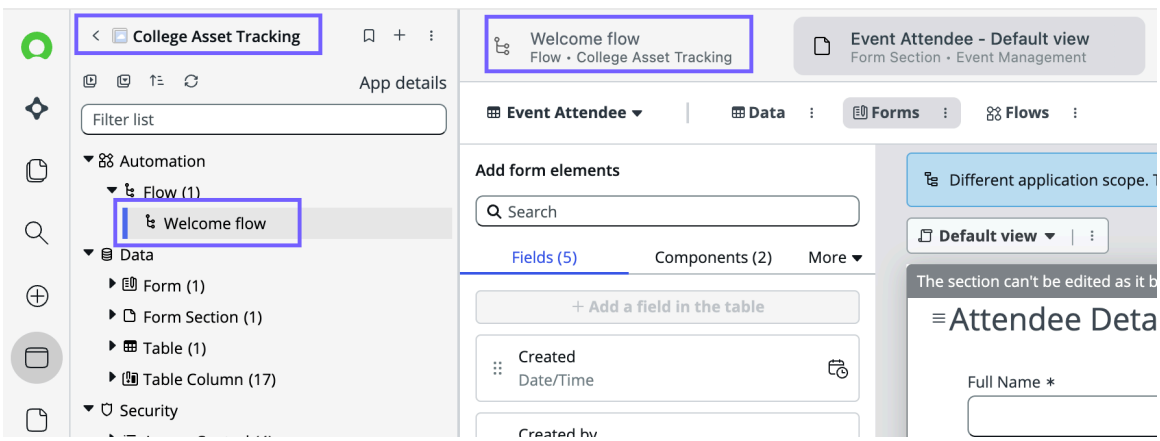
1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Open several files from different applications in the Navigator panel.
3. Select and hold (or right-click) the tab of the file you want to focus on, then select **Show in Navigator**.
   In this example, the file open in the canvas and the Navigator panel are both from the Event Management application. Focus the Navigator panel on the Welcome flow from the College Asset Tracking application.



4. Verify that the Navigator panel now shows the selected file.

   The Navigator panel displays the selected file's location — for example, the **Welcome flow** from the College Asset Tracking application — even with other application files still open.

   Note: The file open in the canvas stays the same until you select a different tab.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Create an update set in ServiceNow Studio

Create an update set in ServiceNow Studio to package your app changes for deployment.

Before you begin

Role required: admin or delegated_developer

About this task

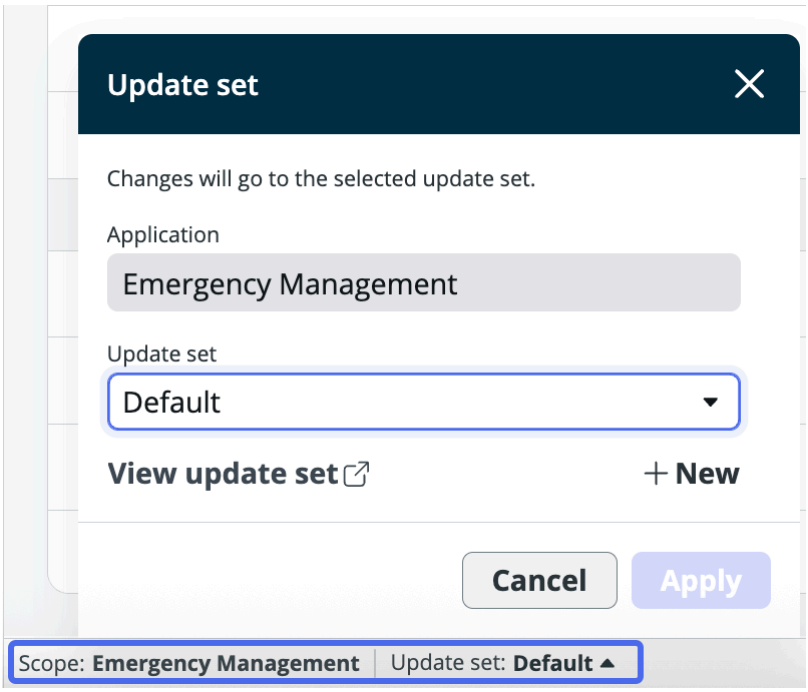
Create an update set without leaving ServiceNow Studio. For more information about update sets on the ServiceNow AI Platform, see General guidelines for planning the update process.

Watch a short video on how to create an update set.

https://player.vimeo.com/video/1155106572?h=620ad6f536&badge=0&autopause=0&player_id=0&app_id=58479

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Open the application that you have changes for that need a new update set.
3. At the bottom-left corner of the canvas, expand the update set dialog by selecting the current update set from the status bar.



4. Select **New**.
5. Enter a name for the new update set.
6. Select **Save**.
7. Select **Apply**.
   The changes you made in your app go to the new update set.

ServiceNow Studio user interface

Use the ServiceNow Studio interface to build, manage, and deploy custom, base system, or global applications and global metadata records. Customize the interface to match your workflow.

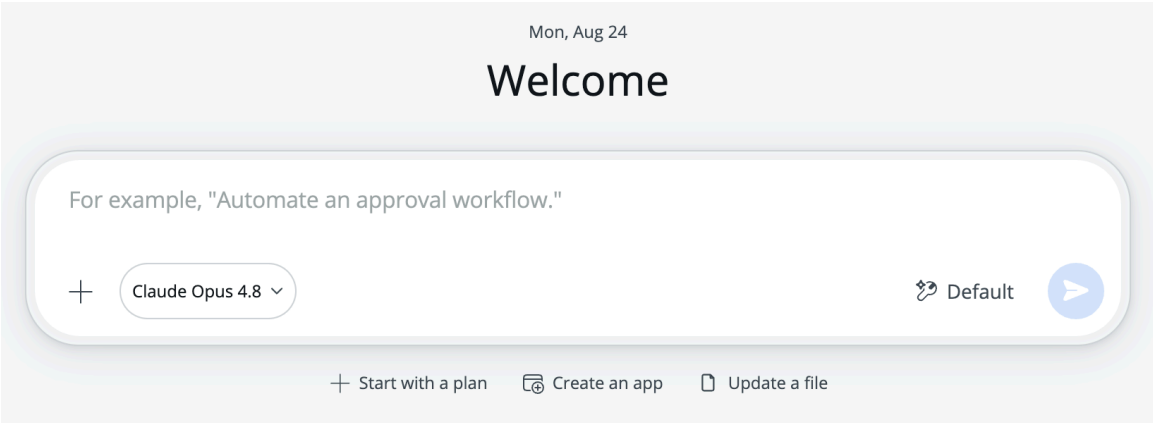
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Access requires an admin or delegated_developer role with access to at least one application.

What is on the ServiceNow Studio home page?

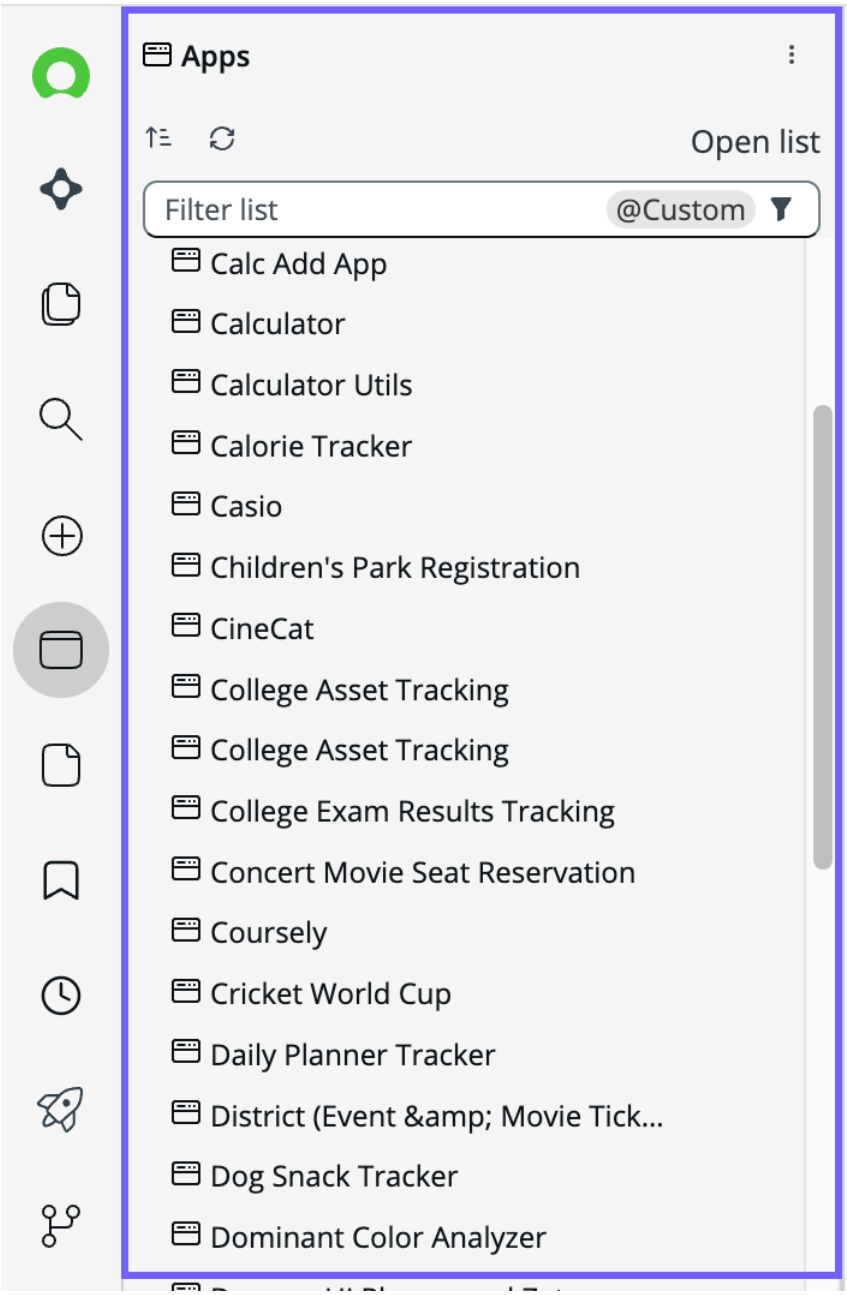
The home page provides access to all tools and services in ServiceNow Studio.

- The main home page shows the Build Agent chat panel, where you can begin creating, updating, or enhancing applications with AI. For more information, see Exploring Build Agent.



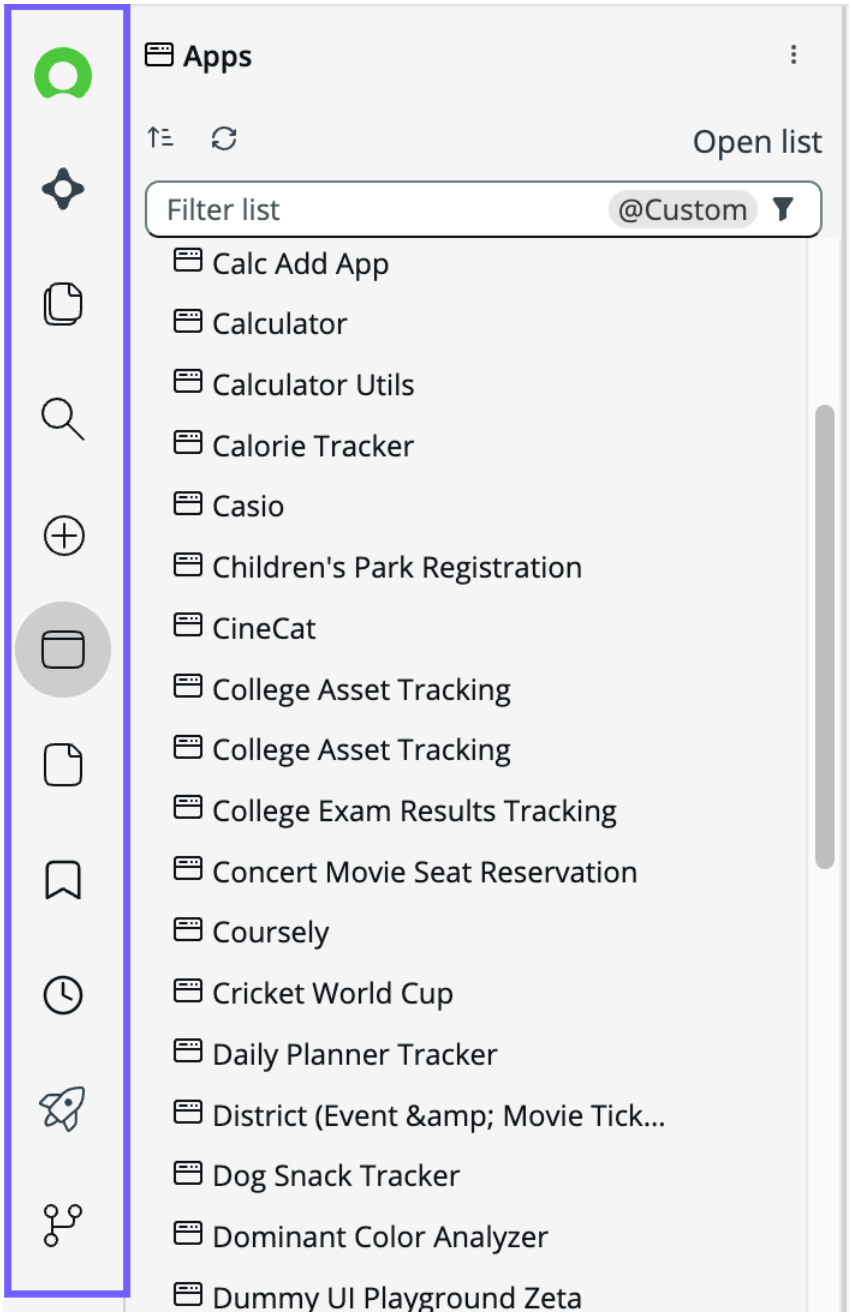
- The Navigator panel enables you to access Build Agent, apps, metadata records, recently opened files, and more. Plan and create apps, create collections, and manage deployments from the Navigator panel. Filter or sort the lists, or select **Open list** to open any list in the platform view.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



- The Activity bar enables you to switch between features to access development tools.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Navigator panel tools

Icon	Navigation item	Description
	Home	Return home to the ServiceNow Studio homepage from other tabs.
	Explorer	Work on apps in source code.
	Conversations	Navigate to conversations you have had with Build Agent.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

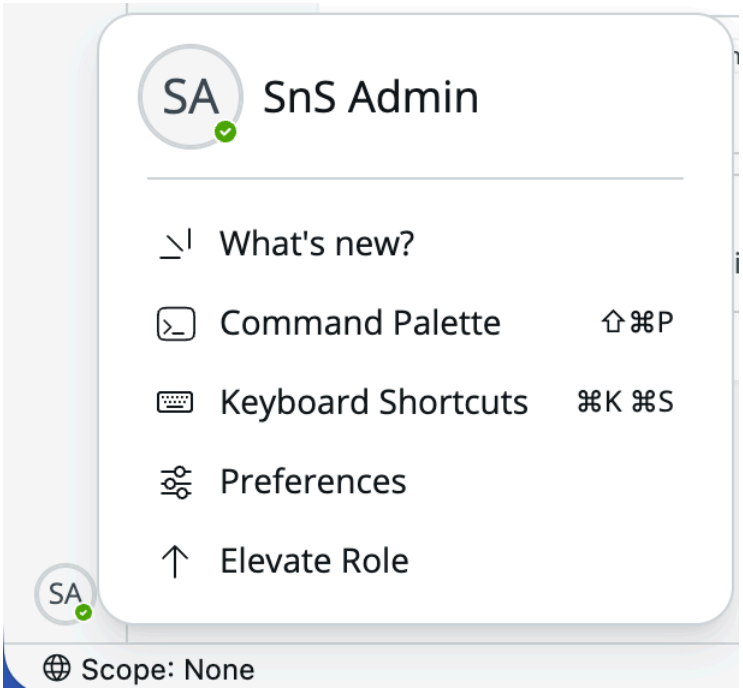
Icon	Navigation item	Description
	Plans	View all the plans Build Agent has created during conversations.
	Search	Search for any file, app, metadata, or tool in any scope on the platform.
	Create	Create an app or file.
	Apps	View all apps.
	File Categories	View, sort, and filter all files and metadata.
	Collections	Access organized collections of bookmarks, files, and apps.
	Recent	Access recently opened files.
	Deployment	Access a list of deployment requests, update sets, and applications.
	Changes	View changes associated with source control.
	User profile	Open the user profile menu, where you can elevate your role and view the ServiceNow Studio command palette, keyboard shortcuts, and user preferences.

- The **User profile** icon  in the activity bar provides access to the ServiceNow Studio command palette, a list of keyboard shortcuts, and user preferences. You can elevate your role, select light or dark theme, and choose whether you want to use Build Agent as your default AI-assisted app generation tool.

  Note: For more information about elevating your role to security_admin, see Elevate your role in ServiceNow Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For more information, see ServiceNow Studio settings.

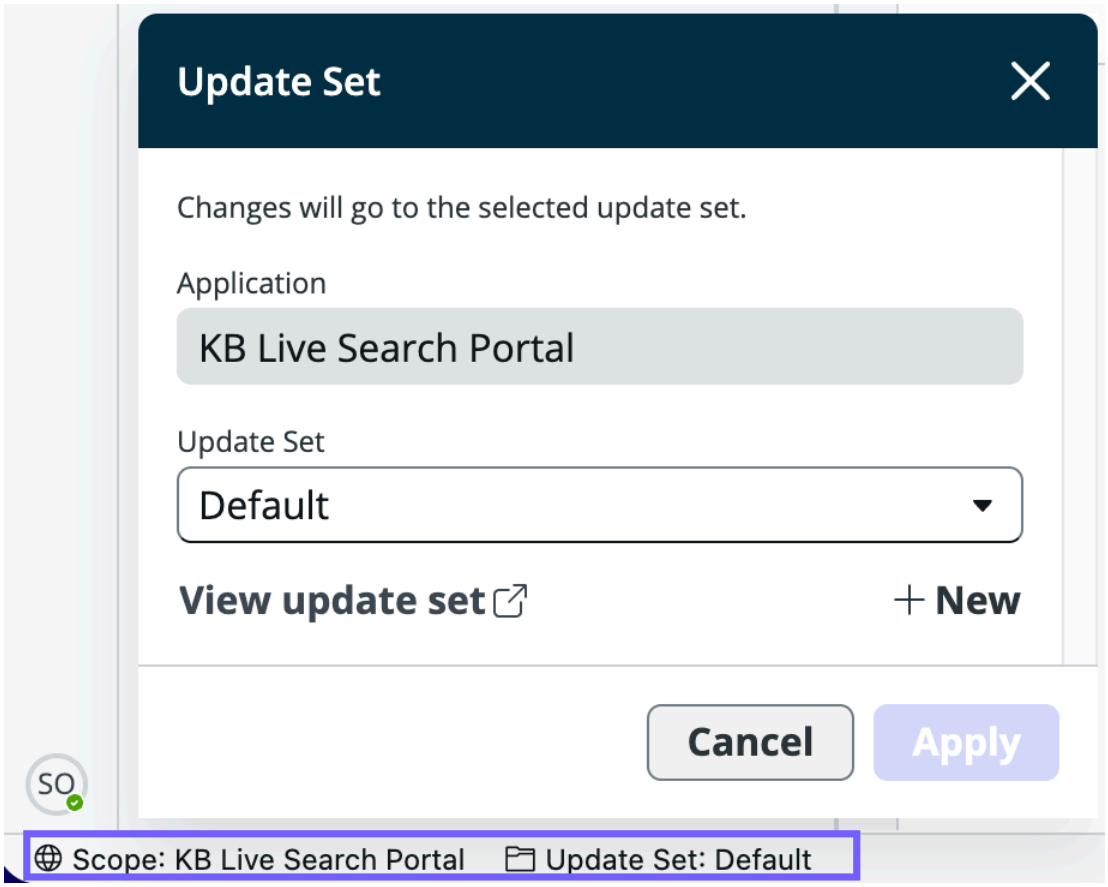


How do I view the scope and update set?

When you have an app or file open, the status bar shows the current scope and the update set associated with the application. To change the update set or create one, open the default update set and select a different one directly within the app. To see all updates associated with the current update set, select **View update set**.

For more information, see Update sets in ServiceNow Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



ServiceNow Studio for App Engine Studio (AES) developers

If you currently develop in App Engine Studio, use ServiceNow Studio to access advanced development capabilities while retaining the core features you already use.

ServiceNow Studio is organized differently from App Engine Studio. Use the following sections to find the features you are familiar with from App Engine Studio.

What features work the same way in ServiceNow Studio?

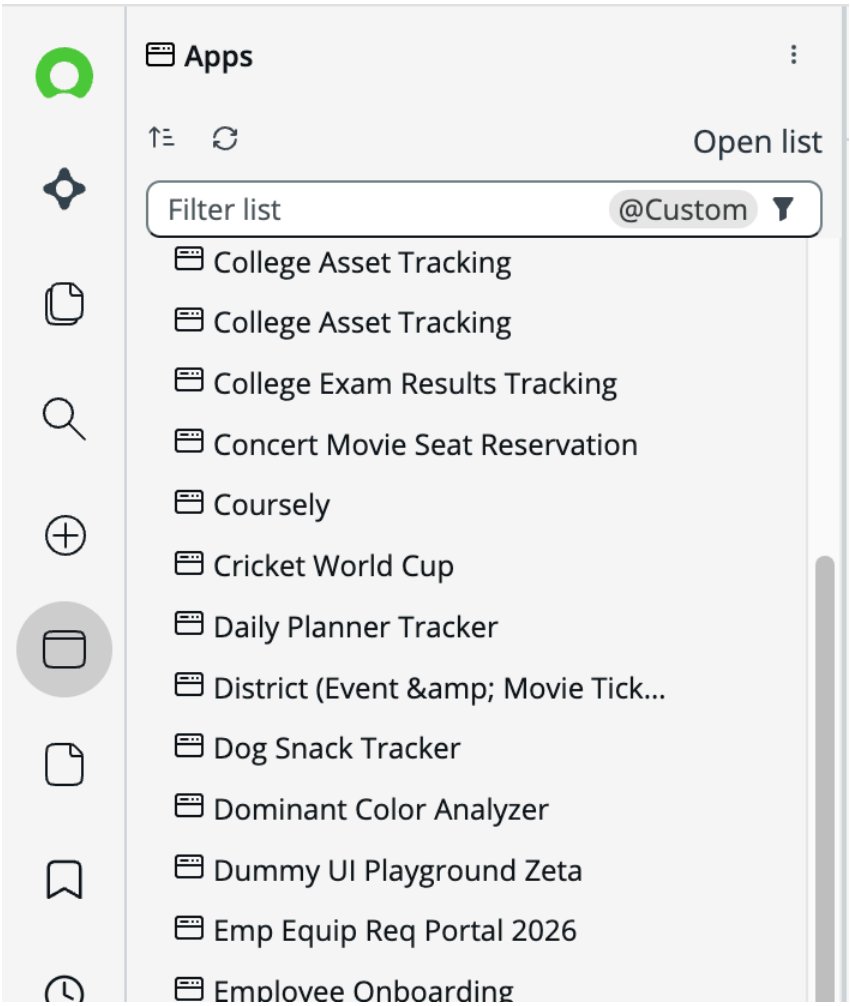
Several key features from AES work the same way in ServiceNow Studio, including app creation, source control integration, and collaboration. For more information, see the following topics.

- Create an application in ServiceNow Studio
- Source control integration in ServiceNow Studio
- Collaborating on apps using ServiceNow Studio

Where are my apps?

When you first open ServiceNow Studio, not all of your apps are visible by default. To find them, open the Navigator panel, select the Apps section, and filter for custom apps.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



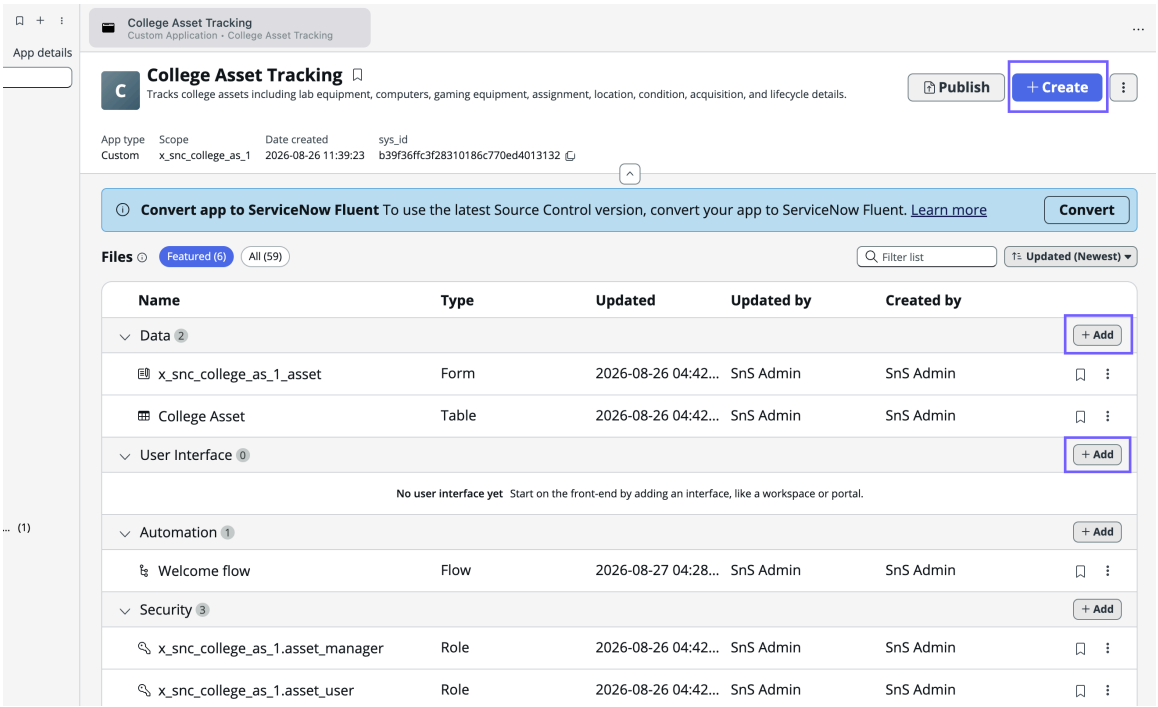
To access frequently used apps, app files, and lists quickly, bookmark them. For more information, see Bookmark apps, files, and lists in ServiceNow Studio.

How do I work with different file types in ServiceNow Studio?

In AES, you can create files in four categories: Data, Experiences, Logic and Automation, and Security.

ServiceNow Studio supports all four of those categories and many additional file types. Unlike AES, where you must be inside an app to create a file, ServiceNow Studio lets you create files from the home page and specify the application afterward. You can also select **Add** next to any category on the App details page to create a file of that type.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



App files define how an application functions. For example, add a business rule to perform server-side actions when a record changes, or add AI files to automate tasks.

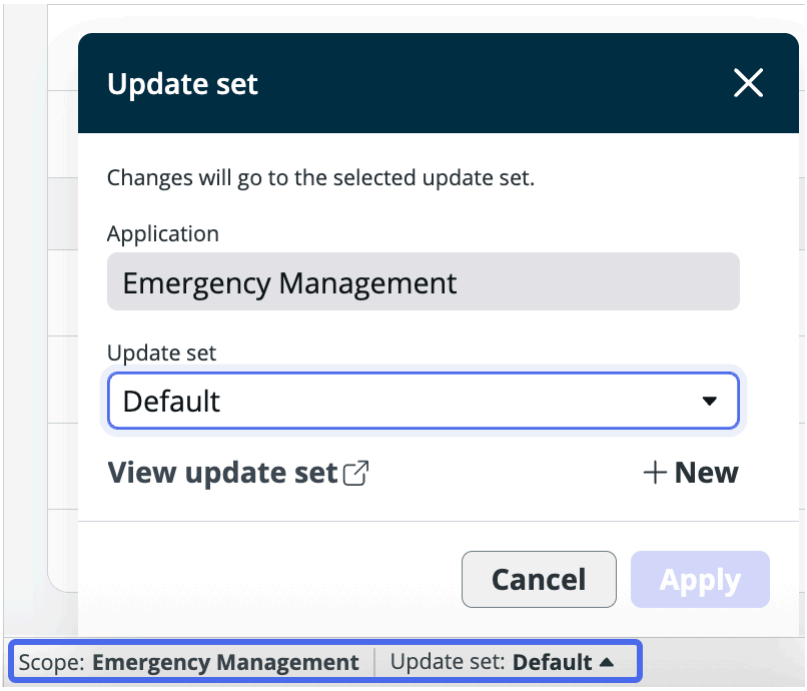
For more information about file types and how to work with them, see the following topics.

- Create an app file in ServiceNow Studio
- Metadata record categories in the ServiceNow Studio Navigator
- ServiceNow Studio Navigator panel taxonomy

How does scope work in ServiceNow Studio?

In ServiceNow Studio, the scope updates automatically based on which app you are working in — you do not need to switch it manually. The current scope and update set appear at the bottom left of the page for each open application.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Some builders override automatic scope switching. For example, when you use Table Builder to edit a form or table file, a message indicates that the scope is controlled by the builder.

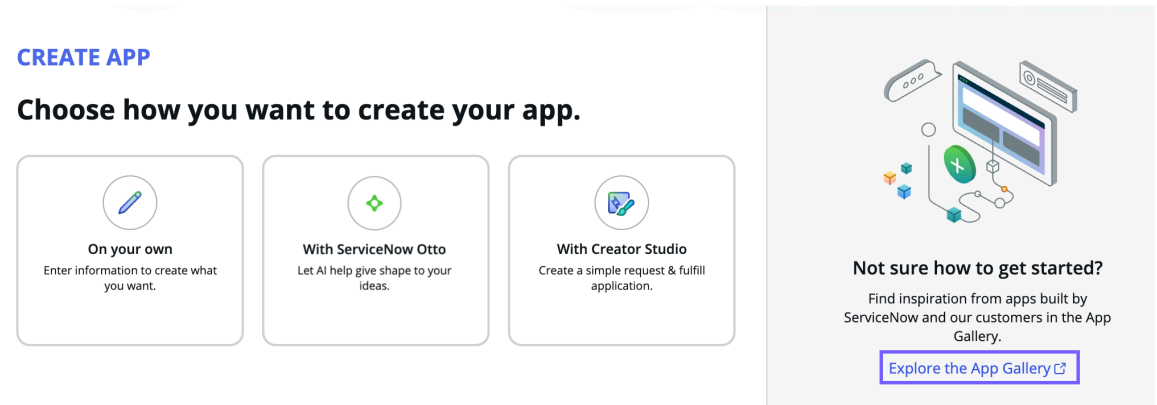


For more information, see Open apps and app files across scopes in ServiceNow Studio.

Where are app templates?

AES provides a library of application templates for creating apps. ServiceNow Studio does not have templates, but the App Gallery provides examples of completed apps and metadata files you can use for reference. To access the App Gallery, select **Create** > **App** and select the App Gallery link on the right side of the page.

Note: The App Gallery is for reference only. You cannot create an application directly from App Gallery files as you would from a template in AES.



Builders in ServiceNow Studio

Builders are specialized tools in ServiceNow Studio for editing metadata records associated with tables that extend sys_metadata. Use builders to create and edit file types such as decision tables, flows, and forms as part of your app development.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

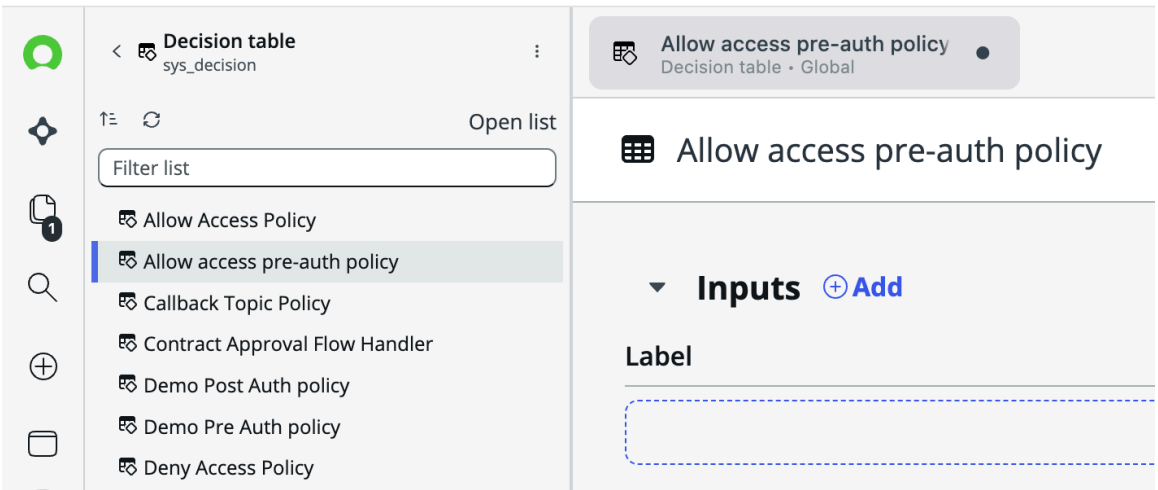
Builders provide features such as inline editing, customizable views, and integrated collaboration. Each builder is independent of specific design libraries or implementations, so it adapts to different technologies and frameworks.

When you open files such as decision tables, flows, or tables in ServiceNow Studio, the files open in a new tab in their designated builder. Some file types, such as mobile app configurations, open in a new browser tab. To access a builder, you must first open a file or application — builders cannot be opened independently in ServiceNow Studio.

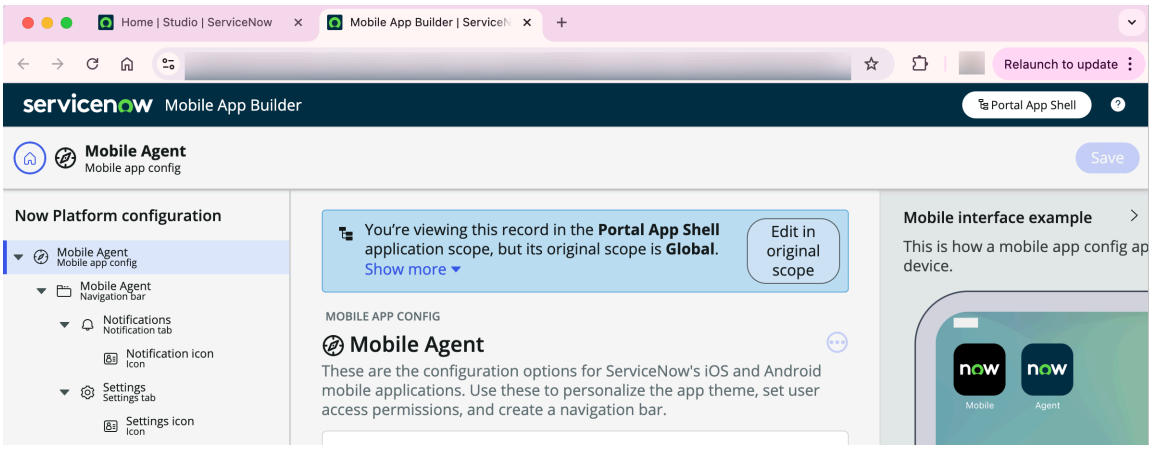
For a full list of the builders integrated in ServiceNow Studio, see Integrated development tools for ServiceNow Studio.

How do builders open in ServiceNow Studio?

Decision tables and most other automation file types open in new tabs within ServiceNow Studio.

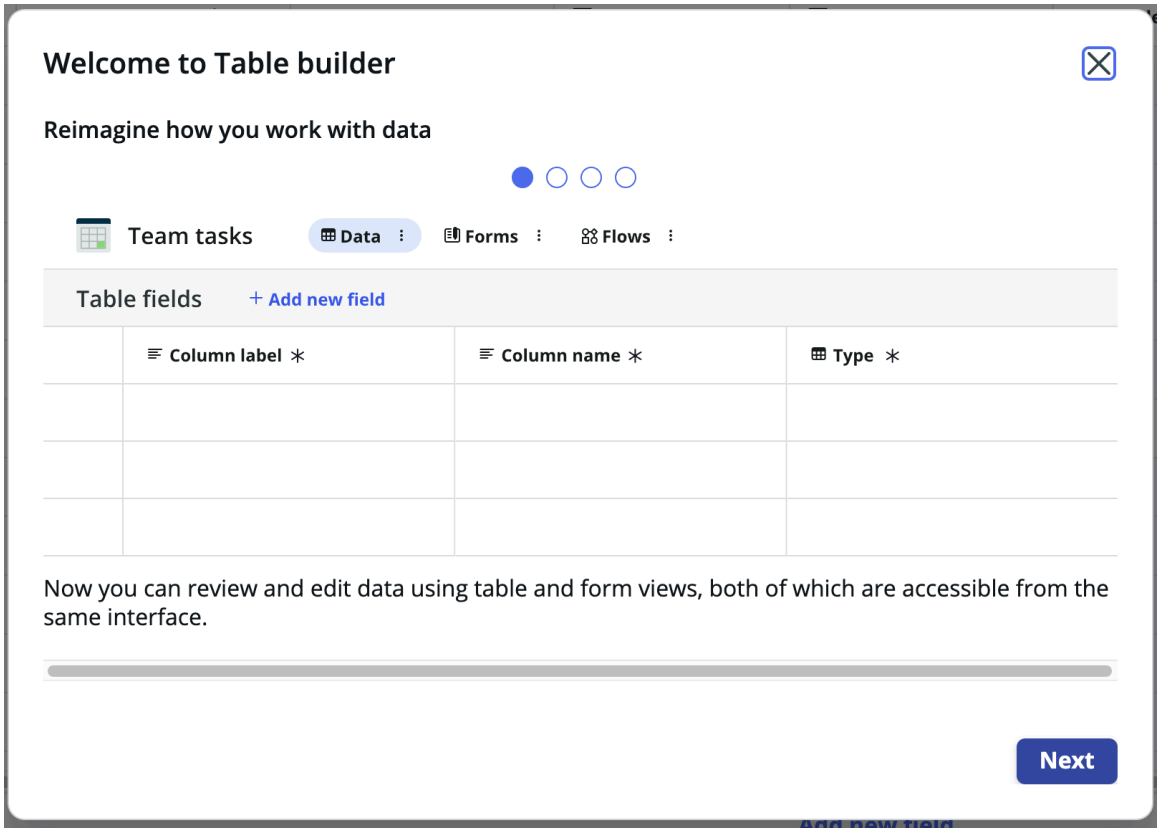


Mobile App Builder opens in a new browser tab.



Some builders, such as Table Builder, display a short tutorial when you first open them.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



How does scope work with builders?

Some builders override automatic scope switching. For example, when you use Table Builder to edit a form or table file, a message indicates that the scope is controlled by the builder.



Integrated development tools for ServiceNow Studio

Use the development tools in ServiceNow Studio to build apps, flows, tables, forms, and more — all without leaving the platform.

What development tools are available in ServiceNow Studio?

Use the following development tools to build apps, flows, processes, tables, and more in ServiceNow Studio. Some tools include the word "Builder" in their name, but all tools listed here enable you to build different parts of an app. Some are automation tools; others are for building your user interface. For more information about builders, see Builders in ServiceNow Studio.

AI

Builder	Description
AI Agent Studio overview	Create, manage, and test AI agents and agentic workflows to automate self-executing processes and achieve business goals.
AI Skill Kit	Create and activate custom prompts and skills for Now Assist using the Now Assist Skill Kit plugin.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Builder	Description
Build Agent	Use Build Agent, an autonomous AI agent, for creating and updating ServiceNow® applications. You can also ask Build Agent general ServiceNow development questions.

Automation

Builder	Description
Decision tables in Workflow Studio	Embed business logic into if-then decision rules. Decision tables read data from inputs, evaluate conditions, and return one or more results when all conditions for a rule are met.
Flows, subflows, and actions in Workflow Studio	Automate work using multi-step flows built from reusable components, without writing code.
Playbooks in Workflow Studio	Author cross-enterprise workflows and create a unified process. Use playbooks to provide end users a simplified, task-oriented view of your process.

Data

Builder	Description
Form Builder	Customize any ServiceNow AI Platform record or form. Form Builder is most often accessed through Table Builder on the Forms tab.
Table Builder	Design tables, forms, and flows visually using a single user interface.

Integration

Builder	Description
Integration Hub Imports	Configure, run, and schedule data imports through a single interface. Integration Hub Imports consolidates data integration capabilities in one environment, eliminating the need to manage multiple forms across the platform, and guides you through configuring data sources, target tables, and data mapping.

Mobile builder

Builder	Description
Mobile App Builder	Configure how ServiceNow applications display on mobile devices.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

User interface

Builder	Description
Catalog Builder	Create or edit a catalog item (catalog item or record producer) using a visual, guided experience with specified restrictions. Catalog Builder enables you to delegate the creation and maintenance of the catalog.
Service Portal	Create or update the portal page layout and add or remove widgets on the page.
Service Catalog	Create or edit a catalog item, which can be a record producer, an order guide, or a content item.
Survey designer	Create survey categories and questions, configure their details, and publish the survey to specific users or groups.
UI Builder	Build pages for workspaces generated by App Engine Studio or custom web experiences using Next Experience Components and custom web components.
Workspace Builder	Automate processes in a single design environment using Workspace Builder.

Other tools

Builder	Description
Script Debugger	Debug server-side JavaScript for users with the script_debugger role.

ServiceNow Studio and legacy products

If you previously used Guided Application Creator or the legacy Studio product to develop apps, use ServiceNow Studio to continue your app development. ServiceNow Studio replaces both products with a unified development environment on the ServiceNow AI Platform.

Deprecated applications

Guided Application Creator (GAC) and legacy Studio are both deprecated as of the November 2024 release.

No additional configuration is required to switch to ServiceNow Studio. Apps created in both legacy products are compatible with ServiceNow Studio.

Note: GAC permissions for app creation are not supported in ServiceNow Studio in the November 2024 release. If you have GAC permissions and want to create an app, create the app in legacy Studio, then open and edit the app in ServiceNow Studio. Contact your company admin for more information.

Both Guided Application Creator (GAC) and legacy Studio are supported until a future release.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

What does ServiceNow Studio improve over legacy products?

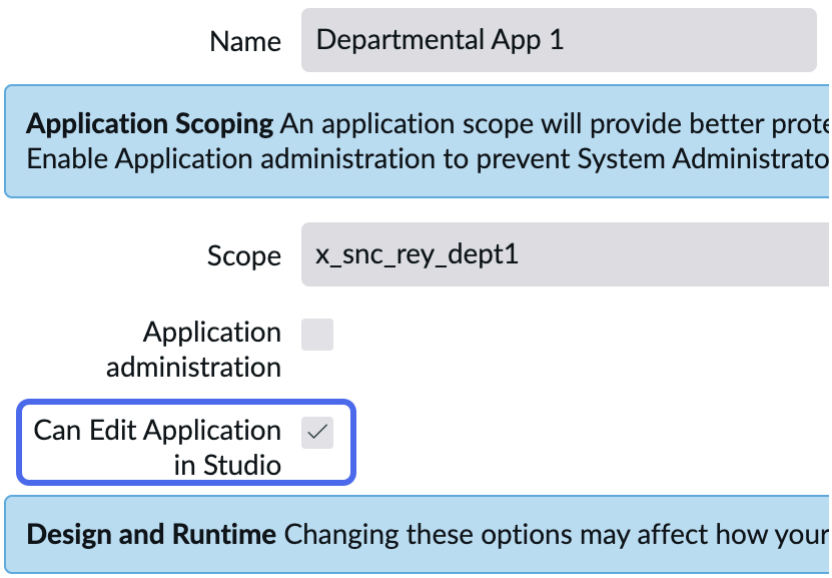
ServiceNow Studio provides a unified environment where you can find, organize, and edit your work using a consistent navigation structure and metadata taxonomy. Every development tool you need is accessible in one place, so you can switch between tools without leaving ServiceNow Studio. When your app is ready to deploy, manage the deployment using update sets or pipelines. ServiceNow Studio also includes native support for ServiceNow Otto, providing access to generative AI capabilities and low-code development tools.

As of version 29.2.11, ServiceNow IDE capabilities have been integrated into ServiceNow Studio. For more information, see Building apps in source code in ServiceNow Studio.

ServiceNow Studio and changed functionality

In the record for each app created on the ServiceNow AI Platform, there is an option to allow the app to be edited in legacy Studio. However, this option does not apply to the new ServiceNow Studio. Any global, scoped, or custom app you have permission to edit can be edited in ServiceNow Studio.

The **Can Edit Application in Studio** option on each app record does not apply to the new ServiceNow Studio.



How do GAC roles work in ServiceNow Studio?

Some features in ServiceNow Studio become available only after you create a scoped application. If you are new to ServiceNow Studio, the Search bar and **Create** > **App/File** options may not be visible initially. This is expected behavior.

To get started, select **Create** and create your scoped app. On the App details page, select **Create** to create metadata files and associate them to your app.

After you create your application, return to the ServiceNow Studio home page and refresh the browser. The Search bar and **Create** > **App/File** options are now available for scoped application development. For more information about roles, see Assign a role to a user.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Configuring ServiceNow Studio

ServiceNow Studio is available by default on the ServiceNow AI Platform and requires no additional configuration. It uses the platform's existing development tools, so you can start building immediately.

Domain separation and ServiceNow Studio

Domain separation is not supported for ServiceNow Studio. Domain separation enables you to separate data, processes, and administrative tasks into logical groupings called domains. You can control several aspects of this separation, including which users can see and access data.

Support level: No support

- The domain field may exist on data tables but there is no business logic to manage the data.
- This level is not considered domain-separated.

For more information on support levels, see Application support for domain separation.

Installing ServiceNow Studio

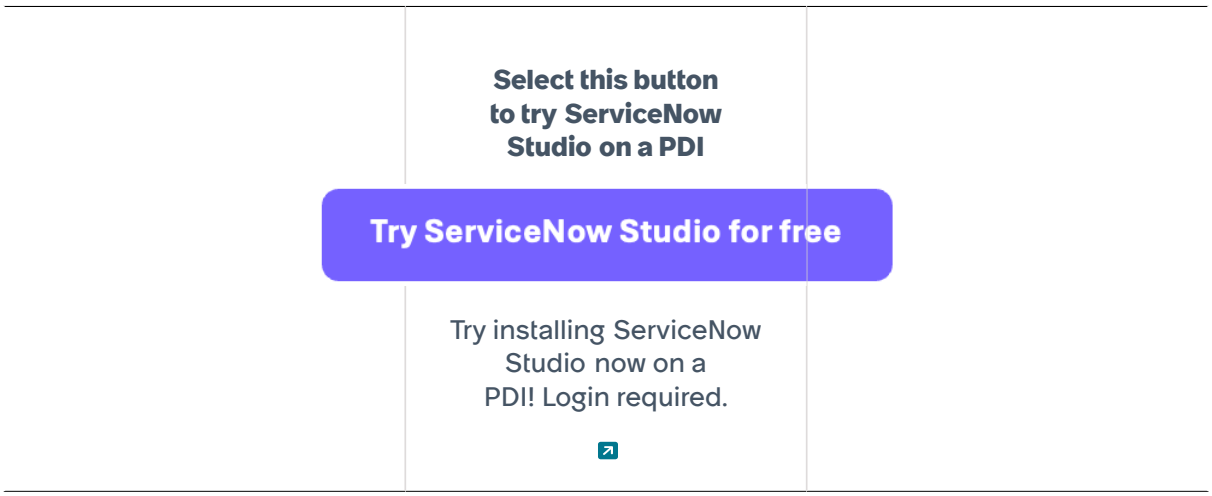
ServiceNow Studio is available for you to use on the ServiceNow AI Platform. You can also access premium features with an App Engine Enterprise license.

ServiceNow Studio is a ServiceNow AI Platform application that is available by default. If your company has an App Engine Enterprise license, you can install additional plug-ins to use in ServiceNow Studio.

No additional configurations must happen for you to use ServiceNow Studio.

Give ServiceNow Studio a try

Ready to give ServiceNow Studio a try? You can test it out using your own Personal Development Instance (PDI), which requires you signing in to the Developer Site. Find out more on PDIs in the Personal developer instance guide.



Premium features available with an App Engine Enterprise

If your company has an App Engine Enterprise license, you can access these additional premium features in ServiceNow Studio.

- Create a table from a spreadsheet. For more information, see Use a spreadsheet to add data.
- Create a table from a PDF. For more information, see Use a PDF to create data tables.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Access the UI Template library.
- Access App Engine Studio Catalog templates. For more information, see Create your app using an application template.
- Access Workspace Builder. For more information, see Workspace Builder.
- Access Workspace UI templates.
- Deploy scoped and global apps using the App Engine pipeline. For more information, see Managing deployments using Pipelines and Deployments.

ServiceNow Studio instance strategy

Install ServiceNow Studio on every ServiceNow instance where you develop applications, then define your company's access and deployment strategy.

Define your company's instance strategy for ServiceNow Studio by deciding how to manage access. Options include:

- Grant access to all users in your company.
- Grant access to a specific group of users.
- Grant access on a case-by-case basis using a form where users describe the app they want to build. An admin then decides whether to approve access for each request.

Use a non-production instance that is configured similarly to your production instance as your test environment. This configuration enables you to identify issues that may appear when the application is deployed to production.

Note: If you plan to clone your production instance to one or more non-production instances, install ServiceNow Studio on your production instance before cloning. For more information, see Instance Clone.

After you establish your instance strategy, complete the following tasks.

1. Establish and automate your approval or review process.
2. Select the non-production environment where ServiceNow Studio will run.
3. Determine how to promote apps from your non-production instance to your test instance, and then to production. Deploy apps using update sets, pipelines, or the Application Repository in ServiceNow Studio.

Components installed with ServiceNow Studio

Activating the ServiceNow Studio plugin installs tables, roles, and dependencies automatically.

What tables and roles does ServiceNow Studio install?

Tables installed with ServiceNow Studio

Table	Description
Resources [sn_sns_resources]	Populates the Resources section at the bottom of the home page.
Experience Configurations [sn_udc_experience_configuration]	Stores default role access configurations for the experience switcher. This table is not configurable.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Tables installed with ServiceNow Studio (continued)

Table	Description
Experience Visibility Controls [sn_udc_experience_visibility_control]	Enables you to grant non-default roles access to the experience switcher. This table is configurable.

ServiceNow Studio does not include specific roles. The Admin and Delegated Developer roles are used. For more information, see ServiceNow Studio personas and roles.

What plugin does ServiceNow Studio install?

The ServiceNow Studio [sn_sns] plugin installs with several dependencies.

ServiceNow Studio plugin

Plugin [ID]	Description	Dependencies
ServiceNow Studio [sn_sns]	ServiceNow Studio provides a unified environment for all ServiceNow development activities.	• sn_udc • sn_studio_commons • sn_app_obj_wizards • sn_deploy_pipeline

Managing access to ServiceNow Studio

Manage access to ServiceNow Studio by assigning delegated development permissions. Admins, delegated developers, and users with Guided Application Creator (GAC) roles can develop apps and app files in ServiceNow Studio.

Delegated development enables designated users without a system admin role to develop or deploy applications on the ServiceNow AI Platform. If you have the application-specific admin role or the system-level admin role, you can delegate application development to designated developers at the application level.

Note: Only admins and GAC users can create apps in ServiceNow Studio. Delegated developers can work on existing apps and app files.

If a user has the Guided Application Creator role (sn_g_app_creator.app_creator), they can access ServiceNow Studio and create apps. After creating an application, they are automatically assigned the delegated_developer role. For more information about delegated developer permissions, see ServiceNow Studio personas and roles.

For more information about delegated development, see Delegated development and deployment. Admins can also assign collaboration permissions to their development team. For more information, see Collaborating on apps using ServiceNow Studio.

Roles and access in app development tools

Access to ServiceNow Studio, Creator Studio, and ServiceNow IDE is controlled by the roles and permissions your admin assigns. Update role assignments when you need to change which tools a user can access.

Each development tool available through the experience switcher has specific role requirements. To access all development tools, you need the roles required for each tool individually. For example, the delegated_developer role provides access to ServiceNow Studio, but without

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

the admin and sn_creatorstudio.user or sn_creatorstudio.restricted_user roles, the experience switcher is not visible because you do not have access to the other tools.

Note: If you expect to see the experience switcher but do not see it, contact your admin. Custom configurations may affect which experiences are visible.

Development experiences and roles

Experience	Available for	NOT available for
Pro-coding experience in ServiceNow IDE	admin	sn_creatorstudio.restricted_user, sn_creatorstudio.user
Platform developer coding experience in ServiceNow Studio	delegated_developer, admin, sn_g_app_creator.app_creator	sn_creatorstudio.restricted_user, sn_creatorstudio.user
No-code experience in Creator Studio	delegated_developer, admin, sn_creatorstudio.user, sn_creatorstudio.restricted_user	Not applicable.

Important: Regardless of your role — including admin and delegated_developer — if you have a Creator Studio role, you cannot access ServiceNow Studio or ServiceNow IDE.

For more information, see Creator Studio roles and personas, ServiceNow Studio personas and roles, and ServiceNow IDE roles.

What are some role assignment examples?

As an admin, you can assign permissions to use ServiceNow Studio, Creator Studio, and ServiceNow IDE. Some roles provide access to more than one tool:

- To grant access to ServiceNow Studio and Creator Studio, assign the **delegated_developer** role. This role provides access to both tools but does not include access to ServiceNow IDE.
- To restrict users to Creator Studio only, assign the **sn_creatorstudio.user** or **sn_creatorstudio.restricted_user** role. These roles provide access to Creator Studio but not ServiceNow Studio.
- To grant access to ServiceNow IDE, the user must have admin permissions or other roles specific to ServiceNow IDE. Users with the **delegated_developer** role cannot access ServiceNow IDE. For more information, see ServiceNow IDE roles.

ServiceNow Studio personas and roles

Admins, delegated developers, and users with Guided Application Creator roles can access and develop in ServiceNow Studio. Each role has different capabilities and permissions.

What can each role do in ServiceNow Studio?

Admins and delegated developers have different capabilities in ServiceNow Studio. Admins can delegate users to work on specific apps and app files by assigning delegated_developer permissions.

For more information about roles and access to different app development tools, see Roles and access in app development tools.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Roles and capabilities

Role	Permissions	Capabilities
Admin	Admins can review and approve tasks related to custom application development.	• Access ServiceNow Studio. • Create apps in ServiceNow Studio. • Create Fluent apps in source code from the Explorer tab. • Edit existing apps and metadata records, or delegate development to another user. • Create, edit, and manage update sets. • Update existing metadata records. • Create and edit global (unscoped) metadata records.
Delegated developer (delegated_developer)	Delegated developers have access only to the apps they are assigned to. Permissions may vary based on your configuration.	• Access ServiceNow Studio. • Update apps they have delegated development permissions for. • Create an update set to package changes. • Create and update app files and other metadata records. • Submit an app for deployment through update sets, pipelines, or the Application Repository.
Delegated admin (Delegated developer with delegated_admin permission)	Delegated admins only have access to the scopes they have delegated developer access to.	Grants access to all delegated development permissions for a specific scoped application so that permissions don't need to be granted individually. For more information, see Developer and deployment permissions.
Guided Application Creator (GAC) roles (sn_g_app_creator.app_creator, sn_g_app_creator.global)	Users with GAC roles can access ServiceNow Studio and create apps. After creating an application, the user receives the	• Access ServiceNow Studio. • Create custom applications. • With the GAC Global role, create custom global apps.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Roles and capabilities (continued)

Role	Permissions	Capabilities
	delegated_developer role for that application.	
ServiceNow Studio admin / Unified Developer Core (UDC) admin (sn_udc.admin)	Provides administrative access to tables within ServiceNow Studio scope and configuration access to the experience switcher roles.	Granular admin role that enables the following actions: • Manage all ServiceNow Studio system properties, settings, and configurations. • Manage roles and permissions specific to ServiceNow Studio. • Configure the experience switcher roles across Creator Studio, ServiceNow Studio, and ServiceNow IDE. For more information about granular admin roles, see Granular admin roles.
Preferred tables admin (sn_prfrd_tables.admin)	Provides administrative access to configure preferred tables for Table Builder.	Granular admin role that enables you to configure preferred tables in your instance for Table Builder. **Note:** Table Builder does not require an additional admin role.
Metadata-specific admin roles • Business rule admin • Script include admin	Provide access to ServiceNow Studio.	Metadata-specific admin roles are existing roles on the ServiceNow AI Platform that provide developers with admin access to a specific metadata type. For example, the script include admin role enables a developer to create, read, update, and delete (CRUD) any script include on the ServiceNow AI Platform and perform development on script includes in ServiceNow Studio. Each metadata-specific admin role must:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Roles and capabilities (continued)

Role	Permissions	Capabilities
		• Enable CRUD access to the metadata types it represents directly within ServiceNow Studio. • Be configurable within the experience switcher.

Elevate your role in ServiceNow Studio

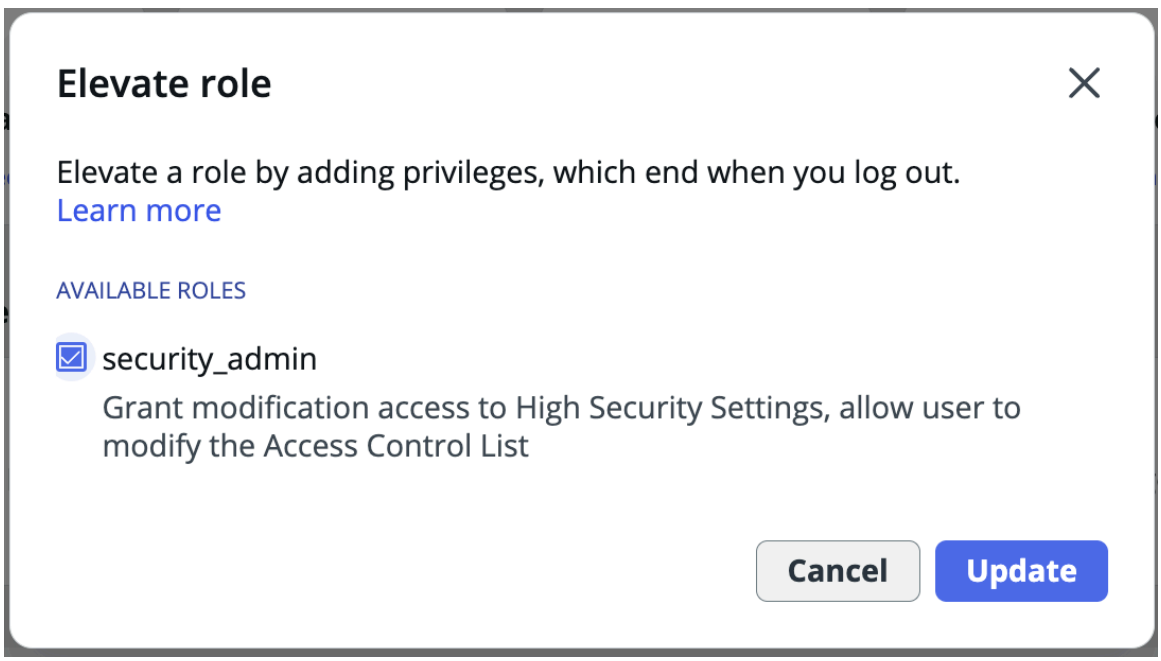
Elevate your role to security_admin in ServiceNow Studio to configure roles and access control lists (ACLs).

Before you begin

Role required: admin, or another role that can be elevated.

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Select the **User Profile** icon in the activity bar.
3. Select **Elevate role**.
4. Select the **security_admin** role.
   The security_admin role grants modification access to high security settings and enables you to modify the access control list (ACL). For more information, see Elevated privilege roles.



5. Select **Update**.

Managing access to the experience switcher

The experience switcher can provide access to Creator Studio, ServiceNow Studio, ServiceNow IDE. However, whether you can see and select all of those depends on your role or access level.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

There are several factors that determine what you can choose from the experience switcher: your role, whether the selected studio is installed on the instance, and which version is installed.

Note: As of version 30.1.1, the experience switcher is no longer available in ServiceNow Studio.

How do roles control access to the experience switcher?

Your role — including user group and security level — determines which products are accessible in the experience switcher.

The following describes default role access to the experience switcher:

- Admins and delegated developers can use the experience switcher because they may need access to any product where they've been delegated to administer or develop an app.
- Creator Studio users and Creator Studio restricted users don't generally have access to the experience switcher because administrators limit them to a more curated experience.

Admins can check the Experience Configurations table [sn_udc_experience_configuration] to see which default roles have access to each product in the experience switcher. The table also shows which roles see the discovery page that appears if Creator Studio is not installed or is below the minimum version required for the experience switcher.

Important: Regardless of your role — including admin and delegated developer roles — if you have a Creator Studio role, you cannot access ServiceNow Studio or ServiceNow IDE.

To grant non-default roles access to products in the experience switcher, update the Experience Visibility Controls table [sn_udc_experience_visibility_control]. For more information, see Configure non-default access to the experience switcher.

Product installation determines access

You must have an application installed on the instance to select it from the experience switcher.

For example, if you're working in ServiceNow Studio or ServiceNow IDE but don't have Creator Studio installed, selecting Creator Studio in the experience switcher displays a page where you can find out more on using Creator Studio.

Product version determines access

To appear in the experience switcher, all products must be on the Yokohama version at a minimum.

For example, if you choose Creator Studio in the experience switcher but have the Xanadu version of Creator Studio installed, selecting it displays a page directing you to update the version.

Configure non-default access to the experience switcher

Grant non-default roles access to the experience switcher so users can switch between development environments in ServiceNow Studio.

Before you begin

Role required: admin

About this task

By default, the Experience Configurations table [sn_udc_experience_configuration] grants admins and delegated developers access to the experience switcher. To grant non-default roles access, add them to the Experience Visibility Controls table

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

[sn_udc_experience_visibility_control]. For more information about default roles, see Roles and access in app development tools.

- Admins and delegated developers can use the experience switcher because they may need access to any product where they've been delegated to administer or develop an app.
- Creator Studio users and Creator Studio restricted users don't generally have access to the experience switcher because administrators limit them to a more curated experience.

Procedure

1. Navigate to **All** > **sn_udc_experience_visibility_control.list**.
2. Select **New**.
3. Enter the experience you are configuring roles for.
   For example, enter `Serv` to select ServiceNow Studio.
4. Specify the roles that should get access to the experience.
   a. Unlock the **Available for roles** field by selecting the lock icon  .
   b. Start typing and then select the name of the role you're giving access to.
   c. Lock the **Available for roles** field by selecting the unlock icon  .
5. Specify the roles that should not have access to the experience.
   a. Unlock the **Not available for roles** field by selecting the lock icon  .
   b. Start typing and then select the name of the role you're restricting access for.
   c. Lock the **Not available for roles** field by selecting the unlock icon  .
6. Select **Submit**.
   The role is added to the Experience Visibility Controls table and the specified users can now access the experience switcher.

Collaborating on apps using ServiceNow Studio

Collaborate with other developers on app development in ServiceNow Studio by inviting them to co-create and develop apps with you.

What is collaboration in ServiceNow Studio?

Collaboration, also referred to as delegated development, builds on the existing delegated development feature set in the ServiceNow AI Platform. It enables developers to invite other developers into apps to co-create and develop together. Depending on your permissions, you can invite others to collaborate on an app or request to join another developer's app. For more information, see Delegated development and deployment.

There are two standard collaborator roles when co-developing an app: owner and editor. Admins can create a custom collaboration role by adjusting permissions.

What are the requirements for collaboration?

ServiceNow Studio supports the collaboration plugin for properly licensed customers.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note:

1. An App Engine Enterprise license is required to use collaboration fully.
2. If you already have the collaboration plugin installed, you can continue to use collaboration.
3. Customers that don't have Collaboration installed will not be able to manage delegated development permissions in ServiceNow Studio. Existing delegated development permissions will still be respected within ServiceNow Studio.

Which apps can you access in ServiceNow Studio?

The admin and delegated_developer roles have different levels of app access in ServiceNow Studio:

- Users with the admin role have access to all apps in ServiceNow Studio.
- Users with the delegated_developer role have access to:
  - Apps they create.
  - Apps they have been invited to edit.
  - All apps within the scope they have access to. For more information about scopes, see Application scope.

To request access to an app that is not visible in ServiceNow Studio, contact your admin and ask them to grant you permission using the Collaboration app. The Collaboration app is automatically installed with ServiceNow Studio. For more information about the Collaboration app, see Application collaboration.

What can owners and editors do?

There are two default collaborator roles for working on apps with other developers: owner and editor.

- If you create an app, you are the owner of that app.
- If an app is visible in ServiceNow Studio and you have been delegated to work on it, open it and begin working with the collaboration role the owner assigned you. That role is usually editor.

Default collaboration descriptors

Descriptor	Description
Owner	Owner of the application. • If you create the app, you are automatically the owner. • Owners can manage other collaborators for the app. • Owners can delete apps because they have the delete app permission. • Owners automatically receive the delegated_developer role for the app.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Default collaboration descriptors (continued)

Descriptor	Description
Editor	• Editors can invite collaborators. • Editors have more limited editing permissions than owners.

Note: For the full list of default owner and editor permissions, see Collaboration permissions for ServiceNow Studio.

What are custom collaboration descriptors?

The collaboration descriptor assigned to a developer determines what they can do in the app — including assigning, managing, and monitoring delegated development permissions. Owners have more permissions than editors by default.

To create a customized collaboration role, create a custom collaboration descriptor and use collaboration permissions to control what developers can do in the app.

Admins can define custom collaboration descriptors to select when managing collaborators using the Collaboration app. For more information about custom descriptors, see Create collaboration descriptors to assign permissions.

How does the collaboration approval process work?

If you invite someone to collaborate on an app and they do not have the delegated_developer role, an App Engine admin must approve the collaboration request. For more information, see Delegated development and deployment.

When you add a user or group to collaborate on an app, the platform generates a collaboration task that initiates an approval flow. The task includes details about which app the developer is being added to and what permissions they receive. If App Engine Management Center (AEMC) is installed, admins can review and approve or deny collaboration requests there.

If AEMC is not installed, admins can navigate to **All** > **App Engine** > **Collaboration** > **Collaboration Tasks** to review collaboration tasks.

View collaborators on an app in ServiceNow Studio

View the collaborators on an app in ServiceNow Studio to see who has access and what role each person has.

Before you begin

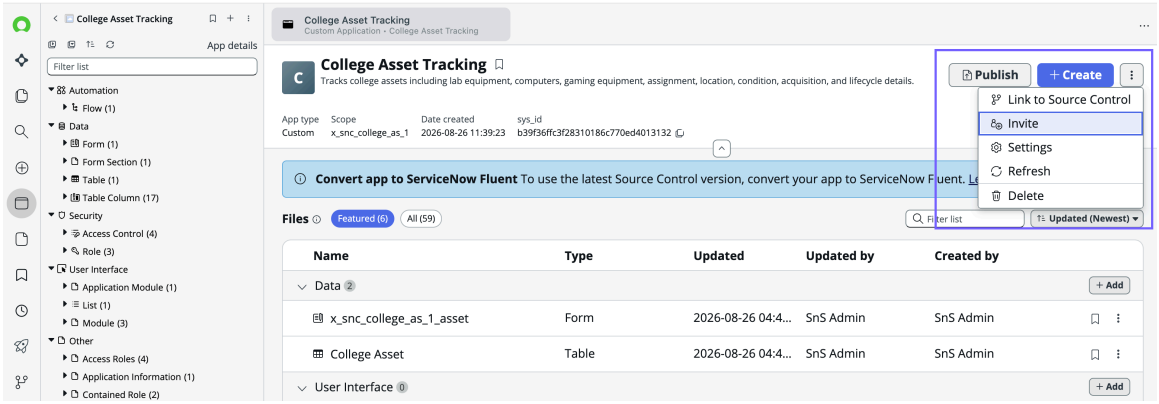
Customers that don't have Collaboration installed will not be able to manage delegated development permissions in ServiceNow Studio. Existing delegated development permissions will still be respected within ServiceNow Studio.

Role required: admin or delegated_developer

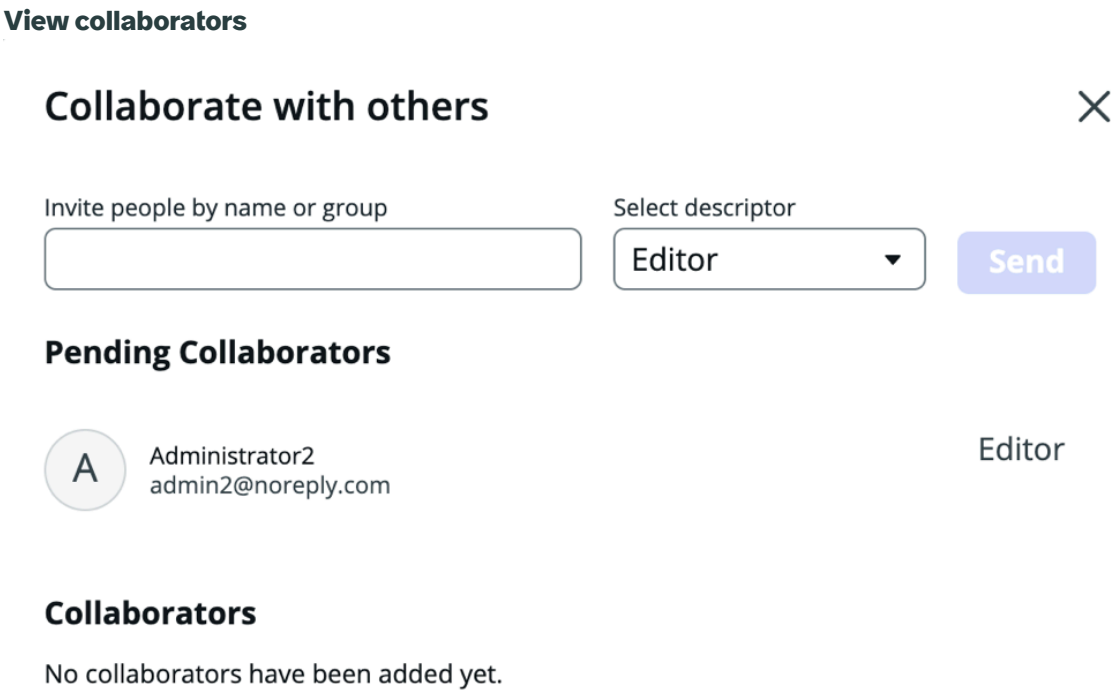
Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Select the app you want to view collaborators for, then select **App details**.
3. Access collaboration settings by selecting the more options icon ⋮ and selecting **Invite**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



4. Review the collaborator names, groups, and descriptors on the **Collaborate with others** dialog.



The **Collaborate with others** dialog displays a list of all current collaborators, their groups, and their assigned descriptors.

Result

From the collaborators view, you can take additional actions. For more information, see the following topics:

- Add collaborators to an app in ServiceNow Studio
- Modify or customize collaboration permissions for a user or group in ServiceNow Studio
- Remove collaborators from an app in ServiceNow Studio

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add collaborators to an app in ServiceNow Studio

Add collaborators to an app in ServiceNow Studio so other developers can co-develop the app with you.

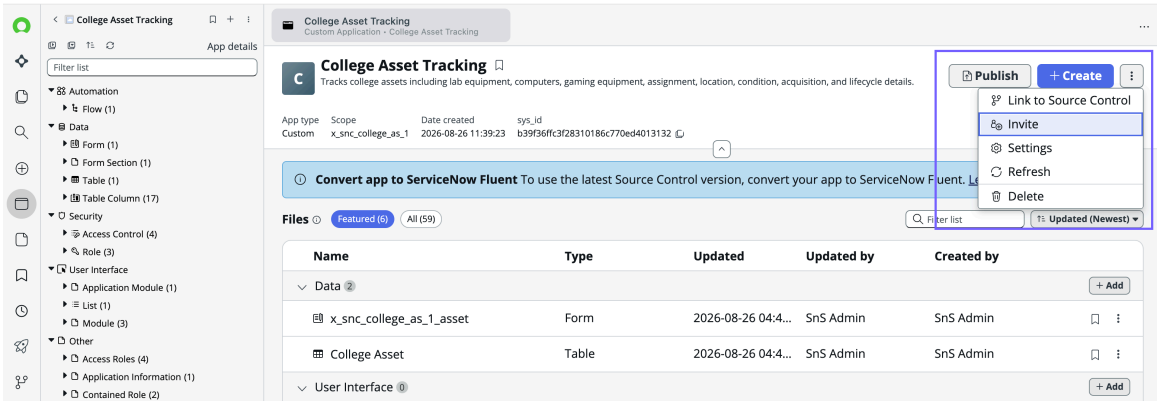
Before you begin

Customers that don't have Collaboration installed will not be able to manage delegated development permissions in ServiceNow Studio. Existing delegated development permissions will still be respected within ServiceNow Studio.

Role required: admin or delegated_developer

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Select the app you want to add collaborators to, then select **App details**.
3. Access collaboration settings by selecting the more options icon ⋮ and selecting **Invite**.



4. In the **Invite people by name or group** field, enter the name of the user or group you want to invite.
   A list of matching users and groups appears. If a user or group appears in the list but is not selectable, they have already been added as a collaborator.
5. In the **Select descriptor** field, select the collaboration role for the user or group.
   If you are an editor for the app, you can select only the editor descriptor.
   - For more information about collaboration descriptors, see Collaborating on apps using ServiceNow Studio.
   - For a list of all collaboration permissions, see Collaboration permissions for ServiceNow Studio.
6. Select **Send** to invite the collaborator.
   - If the user is new to the ServiceNow AI Platform, an admin must approve the request. After approval, both the requester and the user receive an email confirming that the user has been added to the application.
   - If the user is not new to the ServiceNow AI Platform, the collaboration request is auto-approved. Both the requester and the user receive an email confirming that the user has been added to the application.

Modify or customize collaboration permissions for a user or group in ServiceNow Studio

Modify or customize the collaboration permissions for a user or group in ServiceNow Studio to control what they can do in an app.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

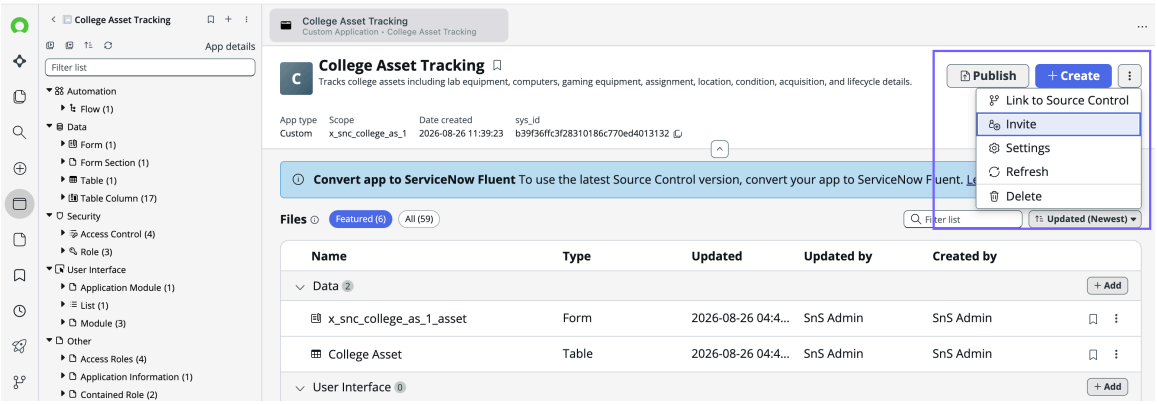
Before you begin

Customers that don't have Collaboration installed will not be able to manage delegated development permissions in ServiceNow Studio. Existing delegated development permissions will still be respected within ServiceNow Studio.

Role required: admin or delegated_developer

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Select the app you want to modify collaborators for, then select **App details**.
3. Access collaboration settings by selecting the more options icon ⋮ and selecting **Invite**.



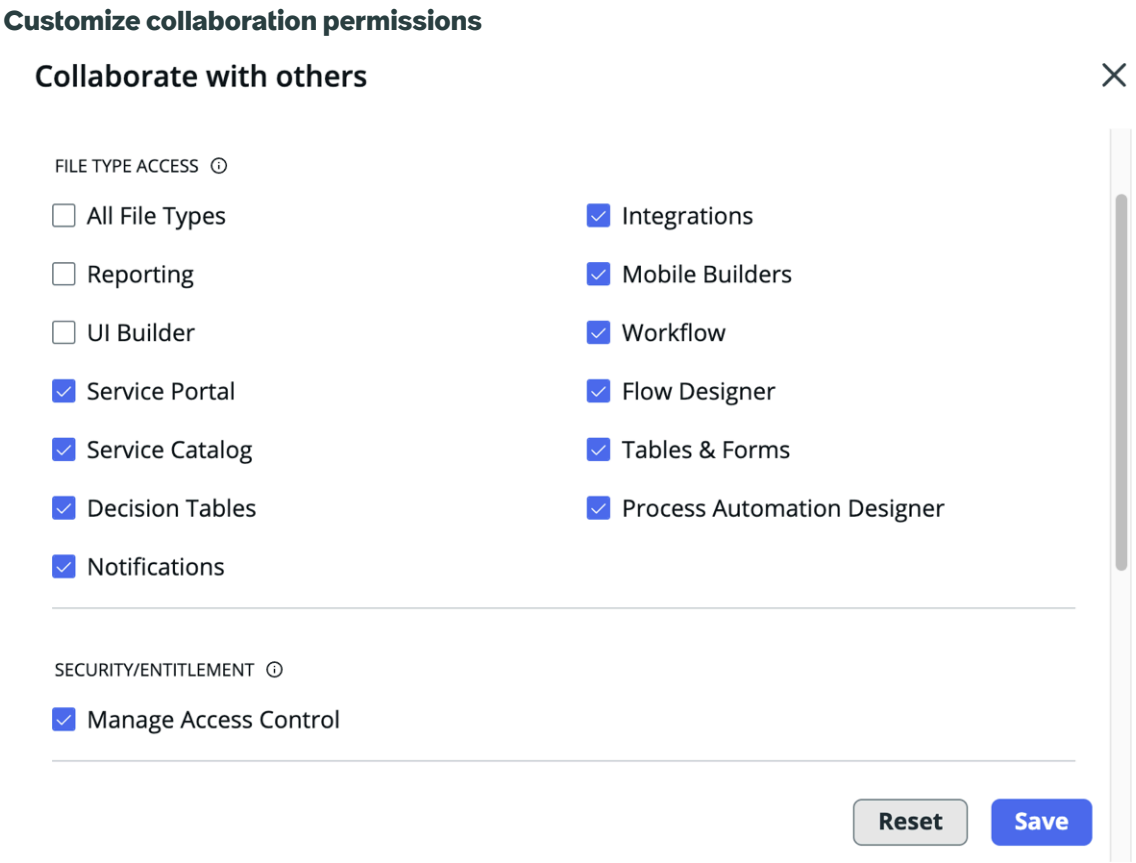
4. In the **Collaborators** section of the **Collaborate with others** dialog, select the new permission level for the user or group.
5. **Optional:** To create custom permissions for the collaborator, select **Customize permissions** for the user or group in the **Collaborators** section.

   a. Select **Customize permissions** for the user or group.



   b. Select the permissions you want the user or group to have.
   For more information about permissions, see Collaboration permissions for ServiceNow Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



c. Select **Save**.

Result
Changes are saved automatically when you close the **Collaborate with others** dialog.

What to do next
Your App Engine admin must approve the changes to collaborators. For more information, see Approve a collaboration request.

Remove collaborators from an app in ServiceNow Studio

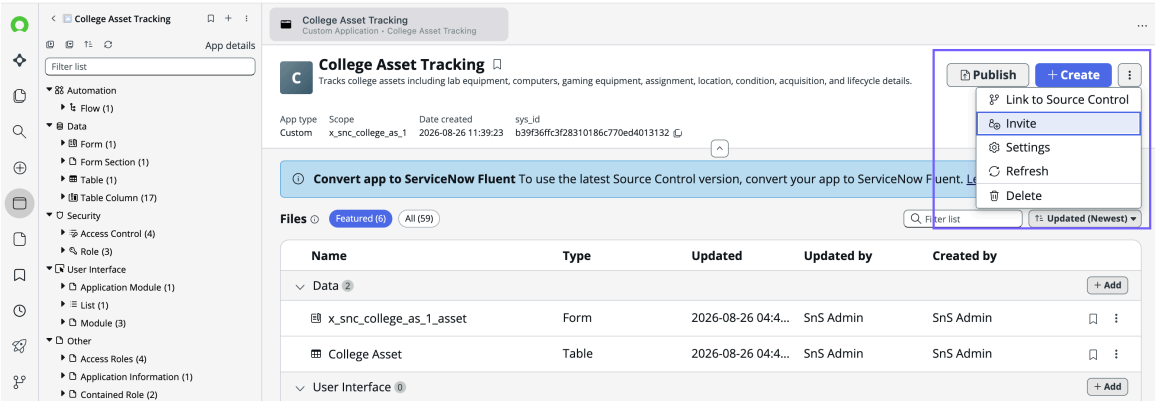
Remove a user or group as a collaborator in ServiceNow Studio to revoke their access to an app.

Before you begin
Customers that don't have Collaboration installed will not be able to manage delegated development permissions in ServiceNow Studio. Existing delegated development permissions will still be respected within ServiceNow Studio.

Role required: admin or delegated_developer

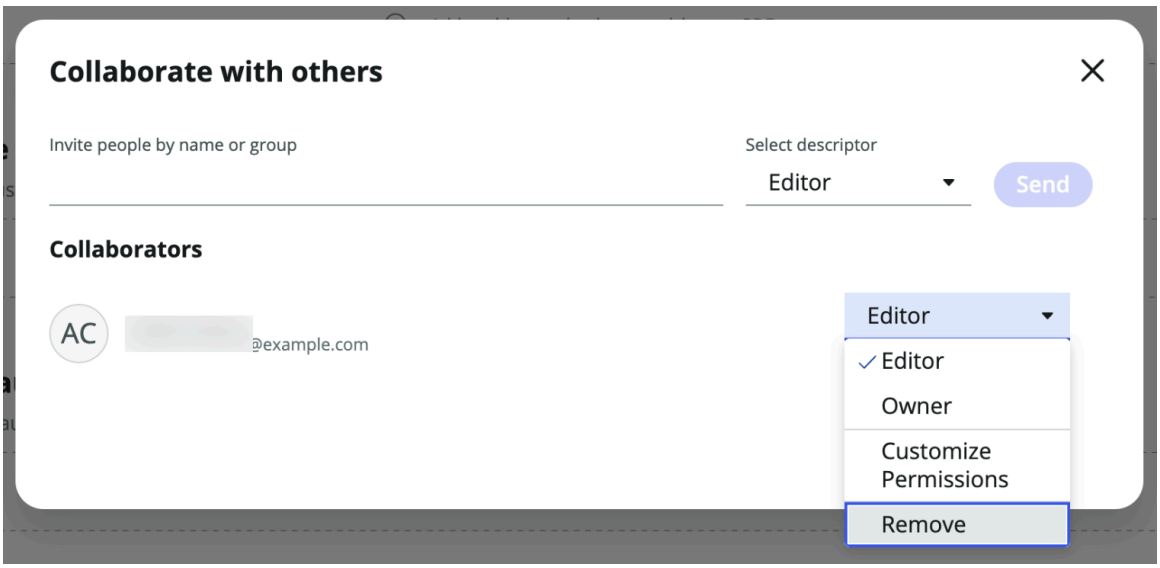
Procedure
1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Select the app you want to remove collaborators from, then select **App details**.
3. Access collaboration settings by selecting the more options icon ⋮ and selecting **Invite**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



4. In the **Collaborators** section of the **Collaborate with others** dialog, select **Remove** for the user or group.

Remove a collaborator



The collaborator is removed and no longer has access to the app.

Result

Changes are saved automatically when you close the **Collaborate with others** dialog.

ServiceNow Studio settings

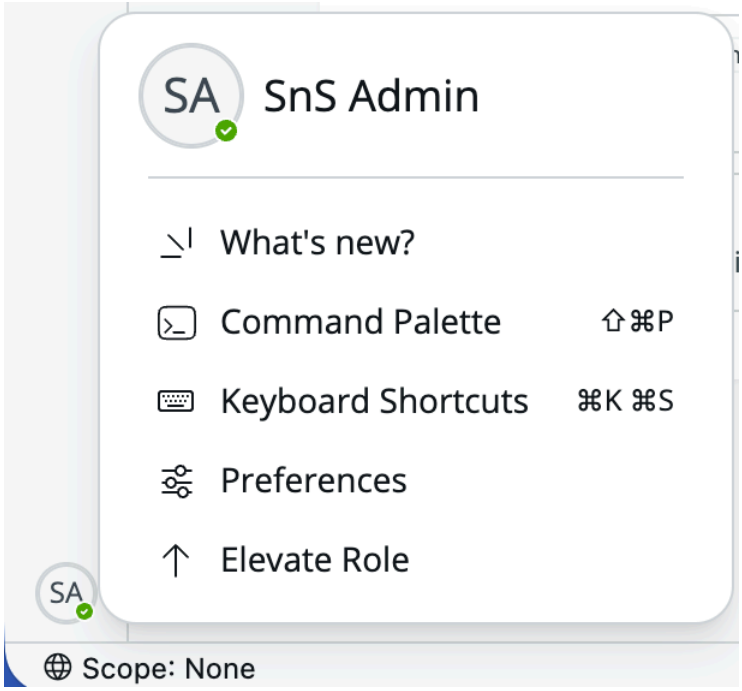
Personalize settings and preferences in ServiceNow Studio to get the most out of the application development experience.

The User profile menu in ServiceNow Studio provides access to the Command palette, a list of keyboard shortcuts, preferences for theme and development style, and to elevate your role.

Select the **Settings** icon in the activity bar to open this menu.

For more information about updating your preferences, see Personalizing the ServiceNow Studio homepage.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

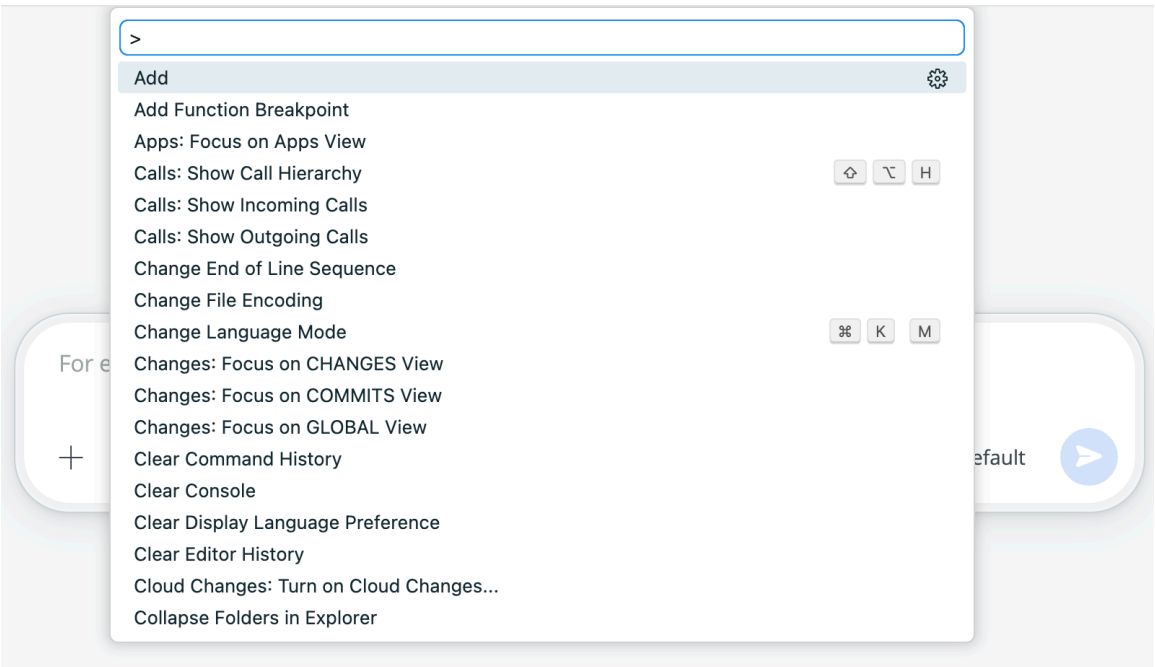


Command palette

Open the command palette from the Settings menu or by using the following keyboard shortcuts:

- Windows: Ctrl-Shift-P
- Mac: Cmd-Shift-P

Start typing the command you want, or scroll to find and select it. Select **Add** to create a keyboard shortcut.

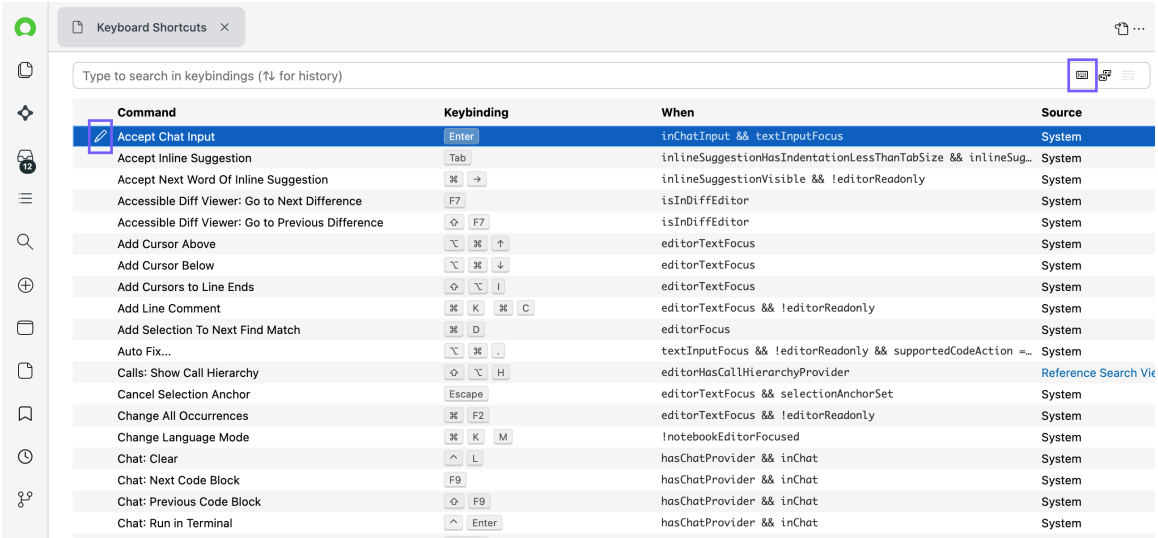


Keyboard shortcuts

Keyboard shortcuts enable you to work more quickly and efficiently in ServiceNow Studio. Select **Keyboard Shortcuts** on the Settings menu to open the list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Search or scroll through the list to see available shortcuts. Update existing shortcuts with new key commands by using the change keybinding icon . Record new keyboard shortcuts by using the record keys icon .



Elevate your role

Elevate your role to security_admin to configure roles and access control lists (ACLs). For more information, see Elevate your role in ServiceNow Studio.

Link an app to source control in ServiceNow Studio

Link an application or application customization to a Git repository in ServiceNow Studio so application developers can manage changes directly from the platform.

Before you begin

- Review Manage customizations to applications before linking a customization.
- Create a dedicated Git repository for the application. For increased security, enable multi-factor authentication for the Git repository.
- Generate an access token for the source control integration to use instead of a password and multi-factor authentication passkey when creating a Credential record. Search for personal access token on GitHub or GitLab.
- Restrict permissions on the access token to allow read and write access to the Git repository.
- Verify that the non-production instance has network access to the Git repository.
- Ensure that each user adds the email address they use in their Git commits to their Users table [sys_user] record.
- Role required: admin

About this task

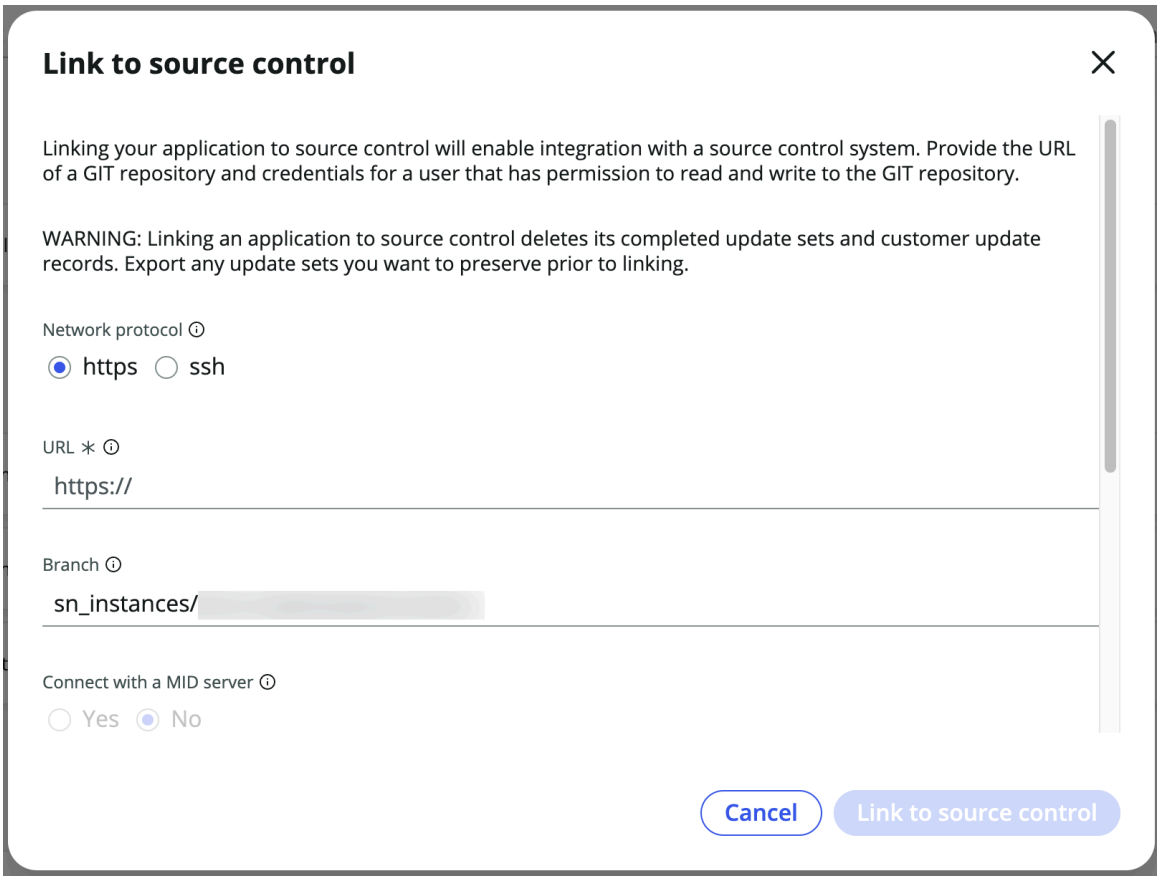
You cannot link an application or customization to source control on a production instance. To install applications on a production instance, use the application repository, an update set, or ServiceNow Studio.

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. In the file navigator, select the application you want to open.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

3. Select **App details** to open the app in the canvas.
4. Select **Source control** > **Link to source control**.



5. Enter the connection details for the Git repository.

Source control connection details

Field	Description
Network protocol	HTTPS or SSH credential type that enables secure channel data exchange.
URL	The URL of the Git repository where you want to save application files. For SSH protocol, use the following command to generate a private key: `ssh-keygen -t rsa -m PEM -b 4096 -C "email@address"`. **Note:** If the SSH URL provided by your Git server does not work, contact your Git server owner or provider for the correct URL. Additional specifications such as scheme protocol prefixes and port numbers may be required.
Branch	The repository branch to use for commits. The default branch is set to "main" if it is not already set in the remote repository. If there is no default branch in the remote Git repository, the instance creates a new default branch named "main". Configure this using the *`glide.source_control.git_default_branch`* system property.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
MID Server Name	The name of the existing MID Server to link through. Use a separate MID Server to prevent conflicts with Discovery activities. Verify that the MID Server user can create files in the [sys_attachment] table and that the table accepts files of the "bundle" type. Connecting through a MID Server enables access to repositories behind a firewall. For more information, see MID Server.
Default email	The committer email address is taken from the sys_user record when available. If a committer's sys_user record email field is empty, the ServiceNow AI Platform generates an alternate email address (`username@instancename.service-now.com`). Enter a default email address to use when no sys_user email is available. To use the default email address in all cases, select the check box.
Credential	The credential to use with the selected protocol. For more information about creating credentials, see Get started with credentials. **Note:** For SSH protocol, enter a valid credential of the SSH private key type. For HTTPS protocol, enter a valid credential of the Basic Auth credentials type.
Commit Comment	An optional description of the repository or application.

Note: All application developers on the instance share a single set of repository credentials.

6. Select **Link to source control**.
   The ServiceNow AI Platform validates the connection and user credentials and displays a success message. All application developers on the instance can now use the linked Git repository to manage changes.

Related topics

MID Server

Getting started with credentials

Import an app from source control in ServiceNow Studio

Import an application from a Git repository into ServiceNow Studio to create a local copy of the app on your non-production instance. The repository must contain a valid ServiceNow application, and your credentials must have read access.

Before you begin

- Verify that the non-production instance has network access to the Git repository.
- Verify that the repository contains a valid application.
- Ensure that each user adds the email address they use in their Git commits to their Users table [sys_user] record.
- Review Manage customizations to applications before importing a customization.
- Role required: admin

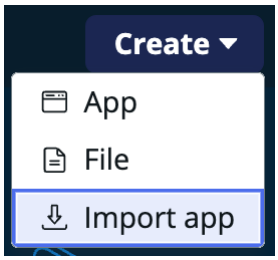
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

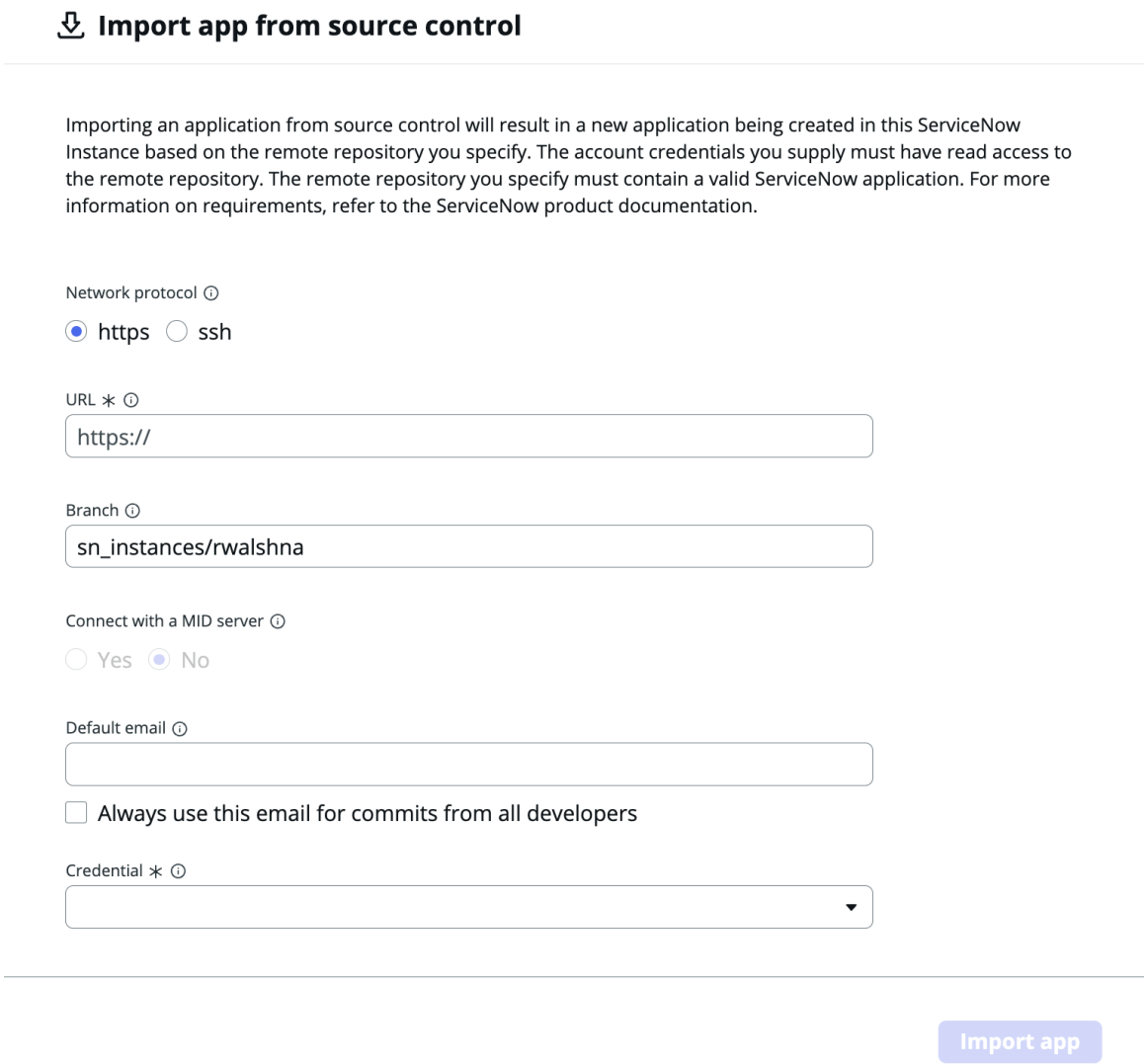
About this task

Source control integration does not support importing an application on a production instance. To install applications on a production instance, use the application repository, an update set, or ServiceNow Studio.

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Select the **Create** dropdown list on the home page, then select **Import app**.





3. Enter the connection details for the Git repository.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Import from source control fields

Field	Description
Network protocol	HTTPS or SSH credential type that enables secure channel data exchange.
URL	The URL of the Git repository where the application files reside. **Note:** If the SSH URL provided by your Git server does not work, contact your Git server owner or provider for the correct URL. Additional specifications such as scheme protocol prefixes and port numbers may be required.
Branch	The repository branch to use for the import. **Note:** The default branch is named after your instance. If you do not enter a name, the branch defaults to **master**.
Connect with a MID Server	Select whether to use an existing MID Server to connect to a Git repository behind a corporate firewall. **Note:** Use a separate MID Server to prevent conflicts with Discovery activities. For more information, see MID Server.
Default email	The committer email address, taken from the sys_user record when available. If a committer's sys_user record email field is empty, the system generates an alternate email address (username@instancename.service-now.com). Enter a default email address to use when no sys_user email is available. To use the default email address in all cases, select **Always use this email for commits from all developers**.
Credential	The credential to use for your Git repository. For more information, see Getting started with Credentials. **Note:** For SSH protocol, enter a valid credential of the **SSH Private Key** type. For HTTPS protocol, enter a valid credential of the **Basic Auth Credentials** type.

Note: All application developers on the instance share the credentials used to link a Git repository to an application.

4. Select **Import app**.
   The system compares the checksum in the `checksum.txt` file to the current checksum. When the checksum values match, the integration skips validation and imports the application. When the checksum values do not match, the integration validates and sanitizes the application files before importing them.
5. Select **Select Application**.
   ServiceNow Studio displays the imported application as a new option in the Switch Applications dialog.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

What to do next

- Review the upgrade logs for any sanitization applied to application files during the import.
- Select the imported application to open and edit it.

Using ServiceNow Studio

Use ServiceNow Studio to build and deploy apps using integrated ServiceNow AI Platform features, development tools, and builders — all in a single environment.

What does ServiceNow Studio help you do?

ServiceNow Studio combines the tools and features you need for app development in one place, so you can work faster across the full development lifecycle. Use ServiceNow Studio to:

- Access and switch between the development tools you need for each stage of your project.
- Find, organize, and edit apps and metadata records.
- Track, package, and deploy apps using update sets, pipelines, or the Application Repository.

Build Agent in ServiceNow Studio

Use Build Agent, an autonomous AI agent, to create and update applications in ServiceNow Studio through conversational interaction.

For full documentation on Build Agent, see Build Agent. To learn how to develop an application using Build Agent, see ServiceNow Studio and Build Agent tutorial.

How does the Build Agent workflow work?

Use Build Agent in ServiceNow Studio by following these steps.

1. The Build Agent chat interface opens automatically during a new ServiceNow Studio session.

   If it is not open, select the ServiceNow Otto icon or select **Open Build Agent** from the lower right corner status bar.
2. Prompt Build Agent to create or update an application.
3. Refine the application through conversational interaction, providing feedback and requesting changes.
4. Roll back changes at any point using checkpoints created by Build Agent.
5. Deploy the app through update sets, pipelines, or the Application Repository.
6. Reopen the conversation at any time to make further updates.

For more information, see Agentic ServiceNow AI Platform development with Build Agent.

What can Build Agent do in ServiceNow Studio?

Use Build Agent in ServiceNow Studio to complete the following tasks.

- Create an app or app file.
- Update an app or app file.
- Summarize the contents of an app.
- Track conversation updates with the conversation change log.
- Roll back conversations and changes to a checkpoint.
- Roll back an entire conversation.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Deploy apps created using Build Agent using update sets, pipelines, or the Application Repository — the same process as any other app.

For more information, see Update sets in ServiceNow Studio, Pipelines in ServiceNow Studio, and Application Repository in ServiceNow Studio.

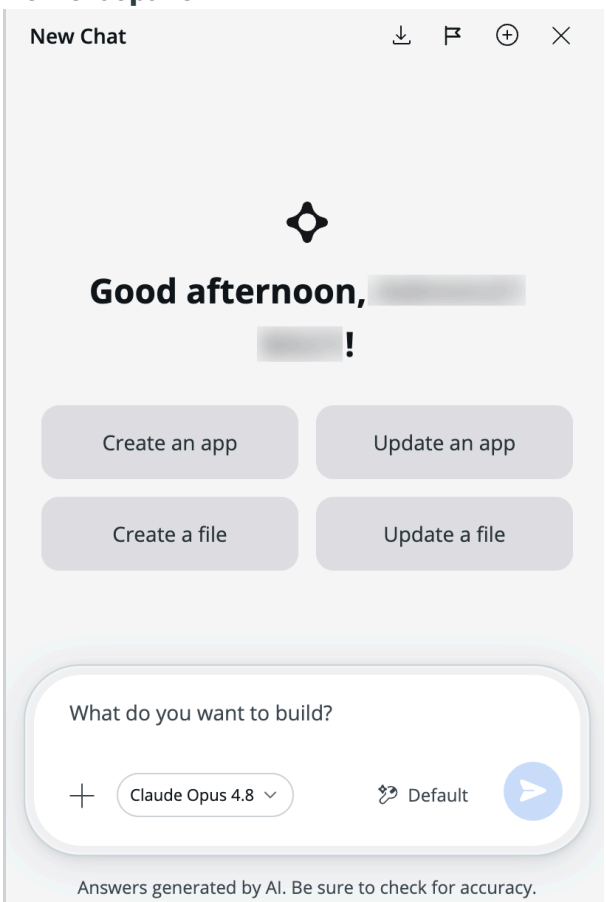
⚠ Warning:

Build Agent is an AI system that may produce inaccurate or incomplete results. Review all AI-generated code and configurations before deploying to production environments.

Build Agent chat panel

Use the Build Agent or Autonomous Engineer chat panel to create or update an app or app file. Make a selection to begin the chat, or enter a prompt. If you have a longer prompt, the chat box expands as you extend your input. The chat panel runs in Build Agent mode by default. To use Autonomous Engineer, select **Build Agent** from the selector in the chat panel and choose Autonomous Engineer.

New chat panel



Continue your conversation in the chat panel until you're happy with the results.

For more information, see the following topics:

- Create an application using Build Agent default mode
- Edit an existing application using Build Agent default mode
- Creating or updating an app file with Build Agent default mode
- Revert app changes with Build Agent

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Checkpoints and conversation change log
- Example prompts

Navigating the Build Agent chat panel

Function	Description
New chat icon	Open a new chat in the Build Agent chat panel. Begin a new chat when you want to start working on a new application or need a fresh start for updates.
Chats icon	See a list of all your chats with Build Agent.
Checkpoints icon	See a list of all the checkpoints within your current chat with Build Agent. Checkpoints show all the progress points in your application. You can revert to any of these checkpoints during the course of developing your app.

Applications in ServiceNow Studio

You can build, manage, and deploy custom, base system, or global applications and global metadata records in ServiceNow Studio — all from a single environment, based on your permissions.

What are ServiceNow applications?

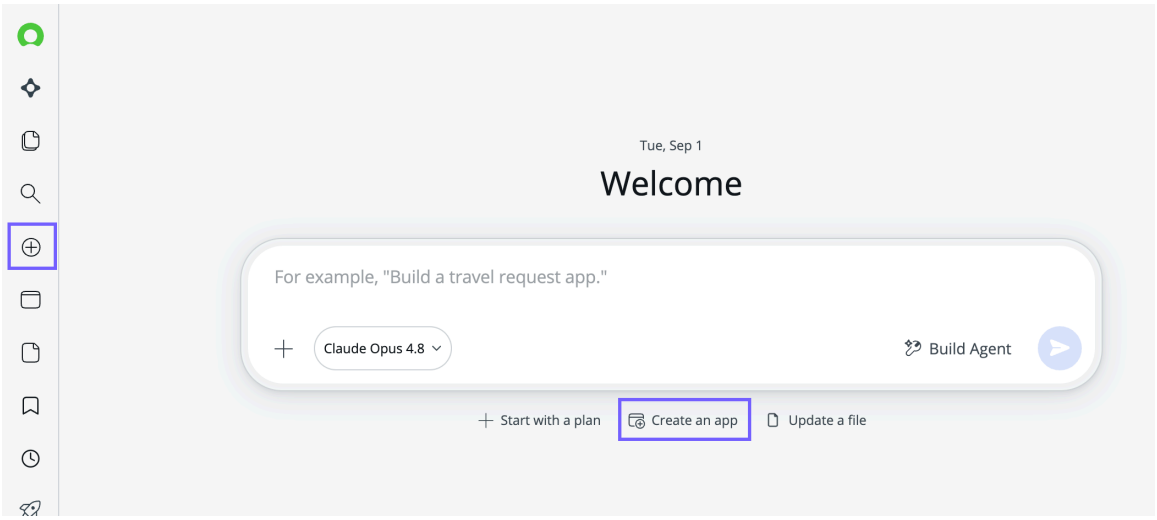
A ServiceNow app is a package that performs a specific task for a specified group of users. Think of an app as a container with a set of rules around who can access and edit it. For example, ServiceNow apps can include an API, a table, a workspace, a form, a flow, or any combination of those things.

Applications range from a few files to thousands of files. In ServiceNow Studio, you can create and work on apps of any size with a variety of file types, based on your permissions.

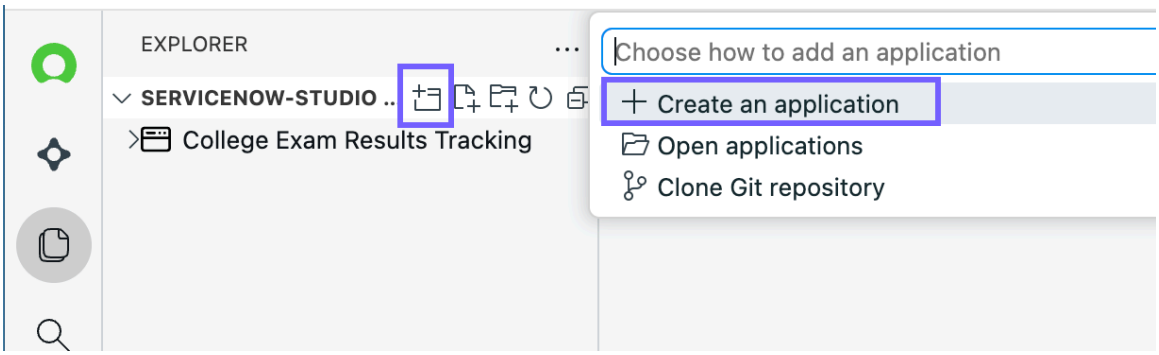
How do I create an application?

Use either **Create** button on the ServiceNow Studio home page to start creating an app with metadata records. For more information, see Create an application in ServiceNow Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



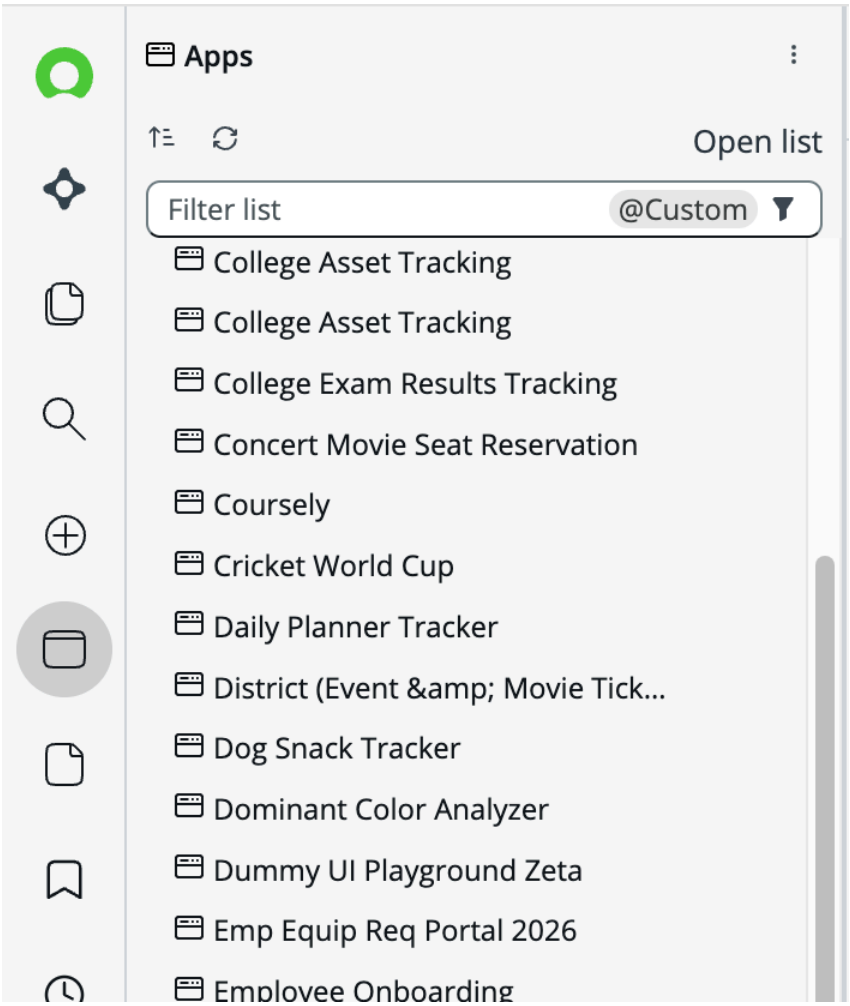
Create an app in source code by selecting the Explorer tab and selecting the add to workspace icon . Then select **Create an application**. For more information, see Building apps in source code in ServiceNow Studio.



Opening an existing application for the first time

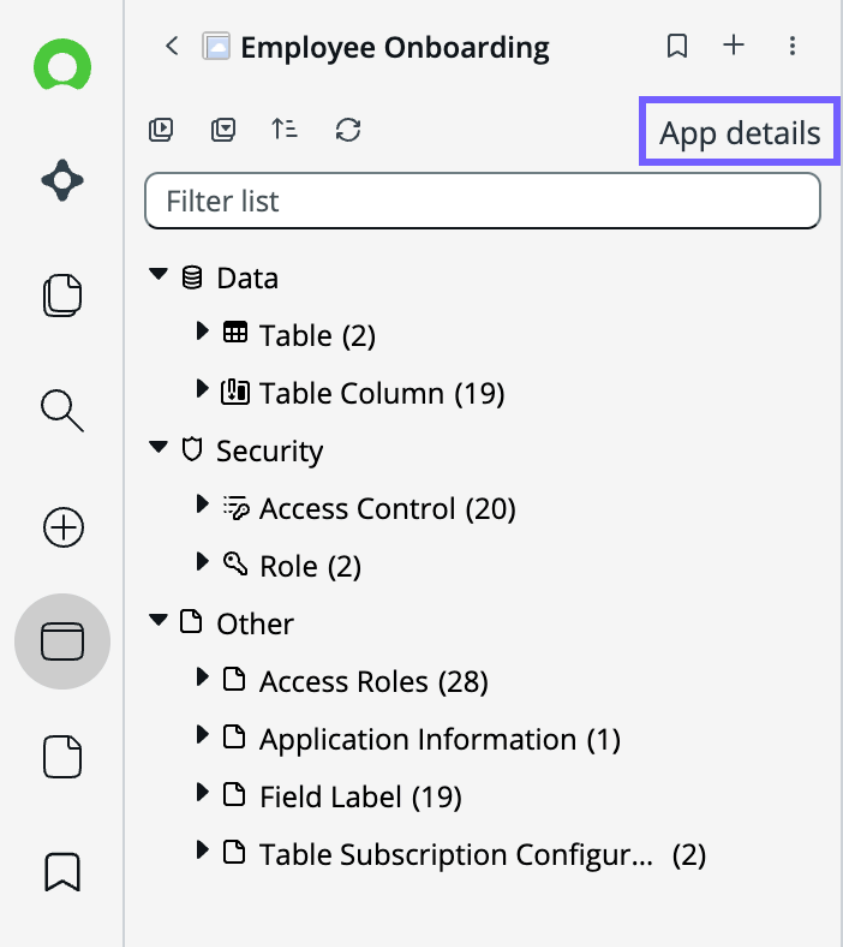
When you open ServiceNow Studio for the first time, open your apps from the Navigator panel on the left side of the screen. Search for the app by name or select it from the list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Selecting an app opens its full contents in the Navigator panel, with all file types organized by category, such as Automation. Select **App details** to open the app in a new tab.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



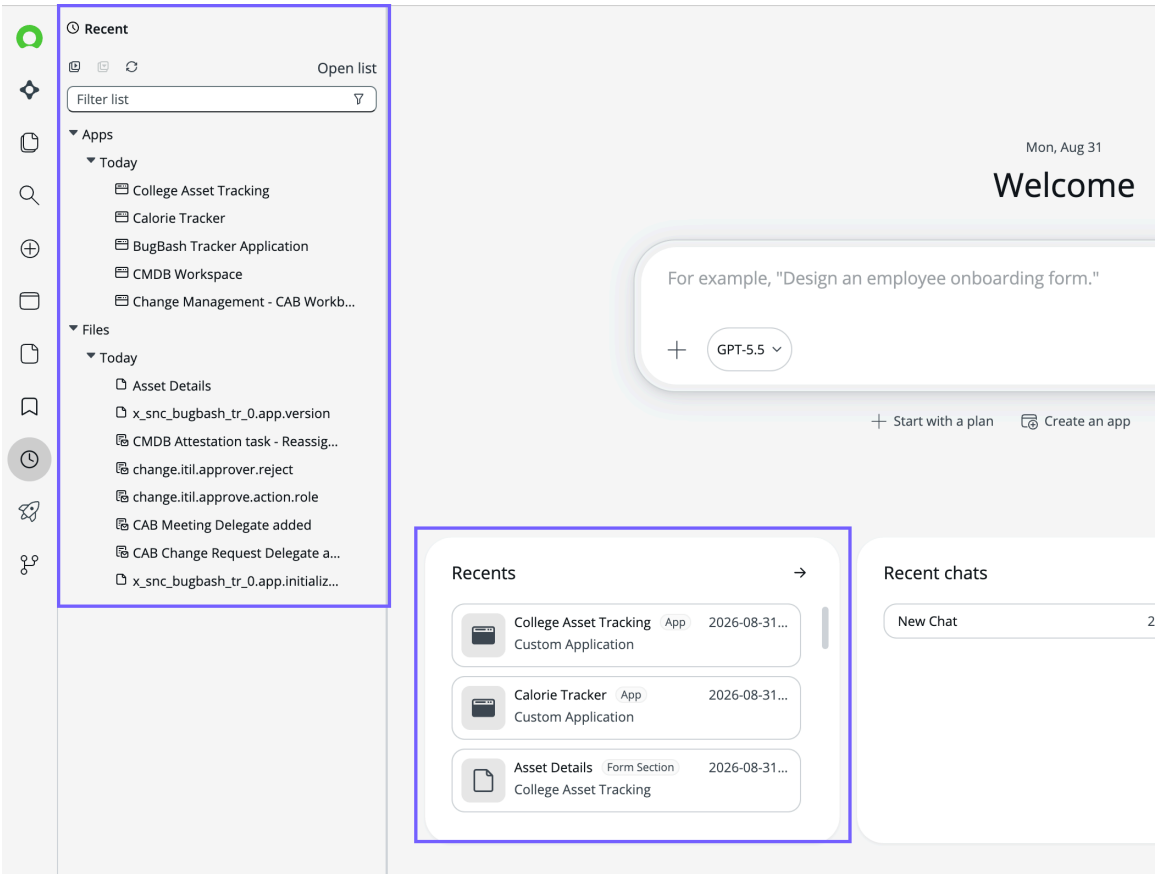
For more information, see Find an app or app file using the Navigator panel.

For more information about working with your application from App details, see App details page.

Opening apps for continued work

After you open an app for the first time, it appears in the Recents section on the ServiceNow Studio home page and the **Recent** section in the Navigator panel.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Create an application in ServiceNow Studio

Create a custom application in ServiceNow Studio, then add data, automation, or other app files using integrated development tools and builders.

Before you begin

As of version 27.1.2, both admins and users with Guided Application Creator (GAC) roles can create apps in ServiceNow Studio. To create a custom global app, the GAC user must have the Global role.

Role required: admin or Guided Application Creator roles

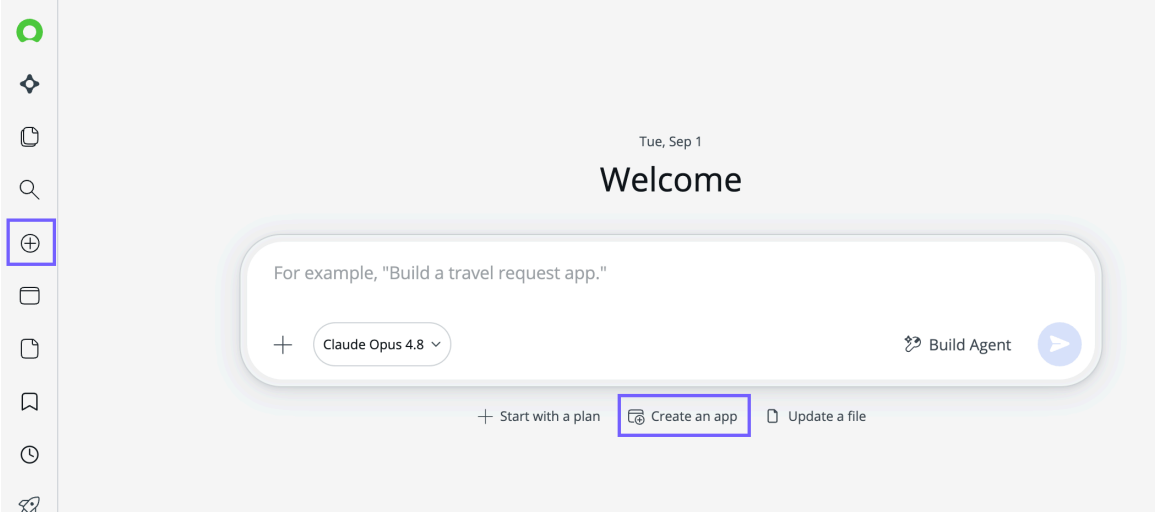
About this task

Use the following steps to create an app manually, or use Build Agent to create your app through conversational interaction. For more information, see Build Agent in ServiceNow Studio.

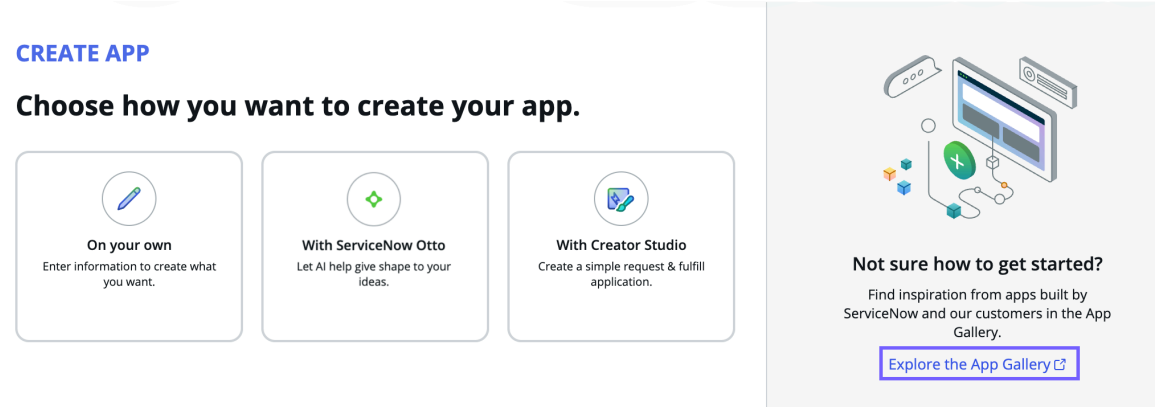
Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Select either the create icon ⊕ next to the Navigator panel or select **Create an app** under the Build Agent chat panel.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



3. Select **App** from the options that appear.
4. Select how to create your app from the available options.



Note: To browse completed apps for reference, select **Explore the App Gallery**. After signing in with your ServiceNow credentials, you can access a library of apps and app files.

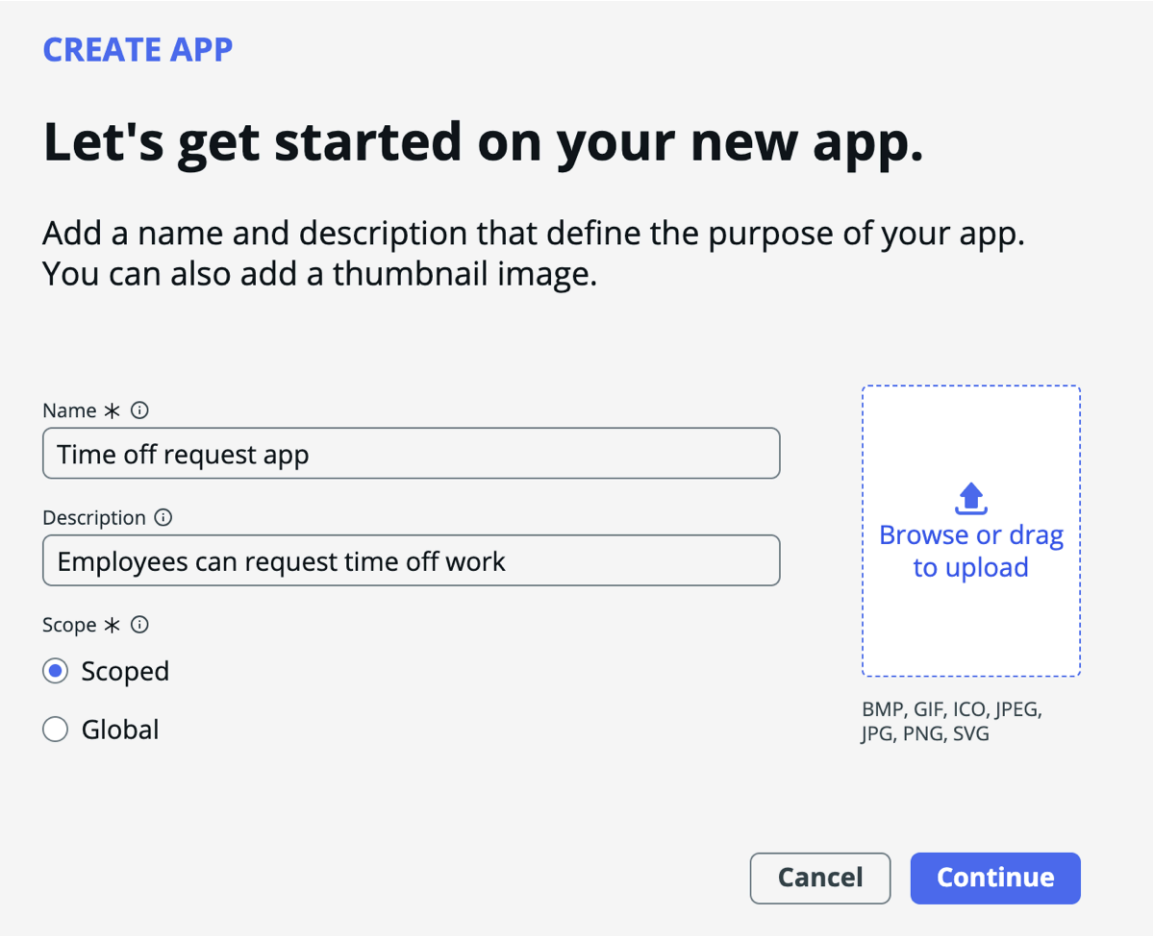
How to create	Description
On your own	Create the application independently, adding content and files of your choosing. Select **On your own** > **Continue** and continue with step 5 in this procedure.
With *ServiceNow Otto*, which opens Build Agent	Begin a conversation with Build Agent to cre ate your application. Select **With *ServiceNow Otto*** > **Start a chat**. The Build Agent chat panel opens. Select **Create an application** or describe what you want to build. For more information, see Create an applica tion using Build Agent default mode.
With Creator Studio	Use Creator Studio to create a simple request and fulfill application.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

How to create	Description
	Select **With** *Creator Studio* > **Continue in** *Creator Studio* to begin creating your application. For more information, see Building apps with Creator Studio.

5. Enter the basic information for the app.

Create an app — basic details



a. Enter a **Name** and **Description** for the application.

b. **Optional:** Add a logo by dragging an image to the **Browse or drag to upload** field, or select the field, choose the image from your file directory, and select **Open**.

c. Specify the **Scope**.

- **Scoped**: The app does not interact with other data on the instance by default. The app can access and change its own tables and business logic, but other apps cannot unless you grant them explicit permission.
- **Global**: All tables and business logic on the instance can interact with the app.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For more information about working with scopes, see Application scope.

d. Select **Continue**.

6. Define user access to the app by adding roles.
ServiceNow Studio automatically defines default admin and user roles. Remove the predefined roles or add more roles as needed. For more information about roles, see Managing roles.

a. Select **Add a role**.

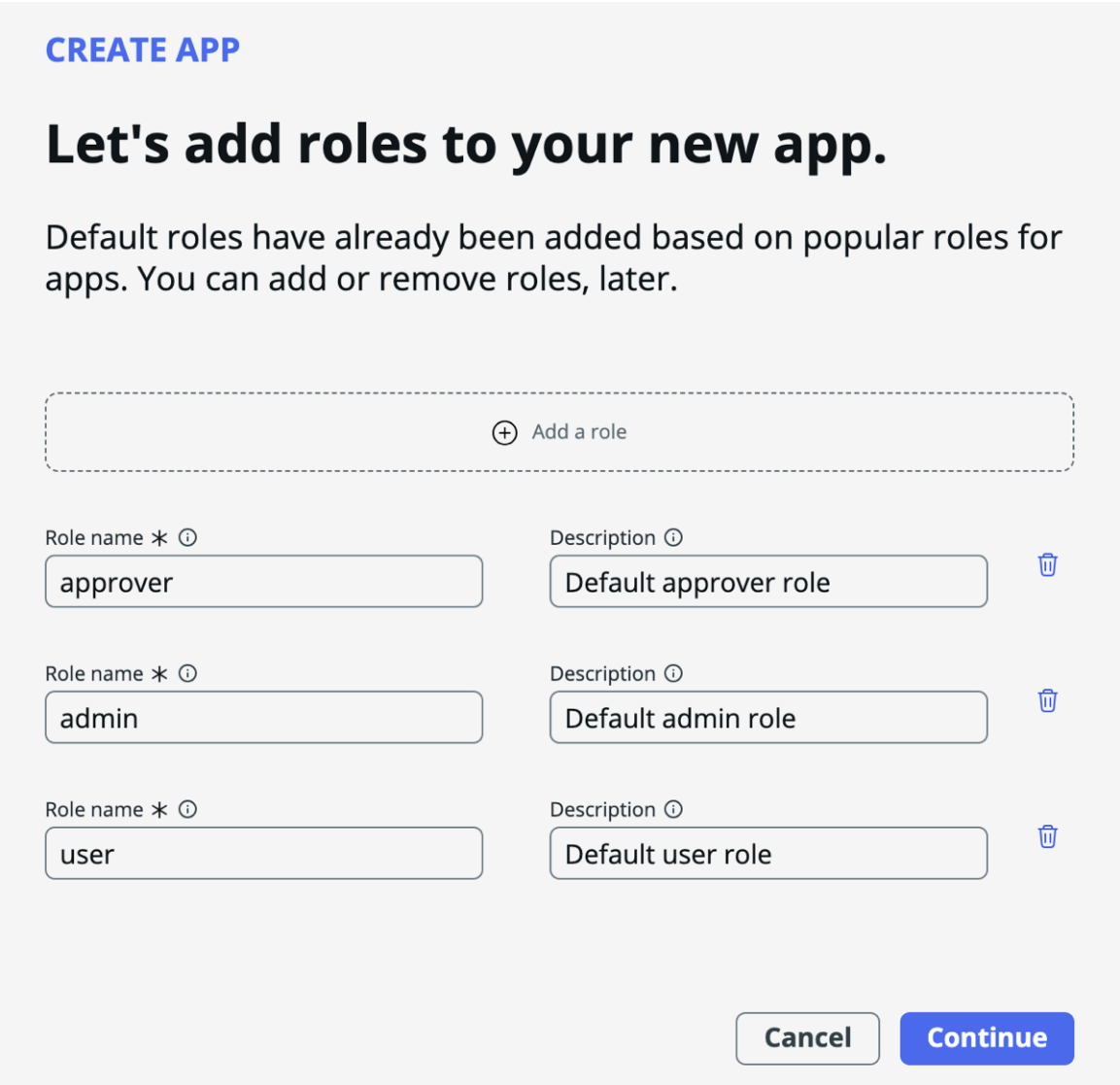
b. Enter the **Role name** and **Description**.

c. Repeat for each additional role you need to add.

d. Select **Continue**.

Note: At least one role is required to continue making updates to the application.

Roles create user access to the app



7. Continue building your app by creating a file or viewing the App details page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Select **Create file** to add files to your application immediately.
- Select **View app details** to open the App details page for your new app.

Your new application is created and ready to develop further.

What to do next

Add application files, dependencies, and cross-scope privileges to your new application.

Add a file to your app in ServiceNow Studio

Add a file to your application immediately after creating it in ServiceNow Studio.

Before you begin

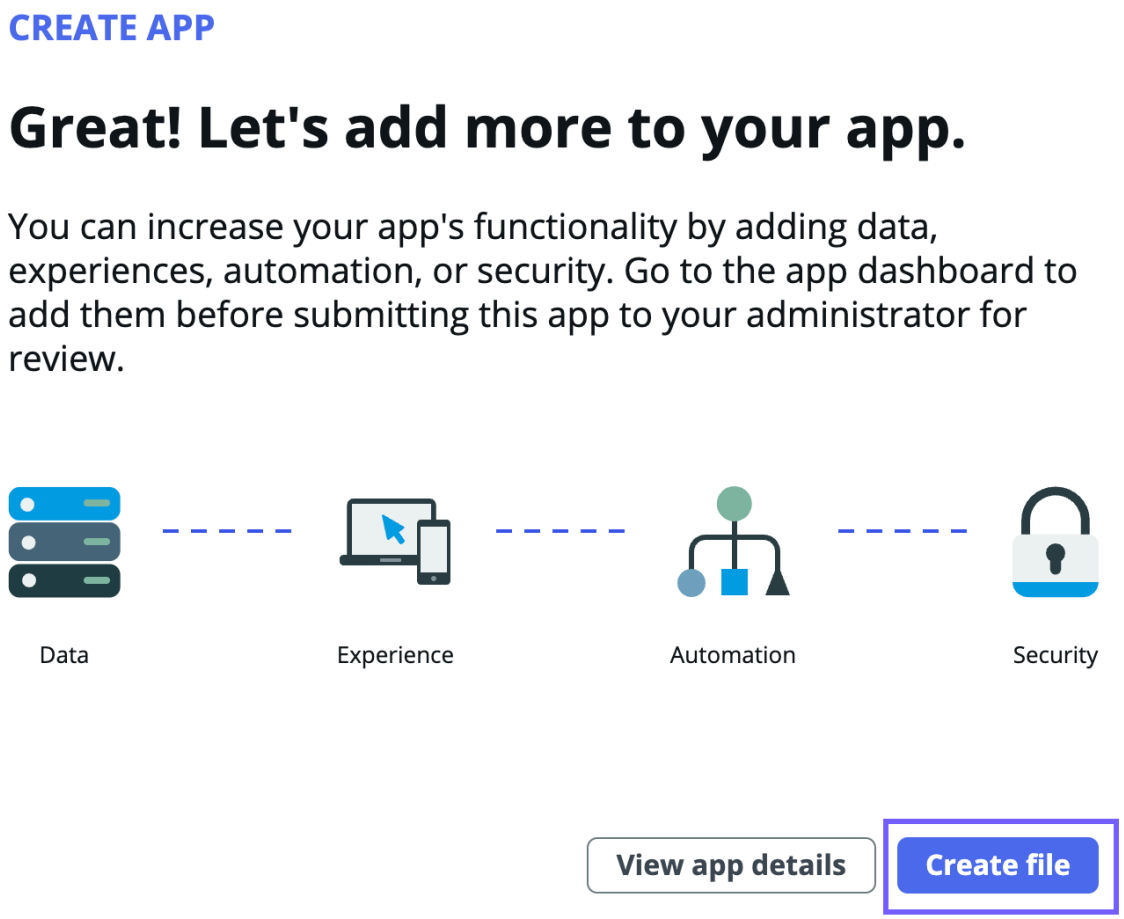
Role required: admin or delegated_developer

About this task

Use this procedure to add files immediately after creating an app. To create files from anywhere else in ServiceNow Studio, see Create an app file in ServiceNow Studio.

Procedure

1. On the app creation confirmation page, select **Create file**.



2. Search for and select the file type you want to create.

Note: Available file types depend on your permissions. If you do not see a file type you expect, contact your administrator.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For more information about each available file type and category, see ServiceNow Studio Navigator panel taxonomy.

3. Select **Continue**.
4. Enter the details for the new app file in the form.
   The form fields vary depending on the file type.
5. Select **Submit**.
   The new file is created and associated with your application.

Create an app file in ServiceNow Studio

Create an app file in ServiceNow Studio to define how an aspect of an application functions — such as which users can access it or how it processes data.

Before you begin

Role required: admin or delegated_developer

About this task

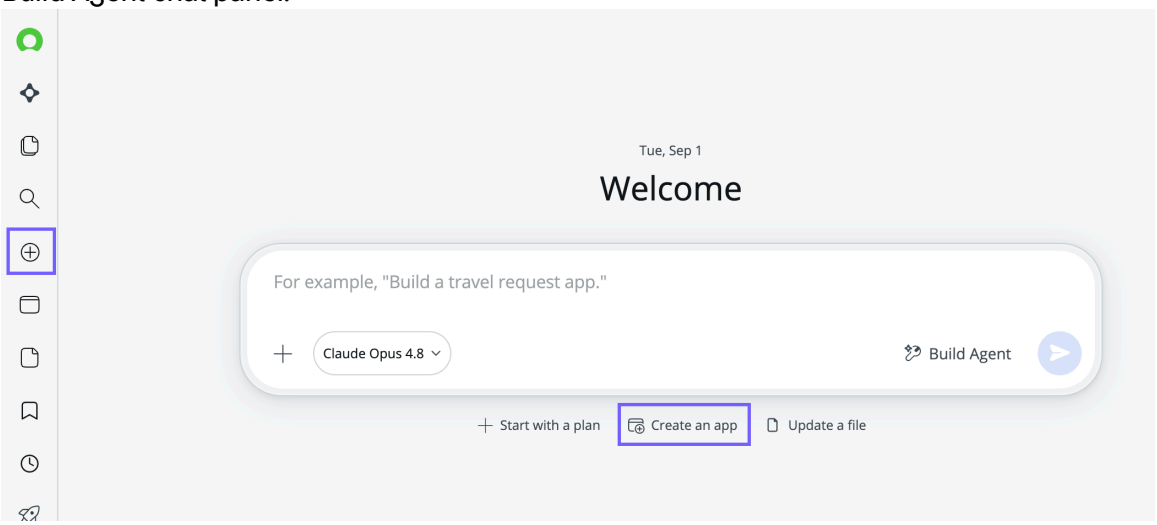
Application files are metadata records for application logic such as business rules, workflows, and script includes. For more information about app files, see Application files and Metadata record categories in the ServiceNow Studio Navigator.

Delegated developers can create app files for apps they have access to. To have an app created, contact your admin.

Use this procedure to create files from anywhere in ServiceNow Studio. To add files to a newly created app, see Add a file to your app in ServiceNow Studio.

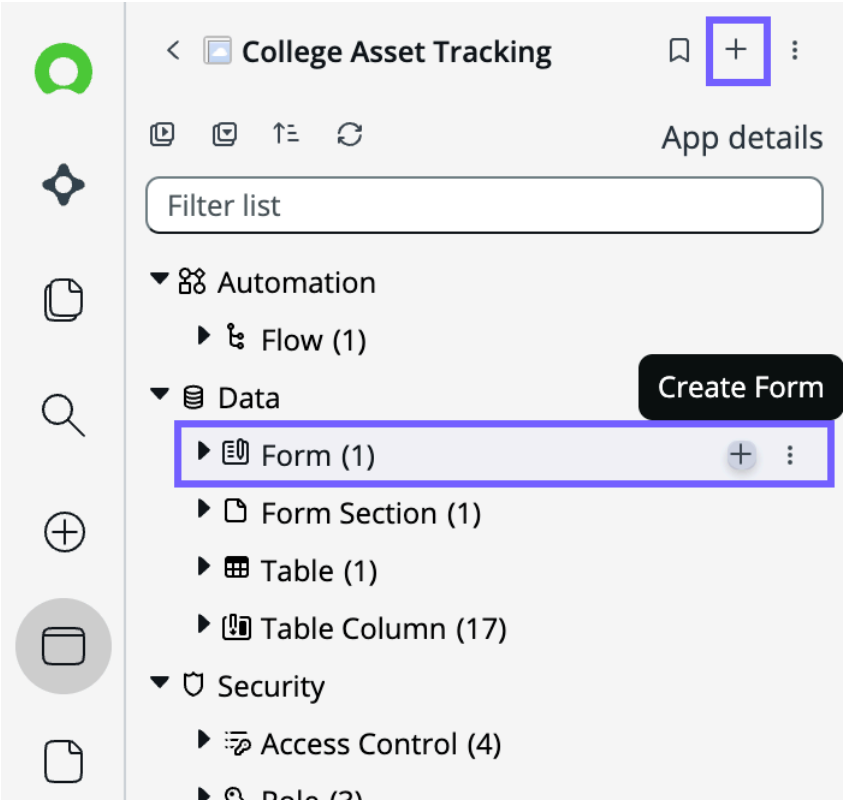
Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Select either the create icon ⊕ next to the Navigator panel or select **Create an app** under the Build Agent chat panel.



3. **Optional:** If an app is already open in the Navigator panel, select the create icon **+** next to any file type to create a file of that type.
   To create a new file of any type, select the create icon **+** next to the application name.

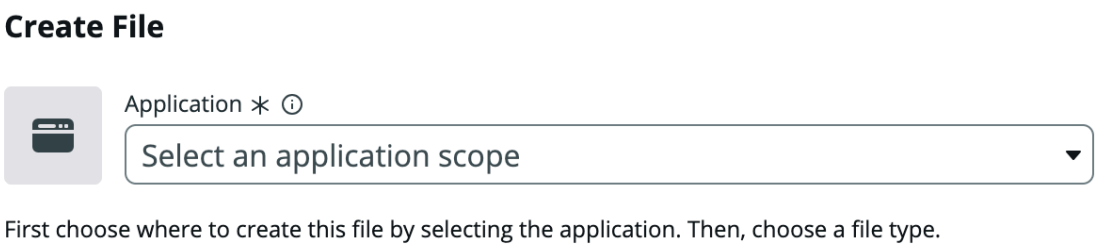
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



4. Select **File**.
5. Specify which scope the file belongs to by entering it in the **Application** field.

   Note: If you are creating a file directly from within an app, the **Application** field is pre-populated with the application name.

   To make the app file available to all apps, select the **Global** scope.



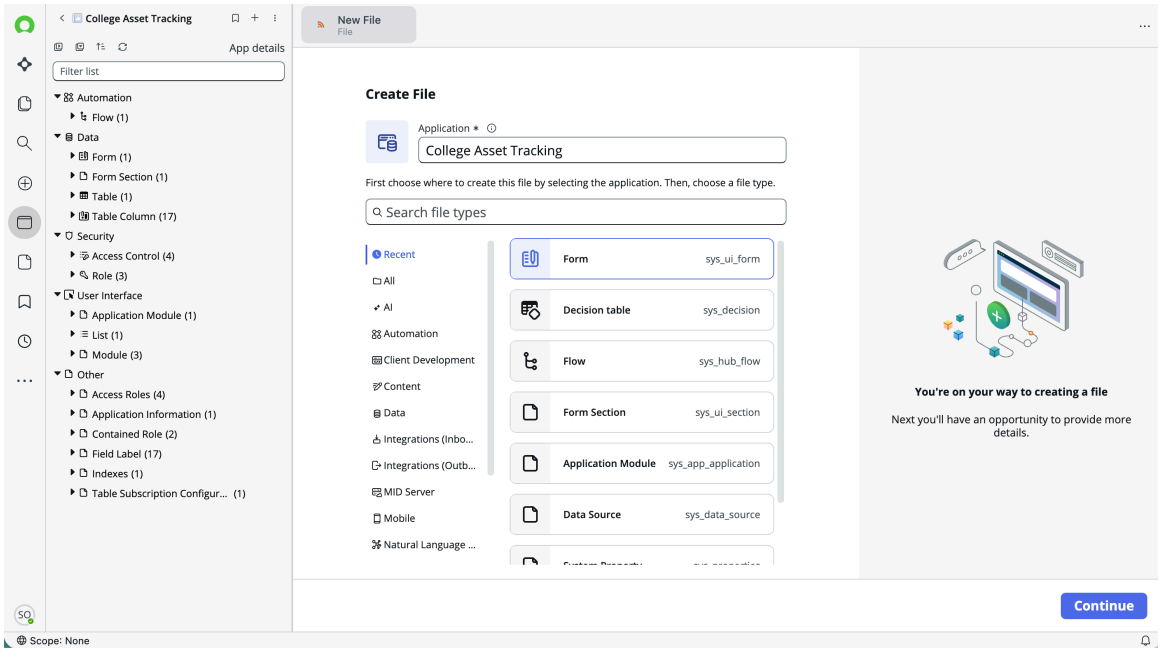
6. Select the card for the file type you want to create.

   Find the file type using one of the following options:

   - Select a file from one of the metadata categories in the **New File** tab.
   - Search for the file type.
   - Select from the list of **Recent** file types.

   Note: Available file types depend on your permissions. If you do not see a file type you expect, contact your administrator.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

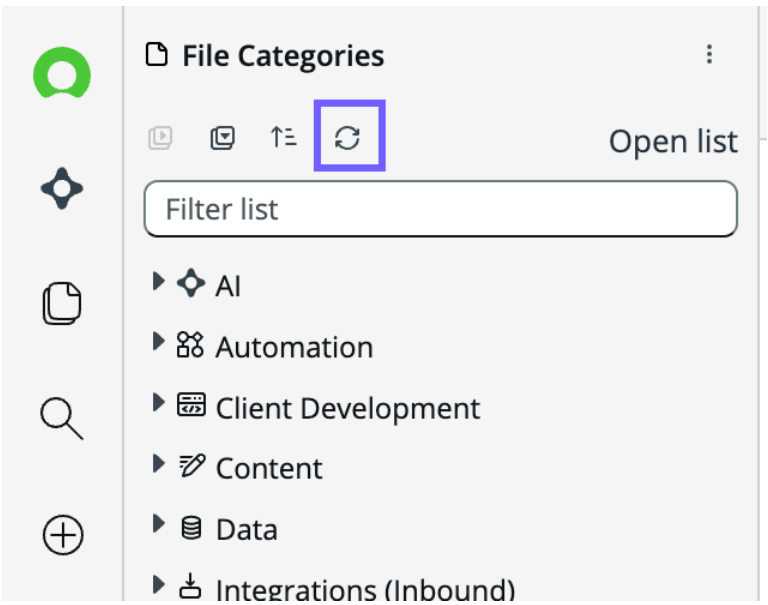


For a list of all available file types, see ServiceNow Studio Navigator panel taxonomy. For more information about each file type, see Metadata record categories in the ServiceNow Studio Navigator.

7. Select **Continue**.
   The record for the app file opens in the same browser tab in ServiceNow Studio.
8. Enter the details for the new app file in the form.
   The form fields vary depending on the file type.
9. Select **Submit**.
   The app file is created and added to your application.

What to do next

Select the refresh icon in the Navigator panel to display the new file in the list of app files.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

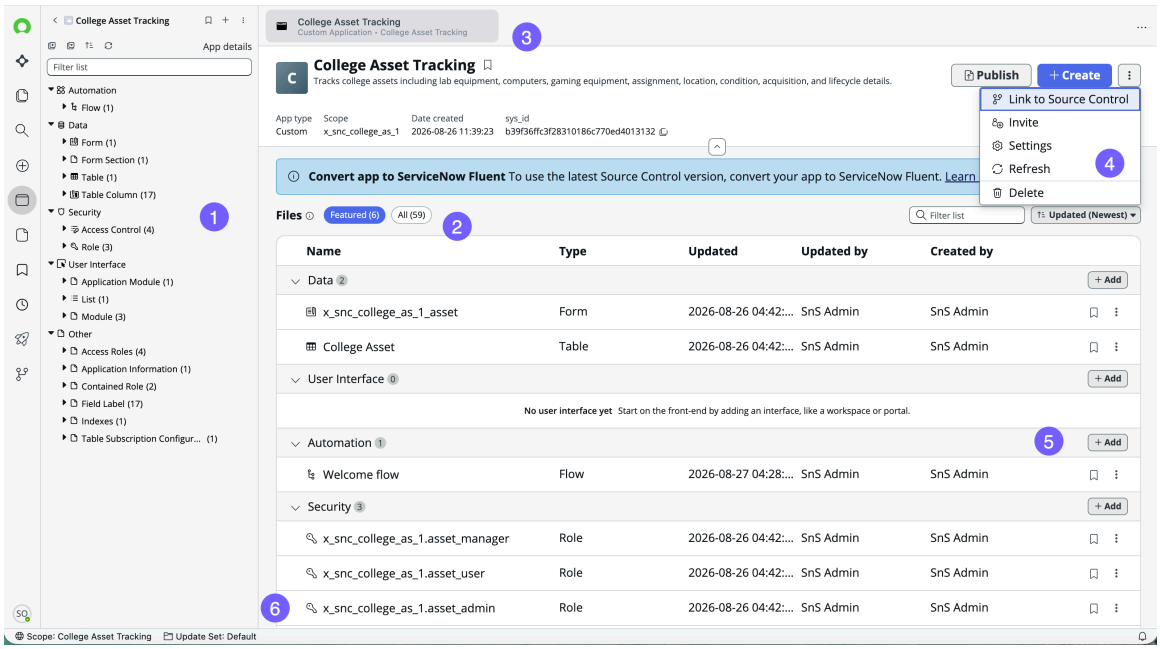
App details page

The App details page in ServiceNow Studio shows metadata about your app and provides access to common development actions — including creating files, updating app settings, and publishing your app for deployment.

App details features

The App details page shows information about your app and provides access to common app actions.

1. **Navigator panel**: View app files organized by category. Bookmark files or apps, open the full list of any file type, or filter the list to find specific files. For more information, see Bookmark apps and app files in ServiceNow Studio.
2. **Files section**: View featured file types (Data, User Interface, Automation, and Security) or all file types. Filter or sort the list to find the files you want to work with. For more information, see Metadata record categories in the ServiceNow Studio Navigator.
3. **Page header**: View app metadata, invite collaborators, or create files directly using the + Create option. Badges next to the app title — such as Creator Studio, AI, or Fluent — indicate where the app was created and how it must be deployed. Easily convert your app to Fluent by selecting **Convert** on the blue menu. For more information, see App details page and Viewing app origination information in ServiceNow Studio.
4. **App actions**: Select the more options icon ⋮ to link the app to source control, invite collaborators, access settings, or publish, refresh, or delete the app. If the app is linked to source control, available source control operations appear in a menu next to the **Publish** button. For more information, see Applications in ServiceNow Studio.
5. **File records**: Select a file to open it, select **Add** to create files of that type, or bookmark files directly from the record. For more information, see Create an app file in ServiceNow Studio.
6. **Scope and update set**: View the scope and update set associated with the current app at the bottom of the page. The scope switches automatically when you switch between tabs in ServiceNow Studio. Create an update set and associate app changes to it so your changes are ready to deploy. For more information, see Open apps and app files across scopes in ServiceNow Studio and Update sets in ServiceNow Studio.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

What can you do from the App details page?

Use the App details page to perform the following actions directly on your app:

- Link an app to source control in ServiceNow Studio
- Work with changes in Git
- Summarize the contents of an app in ServiceNow Studio
- Modify an app's settings in ServiceNow Studio
- Create an app file in ServiceNow Studio
- Create an update set in ServiceNow Studio
- Bookmark apps and app files in ServiceNow Studio
- Delete an app from ServiceNow Studio

Modify an app's settings in ServiceNow Studio

Modify an app's settings in ServiceNow Studio to update the name, description, scoping, or other metadata. Use the Related Links section to publish the app, manage collaborators, and more.

Before you begin

Role required: admin or delegated_developer

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Open the application you want to update.
3. Select **App details** to open the app in the canvas.
   The App details page opens for the selected application.
4. Select the more options icon, then select **Settings**.
   The app settings page opens in a new tab. Use this page to view and edit application scoping, design and runtime, subscription management, primary menu, and end user access settings. The **Related Links** section enables you to publish your app, manage collaborators, and more.
5. On the Custom application form, fill in the fields.
   For a description of the field values, see Custom Application record form.
6. Select **Update**.
   Your changes are saved to the app record.
7. Return to the **App details** tab for the application.
8. Select the reload app details icon to apply the changes from the **App settings** tab.
   The App details page refreshes and displays the updated app settings.

Bookmark apps and app files in ServiceNow Studio

Bookmark apps and app files in ServiceNow Studio so you can access them quickly from the Bookmarks tab in the Navigator panel.

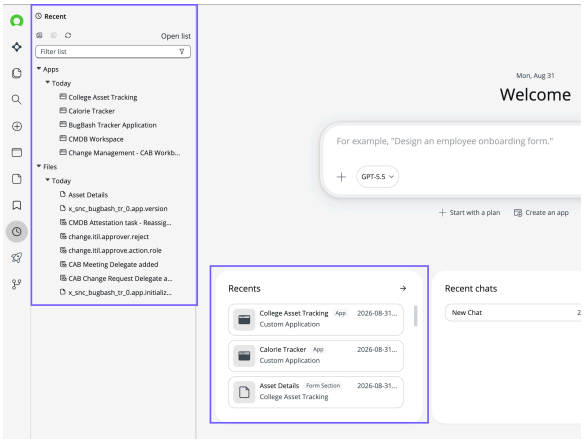
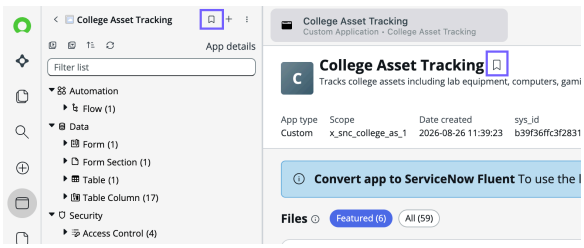
Before you begin

Role required: admin or delegated_developer

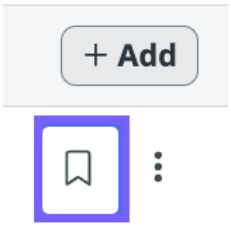
Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Select the Add to bookmarks icon from one of the following locations in ServiceNow Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Location in Service Now Studio	Steps
Home page	**a.** Under **Recently opened files** or **Recently opened apps**, locate the app or file you want to bookmark. **b.** Select the Add to Collections icon on the app or file card. 
Navigator panel	**a.** Select the Apps icon to display all apps, or select the File Cate gories icon to display all files. **b.** Hover over the app or file you want to bookmark. **c.** Select the Add to Collections icon .
Open tab	**a.** Open an app in the canvas. **Note:** If the app you want to bookmark is open in a separate tab, select the tab to bring it into focus in the canvas. **b.** Select the Add to Collections icon next to the application name. 

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Location in Service Now Studio	Steps
App details page	**a.** Open the App details page for your application. **b.** Select the Add to bookmarks icon on the right side of the file you want to bookmark. 

3. **Optional:** To view your bookmarked apps and files, select the Collections icon in the Navigator panel.
   Your bookmarked apps and files appear in the Collections tab in the Bookmarks collection, organized by file type.

Preview an app file in ServiceNow Studio

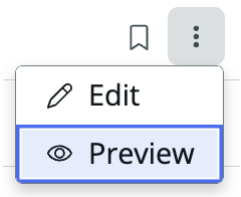
Preview app files in ServiceNow Studio before publishing to confirm that the file looks and functions as expected. Only files with a user interface support preview.

Before you begin
Role required: admin or delegated_developer

About this task
Use preview to verify how an app file looks and behaves before it reaches end users. Files with preview capabilities open directly in preview mode in ServiceNow Studio, bypassing the record view.

Begin or continue a conversation with Build Agent to edit previewed files through a conversational interaction. For more information, see Use Build Agent.

Procedure
1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Preview files from the App details page or from within an open file.

Option	Description
Preview from App details	**a.** Open an application in the canvas. **b.** Next to the file you want to preview, select the more options icon, and select **Pre view** to open the file in preview mode. 

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Description
Preview from an open file	Files with preview capabilities automatically open in preview mode. After you close the preview, reopen the preview by selecting **Pre view** in the file record.

Result
Review the file for layout, functionality, and content errors. Return to the canvas to make corrections, then reopen the preview to verify changes.

Find an app or app file using code search

Use code search in ServiceNow Studio to search through all applications and tables on an instance to locate a specific app, app file, or code snippet.

About this task
Use code search for troubleshooting issues on an instance. For example, if a client script automatically updates a record incorrectly and displays an error message, search for the error message to find where it is being generated. For a detailed example, see the Community article on Using Code Search to make developing on the platform easier.

Before you begin
Role required: admin or delegated_developer

Procedure
1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. In the search bar, select the **Activate code search** toggle.
3. Enter the code snippet to search for in the search bar.
4. Specify whether to search all apps or a specific app.
   - Select **Select all** to search all apps on the instance for the code snippet.
   - Select **Select specific app** and enter the name of the app in the search bar that appears.
5. **Optional:** Limit the search to a table by entering the table name in the **File types / tables** field. Only tables in the same scope are available. For a list of supported file types, see ServiceNow Studio supported file types using code search.
6. Select **View results** to run the search.
   The results open in a new **Code search** tab inside ServiceNow Studio.
7. Expand the script type to review for the code snippet.
   For example, expand the **Notification** section to view all code search results in notification scripts, or the **Access control** section to view all instances of code related to roles and security.

   For a list of types of scripts, see Available script types.
8. Expand the name of an individual script to view the matching code and its corresponding line number.
   Each script that contains the code snippet appears in an expandable section that displays the number of times the code appears in the script.
9. **Optional:** Select **open file** to view the complete script.
   The complete script opens in a new tab in ServiceNow Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Delete an app from ServiceNow Studio

Delete apps that are no longer needed from ServiceNow Studio to remove them permanently from the instance.

Before you begin

Role required: admin or delegated_developer

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Open the application to delete.
3. Select **App details** to open the app in the canvas.
4. Select the more options icon  , and select **Delete**.
5. In the **Permanently delete this app?** modal, enter `Delete`.
6. Select **Close**.

Result

The app is permanently deleted from the instance and no longer appears in ServiceNow Studio.

Source control integration in ServiceNow Studio

Link your application to a Git source control repository in ServiceNow Studio to save, manage, and share multiple versions of a global or scoped application from a non-production instance.

Metadata source control

Metadata source control processes changes in the metadata of your application. Linking an application to source control enables all application developers on a non-production instance to:

- Import applications from a Git repository.
- Pull and apply remote changes from a Git repository.
- Commit all local changes on the instance to a Git repository.
- Create tags to permanently link to a specific version of an application.
- Create branches to maintain multiple versions of an application at the same time.

What are the integration requirements?

To link an application to source control:

- You must have the admin role.
- The non-production instance must have network access to the Git repository.
- Each application must be in its own Git repository.
- The repository user credentials must grant read and write access.

Note: All application developers on the instance share a single set of credentials per repository.

What can you do with source control in ServiceNow Studio?

After linking an application to source control, you can use ServiceNow Studio to manage the repository. From ServiceNow Studio, you can:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Edit the application repository credentials.
- Commit all local changes on the instance.
- Apply remote changes from the repository.
- Create a branch.
- Switch branches.
- Import an application from a remote repository.

Source control integration does not support managing applications on a production instance. To manage applications on a production instance, use the application repository, an update set, or ServiceNow Studio. For more information, see Application sharing.

What can you do with source control from a Git repository?

The ServiceNow AI Platform offers limited support for modifying linked application files outside of an instance. From Git, developers can:

- Move application files to a different Git directory structure.
- Edit application files outside of ServiceNow Studio.

The system generates a properties text file called `sn_source_control.properties` at the root level of the repository. To move application files to a different Git directory structure, set the *`path`* parameter to specify the subfolder path containing the application files. For example, to move your application to the `src/app` subfolder, set the *`path`* to `path=src/app`.

The system generates a `checksum.txt` file in the Git repository to determine if any application files have been changed outside of ServiceNow Studio. When the checksum value from the file matches the current checksum value, the integration skips the validation and sanitization process. When the checksum values do not match, the integration validates and sanitizes the application files as part of the source control operation. The sanitization process:

- Creates upgrade log entries for each sanitization action taken.
- Removes unsupported folders and files from the repository.
- Aborts all source control operations when a system application file fails XML schema validation. For example, if a database dictionary record fails XML schema validation, the system aborts all operations.
- Skips the current source control operation when a non-system application file fails XML schema validation.

The Git integration sanitizes only content within the application path listed in the `sn_source_control.properties` file. Repository content outside the application path is ignored.

How do I connect through a MID Server?

Use an existing MID Server to connect to a source control repository. Connecting an application through a MID Server enables access to repositories behind a firewall.

Source control role permissions

For more information about roles and collaborators, see Application collaboration.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Metadata source control in ServiceNow Studio

Use metadata source control in ServiceNow Studio to manage app versions, commit changes, and collaborate with other developers through a linked Git repository.

After an admin links an app to source control, all application developers on a non-production instance can perform the following actions:

- Import applications from a Git repository.
- Pull and apply remote changes from a Git repository.
- Commit all local changes on the instance to a Git repository.
- Create tags to permanently link to a specific version of an application.
- Create branches to maintain multiple versions of an application at the same time.

For more information, see Link an app to source control in ServiceNow Studio.

Note: When using source control to collaborate with other developers, only changes that have been checked in are available to other developers. For example, if an admin creates a new flow for an app linked to Git, the flow is not available to other users until the admin checks it into Git.

Source control operations in ServiceNow Studio

Most source control operations in ServiceNow Studio run from within the platform. Some operations — such as creating branches and tags — can also be performed directly from the Git repository.

Available source control operations

Operation	Description	Available from
Import from source control	Imports an application from the repository to the local instance. For more information, see Import an app from source control in ServiceNow Studio.	ServiceNow Studio
Link to source control	Enables developers to manage application changes from a Git repository. For more information, see Link an app to source control in ServiceNow Studio.	ServiceNow Studio
Edit repository configuration	Updates the Git repository connection details and credentials. For more information, see Edit a Git repository configuration in ServiceNow Studio.	ServiceNow Studio
Apply remote changes	Updates the local version of the application to match the repository version. For more information, see Pull changes from a repository.	ServiceNow Studio
Commit changes	Updates the repository version of the application to match the local version. For more information, see Commit changes to a repository.	ServiceNow Studio
Stash local changes	Saves local changes and removes them from the working branch so you can return to them later. For more information, see Stash local changes.	ServiceNow Studio
Switch branch	Updates the local version of the application to match a different repository branch. For more information, see Switch repository branches.	ServiceNow Studio

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Available source control operations (continued)

Operation	Description	Available from
Create branch	Creates a branch in the repository to maintain a separate version of the application. For more information, see Create a repository branch.	• ServiceNow Studio • Git repository
Create tag	Creates a tag in the repository to identify a specific application version. For more information, see Create a tag to link to a particular application version.	• ServiceNow Studio • Git repository
Manage stashes	Enables developers to apply or delete stashed changes. For more information, see Manage stashed changes.	ServiceNow Studio
Create repository	Creates a Git repository to store application changes.	Git repository
Create credentials	Creates credentials for repository access.	Git repository
Grant access to repository	Grants read and write access to the repository for a specific set of credentials.	Git repository

Edit a Git repository configuration in ServiceNow Studio

Edit a Git repository configuration in ServiceNow Studio to update the network protocol, credentials, or other connection details.

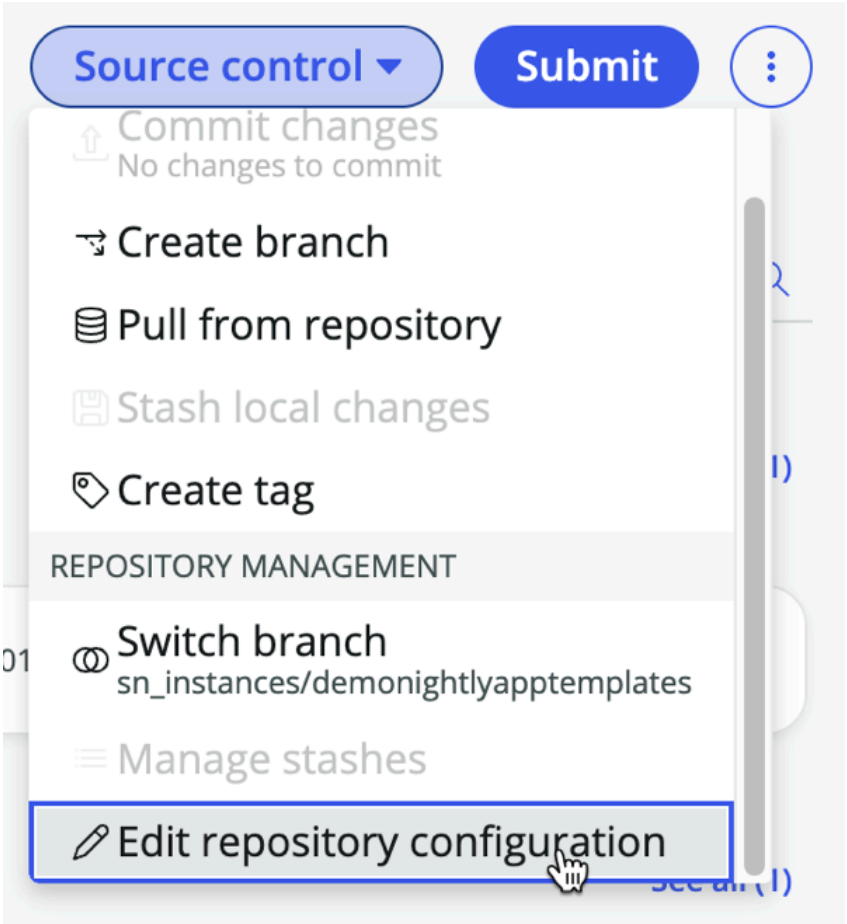
Before you begin

Role required: admin

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. In the file navigator, select the application you want to open.
3. Select **App details** to open the app in the canvas.
4. Select **Source control** > **Edit repository configuration**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



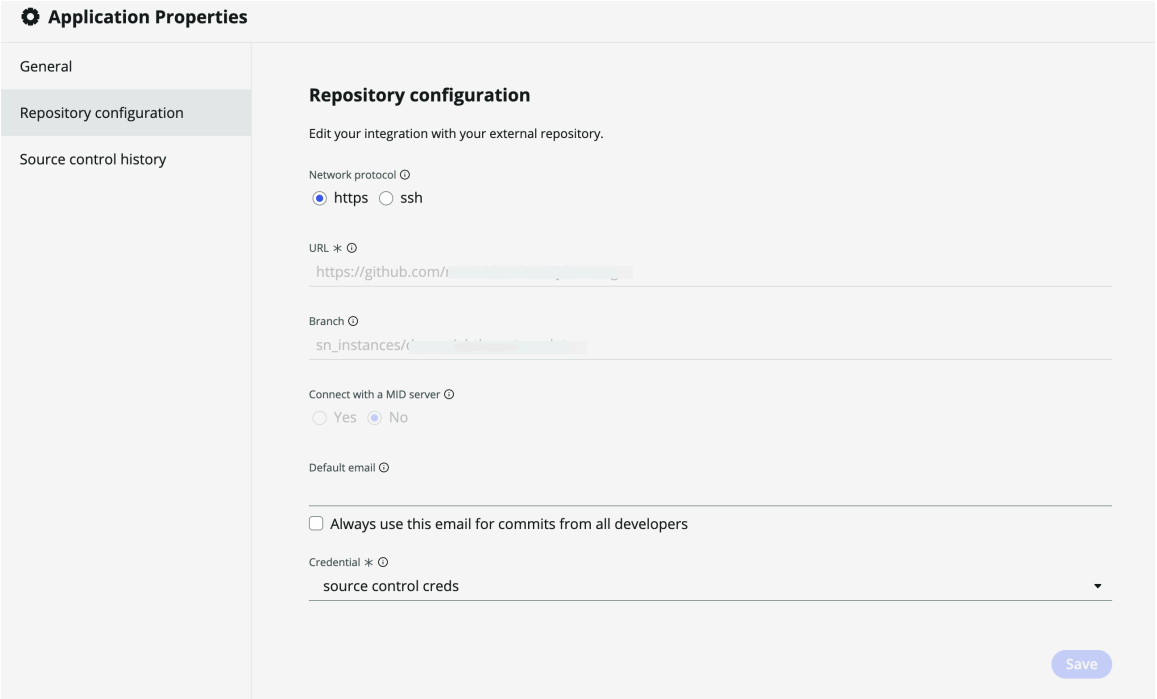
5. Update the fields you want to change.

Repository configuration fields

Field	Description
Network protocol	Protocol for source control: **HTTPS** or **SSH**.
URL	The URL of the Git repository.
Branch	The current branch for the repository.
Connect with a MID Server	Whether or not to use a MID Server connection, which enables access to repositories behind a firewall. **Note:** If you have no MID Server name, select a new one from the drop-down list. If you select a new MID Server, apply the change in the **Source control** menu before performing any further source control operations to avoid errors.
Default email	The committer email address, taken from the sys_user record when available. If a committer's sys_user record email field is empty, the system

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
	generates an alternate email address (username@instancename.service-now.com). Enter a default email address to use when no sys_user email is available. To use the default email address in all cases, select **Always use this email for commits from all developers**.
Credential	The saved credentials to use for the source control connection. All application developers on the instance share a single set of credentials per repository. For more information about working with credentials, see Get started with credentials.



6. Select **Save**.
   The repository configuration is updated and the new settings apply to all future source control operations for this application.

Work with changes in Git

Pull and commit changes in a Git repository directly from ServiceNow Studio to keep app development in sync across your team.

Note: When collaborating with other developers using source control, only changes that have been checked in are visible to other developers. For example, if an admin creates a new flow for an app linked to Git, other ServiceNow Studio users cannot access the new flow until the admin checks it into Git.

Pull changes from a repository

Pull changes from a linked Git repository in ServiceNow Studio to apply remote updates to the local instance.

Before you begin

Link an app to source control in ServiceNow Studio

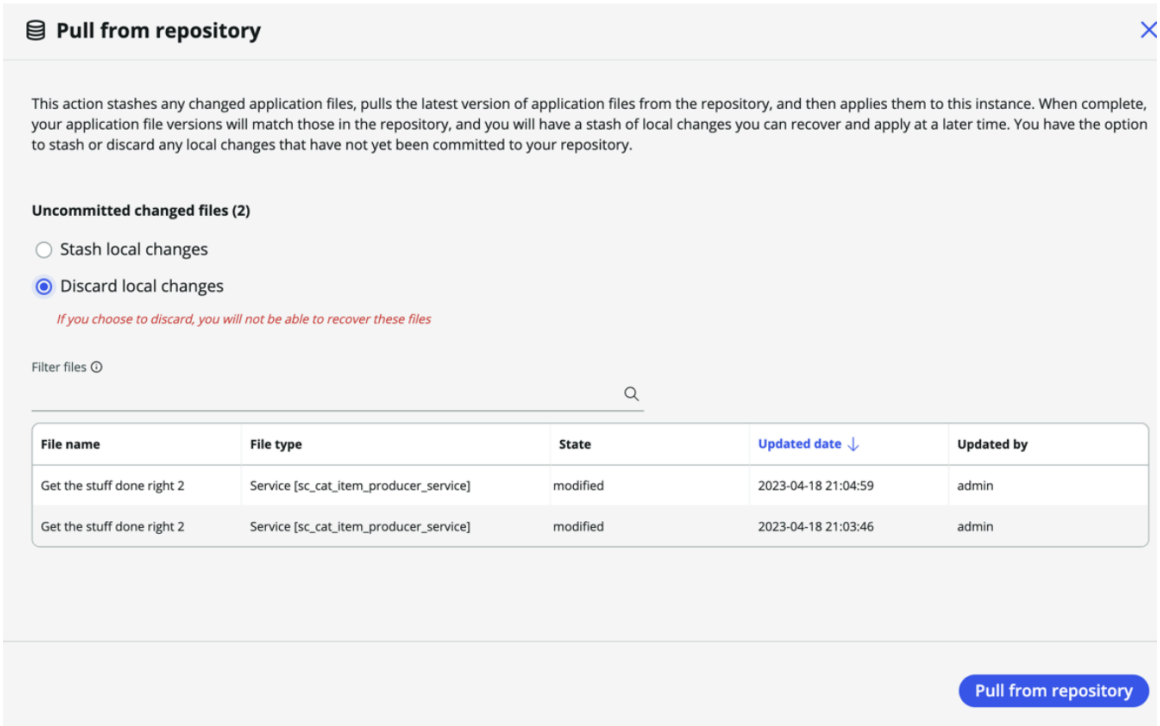
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Role required: admin

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. In the file navigator, select the application you want to open.
3. Select **App details** to open the app in the canvas.
4. Select **Source control** > **Pull from repository**.
5. Choose whether to stash or discard local changes before pulling.

Option	Description
Stash local changes	Saves local changes before switching to an alternate branch. You can later merge or dis card the saved changes.
Discard local changes	Permanently deletes all local changes before switching to an alternate branch. Discarded changes cannot be recovered. **Note:** Use caution when discarding lo cal changes. Because all application de velopers share repository credentials, there is no way to discard only one set of user changes.



6. Select **Pull from repository**.

Result

The following operations occur:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- The ServiceNow AI Platform fetches the most recent changes from the remote repository.
- The ServiceNow AI Platform applies the remote changes to the instance.
- The ServiceNow AI Platform identifies any change conflicts requiring resolution.

If there are conflicts, the system displays the **Resolve Conflicts** window.

Delta loading is enabled by default in sys_properties so your data is not removed. Disable this feature if you want data automatically deleted.

Commit changes to a repository

Commit changes made in ServiceNow Studio to a linked Git repository by selecting individual file changes or committing all changes on the instance at once.

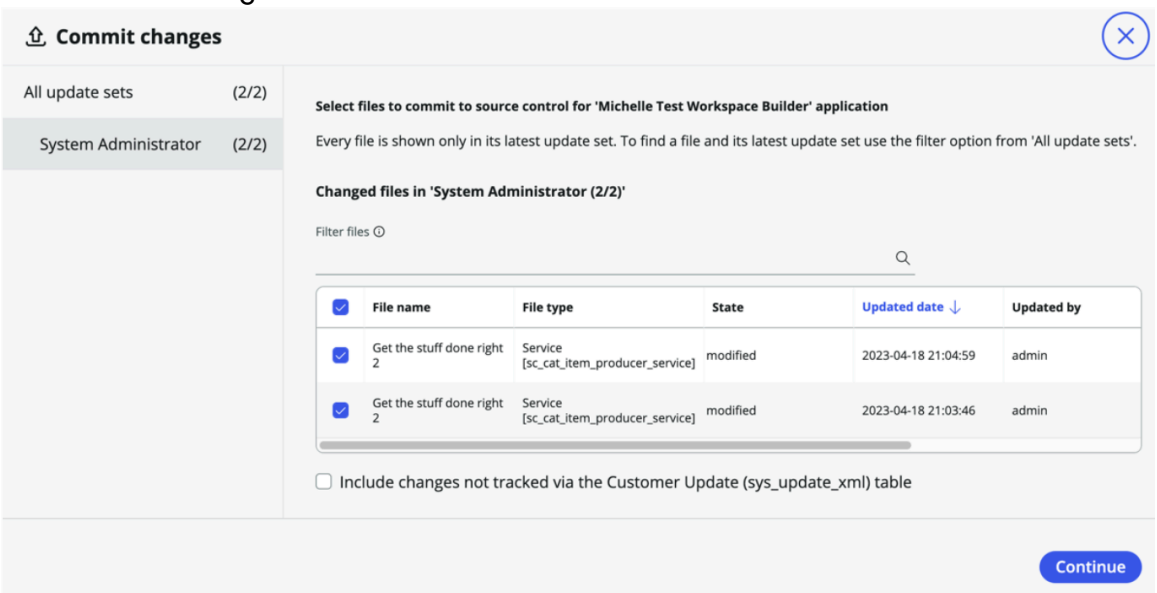
Before you begin

Link an app to source control in ServiceNow Studio

Role required: admin

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. In the file navigator, select the application you want to open.
3. Select **App details** to open the app in the canvas.
4. Select **Source control** > **Commit changes**.
   File changes from all update sets display. By default, changes from the current update set display first.
5. Select the file changes to commit.



6. To include untracked changes, select the **Include changes not tracked via the Customer Update [sys_update_xml] table** check box.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- The default for this check box is set via the *glide.sourcecontrol.default_commit_mode* property.
  - The property can be set to **include_untracked** or **exclude_untracked**.
  - The **include_untracked** mode commits updates to the application that do not generate sys_update_xml records, as well as any user-selected updates.
  - The **exclude_untracked** mode commits only updates selected by the user in the **Select files to commit to source control** modal.
- The base system setting for the property is **exclude_untracked**.
- To hide the check box and use the value of the *glide.sourcecontrol.default_commit_mode* property, create the *sn_devstudio.vcs.allow_commit_mode_selection* property and set it to false.
- Selecting this check box may incur a performance penalty.

Note:

Commits always occur in **include_untracked** mode in the following cases:

- Linking to source control for the first time. For more information, see Link an app to source control in ServiceNow Studio.
- Publishing an application linked to source control from ServiceNow Studio.
- Disabling selective commit mode.

7. Select **Continue**.
8. Enter a comment for the changes in the **Commit comment** field.
9. Select **Commit Files**.

Result

The following operations occur:

- The ServiceNow AI Platform identifies all local changes.
- The ServiceNow AI Platform commits all local changes to the remote repository.

Note: For list of known files that do not have customer update records and are untracked, see Customer Updates table.

Stash local changes

Stash local application file changes in ServiceNow Studio to save them temporarily, pull the latest version from the repository, and recover the stashed changes at a later time.

Before you begin

Link an app to source control in ServiceNow Studio

At least one changed application file must exist before stashing.

Role required: admin

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. In the file navigator, select the application you want to open.
3. Select **App details** to open the app in the canvas.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

4. Select **Source control** > **Stash changes**.
   The system displays a list of locally changed files.
5. Enter a **Stash description**.
6. Select **Stash changes**.

Result
The local changes are saved to the stash. Pull the latest version from the repository and apply the stashed changes when ready.

Manage stashed changes

Apply or delete stashed changes in ServiceNow Studio to restore saved work or remove stashes that are no longer needed.

Before you begin
Link an app to source control in ServiceNow Studio

At least one stashed change must exist.

Role required: admin

Procedure
1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. In the file navigator, select the application you want to open.
3. Select **App details** to open the app in the canvas.
4. Select **Source control** > **Manage stashes**.
5. Select the action next to the stash to manage.

Option	Description
Apply	Commits the stashed changes to the application and checks for conflicts.
Delete	Permanently removes the stashed changes.

6. After applying stashed changes, continue normal pull and push operations to add the changes to the Git repository.
   For more information, see Pull changes from a repository.

Result
The selected action is applied to the stash. If changes were applied, continue with pull and push operations to update the Git repository.

Resolve conflicts

Resolve conflicts in ServiceNow Studio when applying remote or stashed changes that conflict with the local version of the same application file.

Before you begin
Link an app to source control in ServiceNow Studio

At least one stashed change must be applied.

Role required: admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. In the file navigator, select the application you want to open.
3. Select **App details** to open the app in the canvas.
4. Pull from a repository or apply stashed changes.
   If the system detects a conflict, the Resolve Conflicts modal appears and displays the conflicting files.
5. Select an option to resolve the conflict.

Option	Description
Select an action	Apply or discard all stashed changes at once.
Manually merge changes	Select individual changes to apply or discard.

6. To apply or discard all changes, select an **Action**.

Option	Description
Take stashed changes	Applies the application file version from the stashed changes.
Discard stashed changes	Applies the application file version from the most recent pull from the repository.

7. Select **Apply Stashed Changes**.
8. Select **Close dialog**.
9. To merge conflicting changes individually, select **Manually Apply**.
   The ServiceNow AI Platform displays a list of the version differences by field.
10. Select the field values for the merged application file.
11. Select **Save Merge**.

Result

The conflict is resolved and the application file reflects the selected changes. Continue with normal pull and push operations to update the Git repository.

Create versions and branches in Git

Create tags and branches in a Git repository from ServiceNow Studio to version application releases and manage parallel development streams.

Create a tag to link to a particular application version

Create a tag in the Git repository from ServiceNow Studio to mark a specific application version for future reference.

Before you begin

Link an app to source control in ServiceNow Studio

Role required: admin or sn_group_creator.app_creator

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. In the file navigator, select the application you want to open.
3. Select **App details** to open the app in the canvas.
4. Select **Source control** > **Create tag**.
5. Enter the **Tag name**.
6. Select **Create tag**.
   The tag is created in the repository and linked to the current application version.
7. Select **Close**.

What to do next
Commit changes to the new branch.

Switch repository branches

Switch to a different repository branch in ServiceNow Studio to work on a separate version of the application.

Before you begin
A Git repository with one or more available branches must exist.

Role required: admin

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. In the file navigator, select the application you want to open.
3. Select **App details** to open the app in the canvas.
4. Select **Source control** > **Switch branch**.
5. Choose whether to stash or discard local changes before switching.

Option	Description
Stash local changes	Saves local changes before switching to an alternate branch. You can later merge or dis card the saved changes.
Discard local changes	Permanently deletes all local changes before switching to an alternate branch. Discarded changes cannot be recovered. **Note:** Use caution when discarding lo cal changes. Because all application de velopers share repository credentials, there is no way to discard only one set of user changes.

6. Select the branch to switch to.
7. Select **Switch Branch**.

Result
ServiceNow Studio updates the local application to match the branch version from the repository.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Create a repository branch

Create a branch in ServiceNow Studio to develop a new version of an existing app in isolation. The new branch is created in the remote repository and the application, including any uncommitted changes, switches to the new branch.

Before you begin

Role required: admin or sn_group_creator.app_creator

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. In the file navigator, select the application you want to open.
3. Select **App details** to open the app in the canvas.
4. Select **Source control** > **Create branch**.
5. Enter a **Branch name**.
6. **Optional:** To base the branch on an existing tag, select the **Create from Tag** drop-down list and select a tag.
7. Select **Create Branch**.
   The new branch is created in the remote repository and the application switches to the new branch.
8. Select **Close**.

Set a default branch

Set a default branch in ServiceNow Studio to direct new changes to a branch other than main.

Before you begin

Link an app to source control in ServiceNow Studio

Role required: admin

Procedure

1. Follow the steps in Add a system property.
2. Add the *glide.source_control.default_branch_name* property and specify the default branch name of the Git source control repository.
   All developer work is saved to the default branch unless a different branch is specified. If not set, the value defaults to `sn_instances/<instance_name>`.

Result

The default branch is set. New changes in ServiceNow Studio are directed to the specified branch.

View commit history

View the commit history of apps linked to a source control repository in ServiceNow Studio to review what changes were committed, by whom, and when.

Before you begin

Link an app to source control in ServiceNow Studio

Role required: admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. In the file navigator, select the application you want to open.
3. Select **App details** to open the app in the canvas.
4. Select **Source control** > **View history**.
   The commit history list displays with the following details.
5. Sort the commit history list by any of the following details.

Field	Description
Commit message	Comments associated with the commit that describe what was committed.
User	User who committed the change.
SHA-1	Secure hash value identifying this commit in the repository.
Commit date	Date-time stamp of the commit.

6. In the **Files** column, select **View** to see the files associated with the commit.
7. Close the commit history window.

Move application files in a Git repository

Move application files linked to source control to any folder in the repository from ServiceNow Studio to store supporting content, such as automated tests, alongside the applications they support.

Before you begin

Link an app to source control in ServiceNow Studio

Role required: Source control credentials with write access

About this task

Linking an application to source control generates a properties text file called `sn_source_control.properties` at the root level of the repository. The properties file specifies the folder containing the application files. The integration tracks changes to these application files by generating a `checksum.txt` file. When the checksum matches, the integration skips the validation and sanitization process. When the checksum does not match, the integration validates and sanitizes the application files as part of the source control operation. The integration ignores all repository content outside the application path.

Note: Set system properties *`glide.source_control.checksum_required`* to enable optional checksum validations and sanitizations and *`glide.source_control.checksum_quick_install`* to bypass sanitization steps on checksum matches. For more information, see Available system properties.

Procedure

1. Log in to the source control repository linked to the application.
2. Create the destination folder for the application files.
   For example, create the folders `src/app`.
3. Move the folder containing the application files to the destination folder.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For example, move the folder `demo_my_app` to `src/app`.

4. Navigate to the root level of the repository.
5. Open the `sn_source_control.properties` text file in a text editor.
6. For the `path=` property, enter the folder path where the application files were moved.
   For example, enter `path=src/app`.
7. Save the properties file.
   The source control integration reads the updated path and directs subsequent operations to the new folder location.

Fluent source control in ServiceNow Studio

Integrate with remote Git repositories to manage Fluent applications in source control ServiceNow Studio.

You can connect to a Git provider using basic or OAuth 2.0 authentication. Then, initialize a repository for an application in the ServiceNow Studio and push it to a remote repository or clone a remote repository that contains an application. After setting up authentication and connecting to a repository, you can use common Git commands to manage applications in source control.

Instance requirements

If an instance includes modified configurations for saving attachments, the following requirements must be met to perform Git operations:

- If the `glide.attachment.extensions` system property is configured, it must contain `txt,gitdata`.
- If the `glide.security.attachment_type.use_blacklist` system property is set to true:
  - The `glide.attachment.blacklisted.extensions` system property must not contain `txt,gitdata` and
  - The `glide.attachment.blacklisted.types` system property must not contain `text/plain,application/octet-stream`.

If you want to use custom extensions for attachments, set the `sn_glider.git.attachment.extension.text` and `sn_glider.git.attachment.extension.binary` properties to custom values. For more information, see ServiceNow Studio properties.

Connect to a Git provider using basic authentication

Connect to a Git domain or repository using basic authentication credentials to manage applications in Fluent source control from ServiceNow Studio.

Before you begin

- From your Git provider, generate a personal access token to use for basic authentication to the Git domain. You must provide your Git user name and personal access token when configuring your credentials for ServiceNow Studio.
- Create a dedicated Git repository for an application in a Git provider such as GitHub, GitLab, Bitbucket, or Azure Repos.

Role required: admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Use one of the following keyboard shortcuts to open the command palette:
   - Windows: Ctrl-Shift-P
   - Mac: Cmd-Shift-P
3. Enter `Git: Set IDE Git credentials` and press Enter.
4. From the New Git credential form, select **Basic auth**.
5. On the form, fill in the fields.

New Git credential form

Field	Description
Git repository URL	HTTPS URL of a Git repository associated with your Git credentials.
Git username	Git username.
Personal access token	Personal access token that you generate from your Git provider.
Use this credential for all repositories	Option to use the credentials for all repositories in the Git domain associated with the Git repository URL.

6. Select **Submit**.

Result

Your Git credentials are associated with your user on the instance and used for all repositories in the domain from the Git repository URL. If you add different credentials for a repository in the same domain, the new credentials are used and the previous credentials are set to inactive.

What to do next

After initializing or cloning a repository, you can begin using Fluent source control. For more information, see Using Fluent source control in ServiceNow Studio.

To manage existing Git credentials, use the `Git: Manage Git credentials` command from the command palette.

Connect to a Git provider using OAuth 2.0

Set up an OAuth 2.0 application registry and credentials to connect to your Git provider from ServiceNow Studio.

Configure an OAuth 2.0 application registry for ServiceNow Studio

Configure how the client ID and secret are sent to the OAuth 2.0 provider associated with your Git provider.

Before you begin

Create an OAuth application with your Git provider and configure it to redirect to your instance. In this OAuth application, use your instance URL as the homepage URL and `https://<instance>/oauth_redirect.do` for the authorization callback URL. GitHub, GitLab, Bitbucket, and Azure Repos are supported by default.

Role required: admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Important: For Azure Repos, the maximum length of the **Client Secret** field must be updated to 2048 before you add the secret.

1. Navigate to **All** > **System Definition** > **Tables**.
2. Filter the table by entering `oauth_entity` for the **Name** field.
3. Select the Application Registries [oauth_entity] table.
4. In the Columns related list, locate the Client Secret column and enter `2048` as the value of its **Max length** attribute.
5. Select **Update**.

About this task

To use OAuth 2.0 authentication with ServiceNow Studio, you must register the OAuth application you created from your Git provider. Follow this procedure to configure an OAuth application registry [oauth_entity] on your instance.

Procedure

1. Navigate to **All** > **System OAuth** > **Application Registry**.
2. Select **New**.
3. On the interceptor page, select **Connect to a third-party OAuth provider**.
4. On the form, fill in the fields.

   For additional information about fields on the form, see Connect to a third-party OAuth provider.

Application Registries form

Field	Description
Name	Unique name for the third-party OAuth connection.
Client ID	Client ID of the OAuth application in your Git provider.
Client Secret	Client secret of the OAuth application in your Git provider.
OAuth API Script	Script used to customize request and response to the external OAuth provider. Select one of the following depending on your Git provider: ○ GitHub: OauthAPIScriptForGitHub ○ GitLab: OauthAPIScriptForGitLab ○ Bitbucket: OauthAPIScriptForBitbucket ○ Azure Repos: OauthAPIScriptForAzureRepos **Note:** To use other Git providers, you can use these scripts as examples to create your own OAuth API script for your OAuth provider. The name of the script must begin with "Oauth".
Default Grant type	Default grant type used to establish the token. Select **Authorization code**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
	An authorization code is granted to the client to obtain an access token, which is then used to obtain access to the resource.
Refresh Token Lifespan	Time, in seconds, that the refresh token is valid.
Application	Application scope that contains this record.
Accessible from	Option to make the application registry accessible from all application scopes or from this scope only.
Active	Option for turning the OAuth application on or off.
Authorization URL	OAuth authorization code endpoint for your Git domain. For example: ○ GitHub: `https://github.com/login/oauth/authorize` ○ GitLab: `https://gitlab.com/oauth/authorize` ○ Bitbucket: `https://bitbucket.org/site/oauth2/authorize` ○ Azure Repos: `https://app.vssps.visualstudio.com/oauth2/authorize`
Token URL	OAuth server token endpoint for your Git domain. For example: ○ GitHub: `https://github.com/login/oauth/access_token` ○ GitLab: `https://gitlab.com/oauth/token` ○ Bitbucket: `https://bitbucket.org/site/oauth2/access_token` ○ Azure Repos: `https://app.vssps.visualstudio.com/oauth2/token`
Redirect URL	OAuth callback endpoint. If empty, the instance auto-generates a value of `https://<instance>/oauth_redirect.do`.
Use mutual authentication	Option to use mutual authentication. Leave this option cleared to turn off using mutual authentication for token request and revocation.
Send Credentials	Method in which the OAuth client populates the client credentials in the request. ○ For GitHub, GitLab, or Bitbucket, select **As Basic Authorization header**. ○ For Azure Repos, select **In Request Body (Form URL-Encoded)**.

5. Select **Submit**.

What to do next

Configure OAuth 2.0 credentials to connect to a Git provider

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Configure OAuth 2.0 credentials to connect to a Git provider

Connect to a Git domain or repository using OAuth 2.0 credentials to manage applications in source control from the ServiceNow Studio.

Before you begin

- An administrator must configure how the client ID and secret are sent to the OAuth 2.0 provider associated with your Git provider. For more information, see Configure an OAuth 2.0 application registry for ServiceNow Studio.
- Create a dedicated Git repository for an application in a Git provider such as GitHub, GitLab, Bitbucket, or Azure Repos.

Role required: admin

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Use one of the following keyboard shortcuts to open the command palette:
   - Windows: Ctrl-Shift-P
   - Mac: Cmd-Shift-P
3. Enter `Git: Set IDE Git credentials` and press Enter.
4. From the New Git credential form, select **OAuth**.
5. On the form, fill in the fields.

New Git credential form

Field	Description
Git repository URL	HTTPS URL of a Git repository associated with your Git credentials.
Git username	Git username.
Select an OAuth profile	OAuth 2.0 credentials from your Git provider. The OAuth profile is created with the application registry, which must be configured by an administrator.
Use this credential for all repositories	Option to use the credentials for all repositories in the Git domain associated with the Git repository URL.

6. Select **Submit**.

   The first time you use a Git command, you're prompted to authorize your user from the Git provider.

Result

Your Git credentials are associated with your user on the instance and used for all repositories in the domain from the Git repository URL. If you add different credentials for a repository in the same domain, the new credentials are used and the previous credentials are set to inactive.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

What to do next

After initializing or cloning a repository, you can begin using source control. For more information, see Using Fluent source control in ServiceNow Studio.

To manage existing Git credentials, use the `Git: Manage Git credentials` command from the command palette.

Configure a MID Server to use source control

Configure a MID Server to use source control with ServiceNow Studio if your Git provider is behind a firewall.

Before you begin

Install a MID Server with a REST capability. For more information, see Installing the MID Server and Configure MID Server capabilities.

Role required: admin

Procedure

1. Change the application scope of the instance to ServiceNow IDE.
   a. In the Unified Navigation, select the globe icon  .
   b. Select **Application scope**.
   c. Select **ServiceNow IDE**.
2. Navigate to **All** > **MID Server** > **Applications**.
3. Select **New**.
4. On the form, fill in the fields.

MID Server Application form

Field	Description
Name	Name for the MID Server Application.
Default MID Server	MID Server configured with a REST capability.
Application	The ServiceNow IDE.
Included in application ALL	Option to include this MID Server Application in the definition of ALL for a MID Server.

5. Select **Submit**.

What to do next

If you haven't already, configure basic or OAuth 2.0 authentication to connect to a Git domain or repository. For more information, see Connect to a Git provider using basic authentication or Connect to a Git provider using OAuth 2.0.

Note: The MID Server user must have the sn_glider.ide_git_user role or admin role to perform Git operations in ServiceNow Studio. For more information, see Create the MID Server user and grant the role and ServiceNow IDE MID Server User [sn_glider.ide_git_user].

Clone a Git repository from ServiceNow Studio

Clone a remote Git repository to collaborate on applications in Fluent source control in ServiceNow Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

- Set your basic or OAuth 2.0 credentials for ServiceNow Studio to connect to your Git repository. For more information, see Connect to a Git provider using basic authentication and Connect to a Git provider using OAuth 2.0.

Role required: admin

About this task

You can clone a remote Git repository that contains Fluent applications. The repository must contain one or more applications with `now.config.json` and `package.json` files.

Note: ServiceNow Studio supports cloning from Git servers on Git version 2.3.2 or later.

Cloning is intended for developing an application on multiple non-production instances and managing it in a single repository. To publish an application and deploy it to a production instance, use the Application Repository. For more information, see ServiceNow application repository.

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Open the Explorer tab.
3. Use one of the following keyboard shortcuts to open the command palette:
   - Windows: Ctrl-Shift-P
   - Mac: Cmd-Shift-P
4. Enter `Git: Clone` and press Enter.
5. Enter a remote Git repository URL and press Enter.
6. On the Status bar, select **Build and Install**.

Result

The application is added to the instance and your workspace with the files from the remote repository. Only the default branch is cloned initially. To check out another branch, you must fetch the other branches from the remote repository using the `Git: Fetch` command from the command palette.

What to do next

You can check out or create branches in the repository and push changes to the remote repository. For more information, see Using Fluent source control in ServiceNow Studio.

Initialize a Git repository

Initialize a local Git repository for an application in ServiceNow Studio and push it to a remote Git repository to manage an application in source control.

Before you begin

- Create or convert an application with ServiceNow Studio. For more information, see Create an application in ServiceNow Studio or Convert an application to Fluent.
- Create a dedicated Git repository for the application from your Git provider.
- Set your basic or OAuth 2.0 credentials for ServiceNow Studio to connect to your Git repository. For more information, see Connect to a Git provider using basic authentication and Connect to a Git provider using OAuth 2.0.

Role required: admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

About this task

An application on an instance can be connected to only one repository at a time. To clone an application that exists in a remote Git repository, see Clone a Git repository from ServiceNow Studio.

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Add an application that isn't connected to a Git repository to the Explorer tab.
   a. On the Apps tab, select the application you want to add.
   b. Select the more options icon, and select **Add app to explorer**.
3. Use one of the following keyboard shortcuts to open the command palette:
   - Windows: Ctrl-Shift-P
   - Mac: Cmd-Shift-P
4. Enter `Git: Initialize Repository` and press Enter.
5. Select the application for which you want to initialize a Git repository and press Enter.
6. Select **main** as the default branch name or enter another name and press Enter.
7. Select the stage all untracked changes icon.
8. Enter a commit message and select the commit icon.
9. Select the more actions menu icon and select **Push**.
10. Enter a remote repository URL and press Enter.

Result

The application is available in the remote repository.

Trouble?

If your Git credentials aren't configured or are inactive, the application isn't pushed to the remote repository. When prompted to configure your credentials, select **Configure** to configure your Git credentials and then try again.

What to do next

Check out or create branches in the repository and push changes to the remote repository. For more information, see Using Fluent source control in ServiceNow Studio.

Using Fluent source control in ServiceNow Studio

Use Git commands and other source control features in ServiceNow Studio to manage changes to an application across a development team.

Currently, only admins can use Fluent source control in ServiceNow Studio.

Git commands

After initializing a local Git repository in ServiceNow Studio, you can perform Git commands from the Changes tab or command palette, including but not limited to the following commands.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Git commands in ServiceNow Studio

Command	Description
Checkout to (··· > **Checkout to...**) Command palette: **Git: Checkout to...**	Check out another branch from the repository. Select the branch from the list or create a branch and push it to the remote repository. Only one branch per repository can be checked out at a time on an instance as a sandbox in Developer Sandboxes.
Clone (··· > **Clone**) Command palette: **Git: Clone**	Clone a remote repository to add an application to your workspace. For more information, see Clone a Git repository from ServiceNow Studio.
Commit (✓) Command palette: **Git: Commit**	Commit your staged changes to the local repository. Enter a commit message to describe your changes.
Create branch (··· > **Branch** > **Create branch...**) Command palette: **Git: Create branch...**	Add a branch to a repository and check it out.
Discard (↶) Command palette: **Git: Discard Changes**	Discard changes to undo modifications to an application.
Fetch (··· > **Fetch**) Command palette: **Git: Fetch**	Fetch to sync commits and branches from the remote repository into the local repository.
Pull (··· > **Pull**) Command palette: **Git: Pull**	Pull to merge the latest changes from the remote repository into the local repository.
Push (··· > **Push**) Command palette: **Git: Push**	Push your committed changes to the remote repository.
Stage (+) Command palette: **Git: Stage Changes**	Stage the changes in your working directory that you want to commit. When you stage changes, files move from the Changes list to the Staged Changes list.
Stash (··· > **Stash**) Command palette: **Git: Stash**, **Git: Pop**, **Git: Drop**, **Git: List**, **Git: Apply**, **Git: Clear**	Stash all uncommitted changes to save them in your working directory locally and come back to them later. The pop, drop, list, apply, and clear subcommands are supported when stashing. Select **Stash** > **Apply** to reapply your changes in your working directory.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Merge conflicts

If there's a conflict between local and remote changes in a file, the file is listed under Merge conflicts in the Changes tab  . You can review conflicts in the editor and resolve them by accepting the current or incoming change, both, or manually editing the file.

Commit history

The commit history for a branch is listed in the Commits section of the Changes tab  and includes details about the commits. To see the commit history for a file, navigate to the Explorer view  , select the file, and expand the Timeline section. When you select a commit, a Diff editor comparing the changes opens.

Repository changes

To update the remote repository an application is connected to, you can use the `Git: Update remote origin` command from the command palette and enter a different remote repository URL.

Building apps in source code in ServiceNow Studio

Developing and maintaining applications in source code enables you to create and modify complex applications, manage code in source control more easily, and catch errors at build time.

To create apps in ServiceNow Studio, you use ServiceNow Fluent, a domain-specific programming language, to define the metadata that makes up applications. ServiceNow Fluent includes APIs for defining the different types of metadata. For more information, see ServiceNow Fluent.

Workspaces

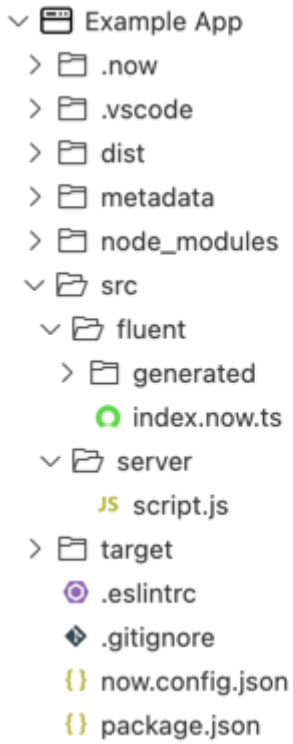
The concept of workspaces is a bit different in ServiceNow Studio than in ServiceNow IDE. In the ServiceNow IDE, you can add any applications that you're working on to a workspace to navigate through them all from one place. You can create multiple workspaces to group different sets of applications. Workspaces are specific to a user, and applications can be added or removed from a workspace at any time.

In ServiceNow Studio, the **Explorer** tab acts as a single, persistent workspace, where you can add any application that has been converted to Fluent. For that reason, workspaces don't need to be created in ServiceNow Studio to create or group applications.

Application structure

Fluent applications created or converted in ServiceNow Studio include source code files and metadata XML files. The `package.json` and `now.config.json` files define the application structure, which is similar to that of Node.js applications or Node Package Manager (npm) packages.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



By default, applications include the following directories and files. You can modify certain aspects of the application structure to suit your needs in the `now.config.json` file.

Application structure in source code

Directory or file	Description
.vscode	Directory containing recommended Visual Studio Code extensions.
dist	Directory containing the build artifacts for packaging. This directory includes the following subdirectories: • `app`: Directory containing the built metadata XML files. • `static`: Directory containing the built static asset files.
metadata	Directory containing the application metadata (XML) of the application, such as table schemas and business rules, organized in the same directory structure as existing ServiceNow applications. **Note:** Application metadata shouldn't be edited from the XML files. Edit application metadata in the source code or on the ServiceNow AI Platform.
node_modules	Directory containing the third-party Node.js modules on which your application depends.
src	Directory containing the source code of your application. This directory includes the following subdirectories:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Application structure in source code (continued)

Directory or file	Description
	• `client`: Directory containing the client-side files for developing user interfaces. • `fluent`: Directory containing ServiceNow Fluent code in `.now.ts` files. The `generated` subdirectory contains the application files converted to ServiceNow Fluent. • `server`: Directory containing JavaScript module code in `.js` or `.ts` files.
target	Directory containing an installable package (`.zip` file) to upload to an instance.
.eslintrc	File containing the ESLint configuration. ESLint helps identify and fix issues in the application code.
.gitignore	File containing a list of directories or files for Git to ignore. These files aren't tracked in source control.
now.config.json	File containing the ServiceNow application configuration. The `now.config.json` file must be in the base directory for an application. You can configure aspects of an application by adding support parameters. For more information, see Custom application configuration in source code.
now.prebuild.mjs	Auto-updated file containing complete information about dependencies and their versions. This file is only available with the ServiceNow SDK.
package-lock.json	Auto-updated file containing complete information about dependencies and their versions. This file is only available with the ServiceNow SDK.
package.json	File containing information about your application and custom or third-party module dependencies. The `package.json` file must be in the base directory for an application. On an instance, the `package.json` path is specified in the **Package JSON** field of the custom application record [sys_app] in the format `<scope>/<package-name>/<version>/package.json`.

Create an app in source code

Create an application to develop in source code in ServiceNow Studio.

Before you begin

Note: You can use Build Agent to help you create and edit applications in ServiceNow Studio. For more information, see Build Agent.

Role required: admin

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Select the Explorer tab icon in the activity bar and add an app in one of two ways.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. If there are no applications in the Explorer yet, select **Create an app**.

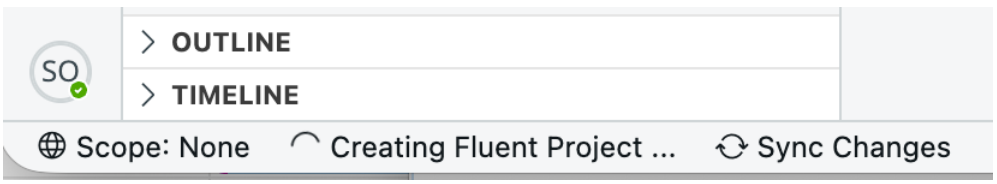
b. If there are apps in the Explorer, select the add to workspace icon and select **Create an application**.

A prompt box opens at the top of the screen.

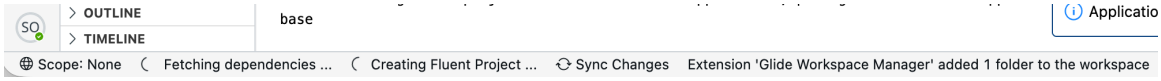
3. Select the type of application to create.
   - Scoped: Create a scoped application that is protected by identifying and restricting access to application files and data.
   - Global: Create an application in the global scope to allow it to be accessible to other global applications.
4. Enter a name for the application and press Enter.
5. Enter a description for the application and press Enter.
6. For scoped applications, enter a scope name and press Enter.
   The scope name should fill automatically from the application name. The scope name must be unique on the instance, begin with x_<prefix>, and be 18 characters or fewer. For more information, see Namespace identifier.

   If you receive an error because the scope name is not unique, begin the process again from the beginning.
7. Enter a package name for the application and press Enter.
   The package name should fill automatically from the application name. The package name must adhere to Node Package Manager (npm) package naming standards.
8. Select a template that defines the default application structure and press Enter.
   - Basic now-sdk boilerplate: An application with only the basic structure necessary for development in source code.
   - now-sdk + AIUX Extension starter: An AIUX extension application that overlays pages onto an existing host experience.
   - JavaScript now-sdk + basic: An application configured for development in ServiceNow Fluent and JavaScript.
   - JavaScript now-sdk + fullstack React: An application configured for development in ServiceNow Fluent, JavaScript, and React.
   - TypeScript now-sdk + basic: An application configured for development in ServiceNow Fluent and TypeScript. TypeScript source files in the `src/server` directory are transpiled into JavaScript modules.
   - TypeScript now-sdk + fullstack React: An application configured for development in ServiceNow Fluent, TypeScript, and React. TypeScript source files in the `src/server` directory are transpiled into JavaScript modules.

In the status bar at the bottom of your browser, watch the progress of the app as it builds.



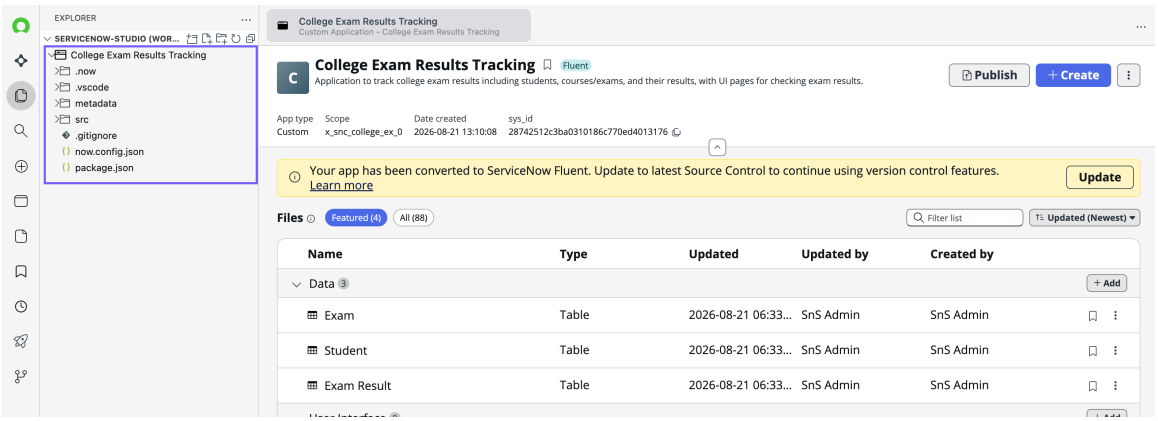
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.





Result

An application with the default application structure is added to the instance and open in the Explorer. For information about the application structure, see the Application Structure section of the Building apps in source code in ServiceNow Studio topic.



Trouble?

In the status bar, a message confirms whether the application was created. If the application creation fails, review the output logs in the panel.

Note: For ServiceNow Studio to install the required dependencies in an application, the public npm registry must respond with the HTTP `Access-Control-Allow-Origin` header.

What to do next

From your Git provider, create a dedicated Git repository for the application. Initialize a local Git repository for your application and push it to the remote repository. For more information, see Initialize a Git repository.

In ServiceNow Studio, start developing your application in source code with ServiceNow Fluent, writing custom JavaScript modules, or adding third-party libraries.

Convert an application to Fluent

Convert an existing application to support development in source code with ServiceNow Studio.

Before you begin

Role required: admin

About this task

Existing applications that weren't created in source code must be converted to support development. Converting an application adds the necessary files and directories for developing it in source code. You can choose whether to convert existing application metadata into ServiceNow Fluent code.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Important: Only admins can work on Fluent apps. Don't convert the app if you want delegated developers to continue working on the application.

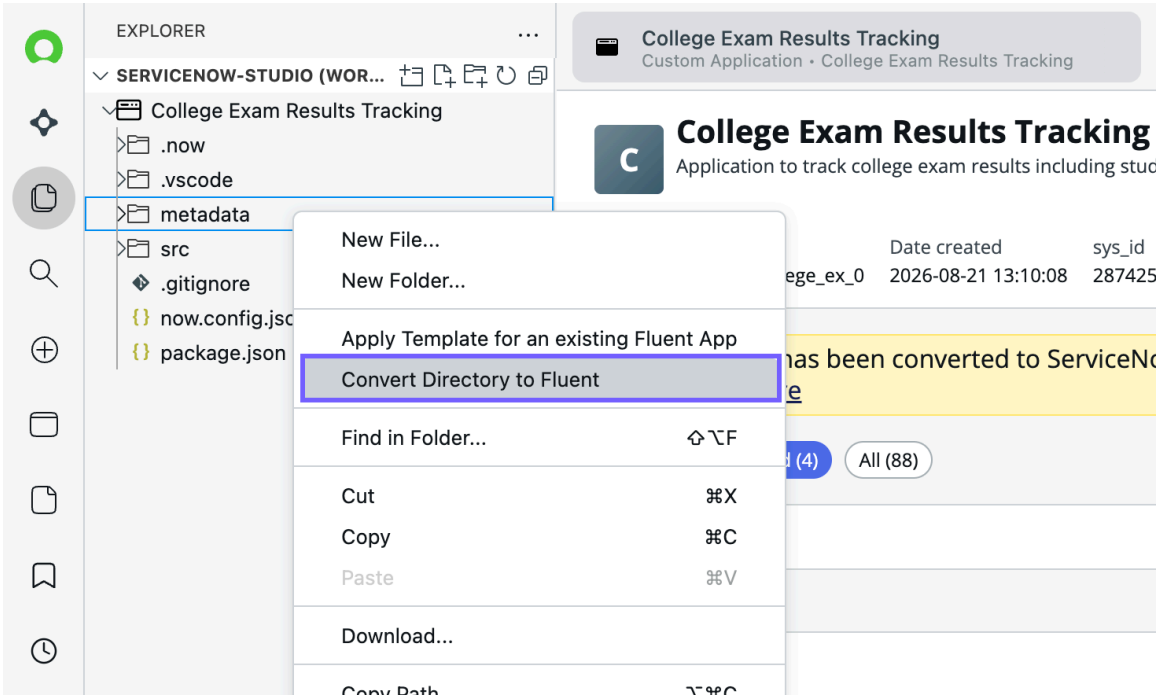
Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Open the application you want to convert from the Apps tab .
3. Select **App details**.
4. In the **Convert app to ServiceNow Fluent** message, select **Convert**.
5. Select **Convert**.
   The application is added to your workspace with the default application structure, but the application metadata isn't converted into ServiceNow Fluent code. For more information, see Building apps in source code in ServiceNow Studio.
6. **Optional:** To convert existing metadata into ServiceNow Fluent code, complete the following steps.

   a. From the Navigator panel, select the Explorer tab icon .

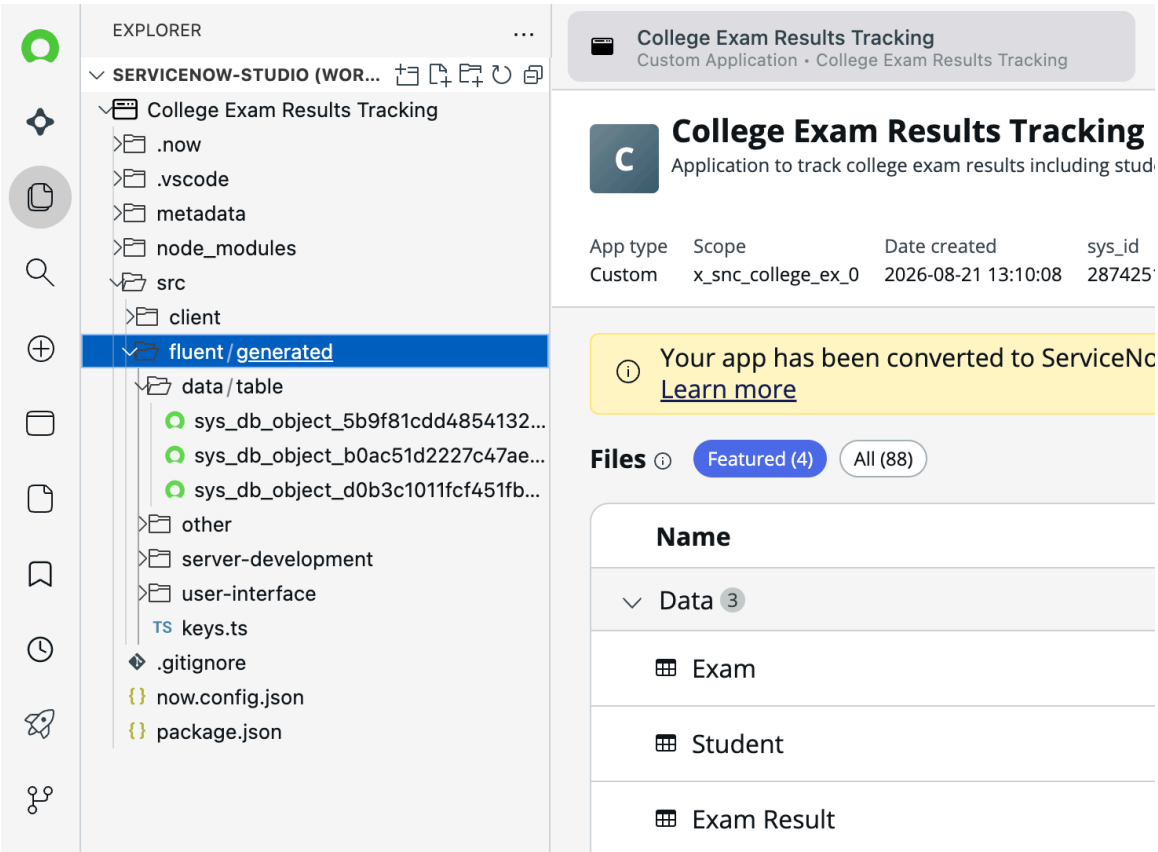
   b. Select and hold (or right-click) the `metadata` directory for the application and select **Convert Directory to Fluent**.

Example



Application metadata is defined in ServiceNow Fluent code in the `fluent/generated` directory and removed from the `metadata` directory and its sub-directories.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Note: A limited number of metadata types, such as Metadata Snapshots [sys_metadata_link] and UX Assets [sys_ux_lib_asset], can't be represented as ServiceNow Fluent code and aren't transformed. These metadata types remain as metadata XML files in the `metadata` directory of your application.

7. **Optional:** Build and install your application to compile source code into application metadata and make your changes available across the instance.
   For more information, see Build and install a Fluent app in ServiceNow Studio.

Result

The converted application is added to your workspace with the necessary files and directories to support development in source code. After installing a converted application, the **Package JSON** field of the custom application record [sys_app] contains the path to the `package.json` file for the application.

Trouble?

Note: For ServiceNow Studio to install the required dependencies in an application, the public npm registry must respond with the HTTP `Access-Control-Allow-Origin` header.

What to do next

From your Git provider, create a dedicated Git repository for the application. Initialize a local Git repository for your application and push it to the remote repository. For more information, see Initialize a Git repository.

Continue editing your application in ServiceNow Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Build and install a Fluent app in ServiceNow Studio

Build an application to compile its source code and install application changes across an instance from ServiceNow Studio.

Before you begin

Create, convert, or clone an application and add it to the Explorer. For more information, see Create an app in source code.

Role required: admin

About this task

When you build an application, ServiceNow Fluent code is compiled into application metadata, and modules in the application are added to the EcmaScript Module [sys_module] table on the instance.

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. From the Status bar, select **Sync Changes** to synchronize changes made to the application metadata from across the instance.
   Synchronizing an application confirms you have the latest changes before making additional changes. Only the changes made since the last time the app was synched are downloaded and transformed. For more information, see Synchronizing Fluent apps in ServiceNow Studio.
3. Select the Explorer tab, and modify your application in one of the following ways.
   - Define application metadata in code with ServiceNow Fluent in the ServiceNow IDE
   - Create and use JavaScript modules in applications in the ServiceNow IDE
   - Use third-party libraries in applications in the ServiceNow IDE
4. From the Status bar, select **Build and Install**.
   The active file that's open in the editor determines which application to build. If no files are open, select the application to build when prompted.

   Trouble?
   In the status bar, you can see a message that confirms whether the build was successful or failed. If the build fails, review the output logs and any problems in your code from the panel.
5. **Optional:** Review your changes as application metadata.

   a. From the Navigator panel, select the File Categories or Apps tabs.

   b. Open the metadata files that you changed and review your changes from the embedded ServiceNow AI Platform user interfaces.

   Tip: To see your changes in source code and metadata side by side, select the show XML icon  .

Result

The application is updated across the instance to reflect your changes. Other users can modify the application metadata simultaneous with modifications to the source code. Also, other developers can reuse module code in other modules or scripts within an application.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Trouble?

Confirm that the metadata on the instance matches the metadata in the installation package by uninstalling and reinstalling the application on the instance using the `Fluent: Reinstall Fluent App` command. For more information, see ServiceNow IDE commands.

Warning: Changes to metadata on the instance that haven't been synced into your application in ServiceNow Studio are removed.

What to do next

Use source control to stage, commit, and push your changes to a remote Git repository. For more information, see Using Fluent source control in ServiceNow Studio.

Synchronizing Fluent apps in ServiceNow Studio

Synchronizing an application in ServiceNow Studio downloads and transforms application metadata into ServiceNow Fluent code.

ServiceNow Fluent supports two-way synchronization, which allows changes to metadata to be synced from other ServiceNow AI Platform user interfaces into source code. When you build and install an application, changes to source code are compiled into metadata across the instance. This process enables collaborating with users of different skill sets because developers can modify the ServiceNow Fluent code while others modify the application metadata across the platform.

In ServiceNow Studio, when metadata in an application is changed, you're prompted to sync the application to get the latest changes. If metadata changes are detected in application, synchronizing is required before you can install the application across the instance.

Note: If you ignore the notification to sync an application, you aren't prompted to do so again for the remainder of the session. If you try to install the application or refresh the browser, you're prompted to sync again.

You can sync an application on demand from the status bar, or from the command palette with the `Fluent: Sync Fluent App with changed metadata` command.



Synchronization process in ServiceNow Studio

The following list explains the stages of the synchronization process for applications in ServiceNow Studio:

1. Before the first time an application is synced:
   - You should initialize a repository.
   - You should stage any changes to the application to retain them when the application is synced.
2. The first time that an application is synced:
   - New metadata that isn't defined in source code is transformed into source code in the `src/fluent/generated` directory.
   - Changes to metadata that is defined in source code are downloaded into source code in the `src/fluent` and `src/fluent/generated` directories.
   - Changes to existing metadata XML that hasn't been converted into source code are downloaded into the `metadata` directory.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

3. Subsequent times that an application is synced:
   - Only the changes made since the last time the application was synced are downloaded and transformed.
   - Optionally, you can synchronize all metadata in the application with the `Fluent: Force Sync of Fluent App with all metadata` command from the command palette. Before running this command, you should stage any changes to retain them.

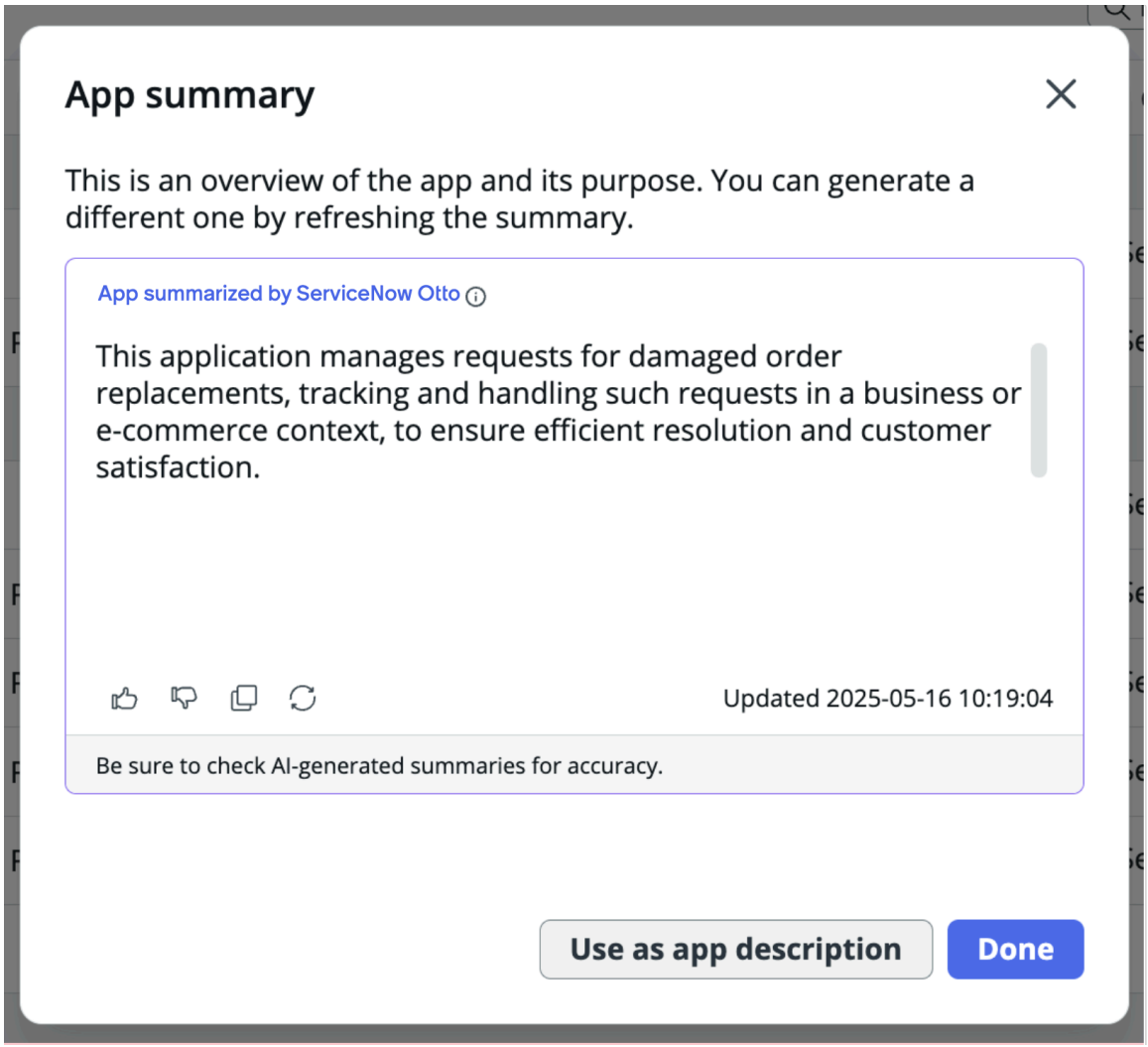
AI tools and files in ServiceNow Studio

Use AI tools and files in ServiceNow Studio to create and summarize apps using generative AI. You can also add AI files to applications to enhance user productivity through conversation and proactive experiences.

The main AI tools available in ServiceNow Studio are the ServiceNow Otto for Creator app summary generation skill and Build Agent. Use app summary generation to create short descriptions of existing apps. For more information about Build Agent, see Build Agent in ServiceNow Studio.

App summary generation

After creating an app, use the app summary generation skill to create a summary of the contents and purpose of the app. Rate or update the summary and use it as the app description. Accurate app descriptions support app ecosystem governance and help identify duplicate apps.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For more information, see App summary generation.

What AI file types can I add to an application?

Add three types of AI files to applications: agentic workflows, AI agents, and skills. These files open in a new tab for editing in AI Agent Studio. After creating or editing an AI file, return to the App details page and select the reload app details icon to capture the changes. For more information about AI file types, see ServiceNow Studio Navigator panel taxonomy.

Important: AI options in the ServiceNow Studio file taxonomy are only available if the AI components have been installed and configured.

Where can I find more information about AI capabilities?

The following ServiceNow resources cover AI capabilities, agentic workflows, AI Agents, and the ServiceNow Otto Skill Kit.

AI capabilities

- Exploring AI Admin Hub
- Build
- ServiceNow Otto for App Engine

Agentic workflows

- Create an agentic workflow
- Modify an agentic workflow

AI Agents

- Explore AI agents
- Create an AI agent
- Examples of using AI agents

ServiceNow Otto Skill Kit

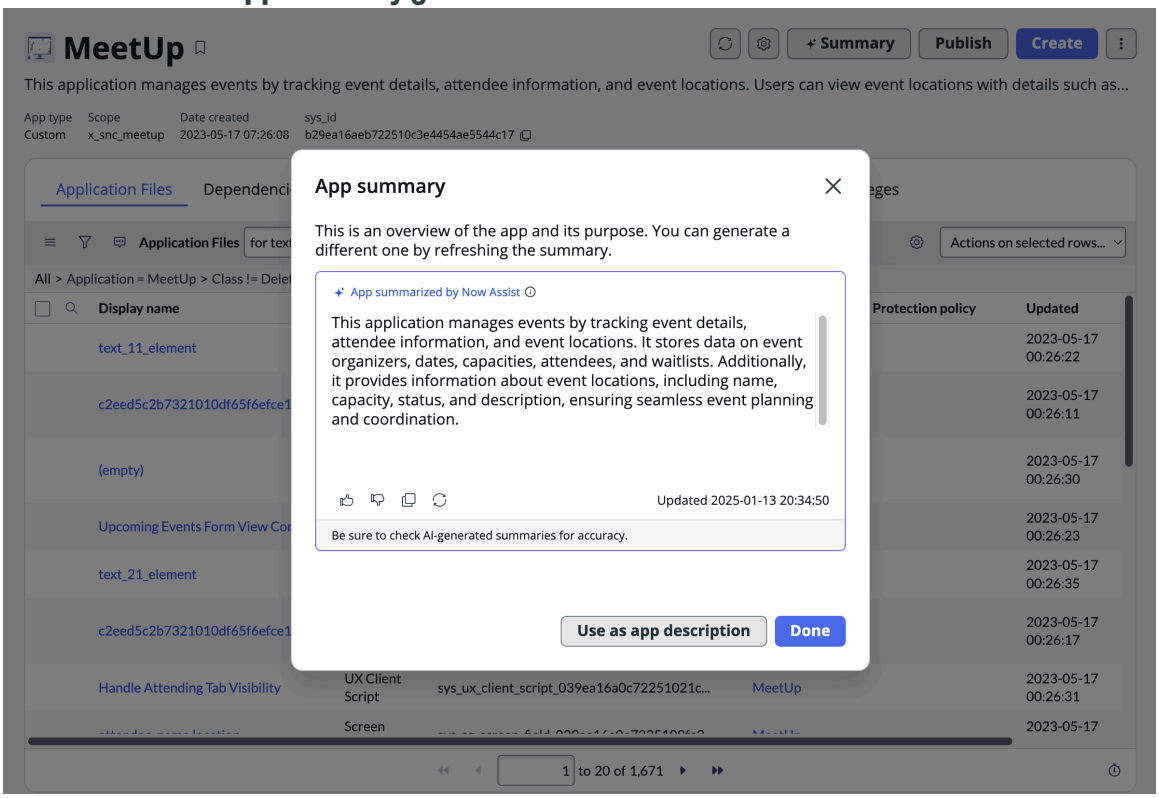
- Exploring AI Skill Kit
- Create a skill

App summary generation

Use the ServiceNow Otto for Creator application to use generative AI for summarizing an app. With a single button, ServiceNow Otto for Creator generates the app summary that you can then copy to the app description, or use to find duplicate apps.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ServiceNow Otto app summary generation



Get started

Explore Learn about summarizing apps	Configure Set up app summary generation
Use Generate a summary of your app	Reference Get details about properties, roles, and more

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Important:

- Not all model providers are available for customers with in-country SKUs, and some AI products/features are currently unavailable for in-country customers. For more information, see the KB1584492 article in the Now Support Knowledge Base. Be sure to check for model provider availability updates in future releases.
- Some AI products/features are currently unavailable for customers in the FedRAMP, NSC DOD IL5, or Australia IRAP-Protected data centers, self-hosted customers, or in other restricted environments. For more information, see the KB0743854 article in the Now Support Knowledge Base. Be sure to check for availability updates in future releases.
- Some AI products/features are currently available only for customers in some regions. Be sure to check for availability updates in future releases.
- Some AI products and skills are not available in Regulated Markets. For more information, see KB2593939: Regulated Markets AI Products/Skills Not Available. Be sure to check for availability updates in future releases.

Helpful resources

Some ServiceNow resources that can provide you with helpful information are:

ServiceNow Community

ServiceNow Community

Support

AI limitations

This application uses artificial intelligence (AI) and machine learning, which are rapidly evolving fields of study that generate predictions based on patterns in data. As a result, this application may not always produce accurate, complete, or appropriate information. Furthermore, there is no guarantee that this application has been fully trained or tested for your use case. To mitigate these issues, it is your responsibility to test and evaluate your use of this application for accuracy, harm, and appropriateness for your use case, employ human oversight of output, and refrain from relying solely on AI-generated outputs for decision-making purposes. This is especially important if you choose to deploy this application in areas with consequential impacts such as healthcare, finance, legal, employment, security, or infrastructure. You agree to abide by ServiceNow's AI Acceptable Use Policy, which may be updated by ServiceNow.

Data processing

This application requires data to be transferred from ServiceNow customers' individual instances to a centralized ServiceNow environment, which may be located in a different data center region from the one where your instance is, and potentially to a third-party cloud provider, such as Microsoft Azure. This data is handled per ServiceNow's internal policies and procedures, including our policies available through our CORE Compliance Portal.

Data collection

ServiceNow collects and uses the inputs, outputs, and edits to outputs of this application to develop and improve ServiceNow technologies including ServiceNow models and AI products. In addition, this application will collect information about applications (and associated application files) in which App generation was utilized. Customers can opt out of future data collection at any time, as described in the Now Assist Opt-Out page.

Exploring ServiceNow Otto for app summary generation

With the ServiceNow Otto for Creator application, you can generate a summary of an app. You can then copy the summary to the description for the app, and use it to check for duplicate apps.

ServiceNow Otto for app summary generation overview

ServiceNow Otto for app summary generation generates a summary of what the app does, which you can copy to the app's description by selecting the **Use as app description** button.

Additionally, ServiceNow Otto for app summary generation has the following features, which are accessed via their icons on the App summary modal:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Rate the summary with a thumbs up or thumbs down icon.
- Copy the summarized text to your clipboard with one click.
- Regenerate the summary if you want it worded differently.

ServiceNow Otto for app summary generation users

Note: In addition to the following personas, users must have the sn_app_summary.app_summary_user role to use ServiceNow Otto for app summary generation.

Users

User	Description
Delegated developer	Person who has been delegated to work on an app.
Admin	Person who can review and approve tasks related to custom application development.

ServiceNow Otto for app summary generation moves to an agentic architecture

App summary moves to an agentic architecture App summary now uses an AI agent to generate application summaries. This change moves app summary from a skill-based architecture to the AI agent orchestration model.

Previously, app summary depended on table summary and app summary skills. Administrators enabled these skills from the AI Skills admin console before users could generate application summaries.

With this release, users who install the App Summary plugin receive the App Summary AI agent. App summary no longer depends on dedicated skills to complete summary generation.

The App Summary AI agent is turned off by default after plugin installation to help avoid unexpected charges. An administrator must enable the agent in AI Agent Studio before end users can generate application summaries. For more information, see Enable the App Summary AI agent.

The end-user experience remains the same. The change affects only the underlying architecture, which now uses an agentic model instead of the earlier skill-based model.

ServiceNow Otto for app summary generation benefits

ServiceNow Otto for app summary generation benefits

Benefit	Feature	Users
Quickly generate a description of an app, which you can use to check for duplicate apps.	Summarize the contents of an app in ServiceNow Studio	Developer, admin

Configuring ServiceNow Otto for app summary generation

Enable the app summary generation skill in the ServiceNow Otto for Creator application so that you can get started with summarizing applications.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Install ServiceNow Otto for app summary generation

Install the ServiceNow Otto for Creator application so that you can use app summary generation for your organization.

Before you begin

Review the ServiceNow Otto for Creator application listing in the ServiceNow Store to learn about dependencies, licensing, and subscription requirements, and release compatibility. ServiceNow Otto for Creator installs the ServiceNow Otto for app summary generation skill.

Note: The ServiceNow Otto table summary generation skill must also be active to use the app summary generation skill.

Role required: admin

Procedure

1. Search for and open the ServiceNow Otto for Creator application.
2. On the ServiceNow Otto for Creator application page, select **Get**.
3. After your request is approved, navigate to **All** > **System Applications** > **All Available Applications** > **All**.
4. Find and select the ServiceNow Otto for Creator application (sn_now_creator) by using the filter criteria and search bar.
5. Select **Install**.
6. Verify that ServiceNow Otto for Creator is installed.
   a. Navigate to **All** > *AI Admin Hub* > **Features**.
   b. In the workflow list, select **Creator**.
   c. Verify that the app summary generation skill and the table summary generation skill are active by selecting **View details** on the **App** card.

   For more information about using the Now Assist Admin console to access information about setting up, configuring, and monitoring Now Assist applications, see Now Assist Admin console.

What to do next

Grant the admin and now.assist.creator roles, or the sn_g_app_creator.app_creator and now.assist.creator roles, to each user that you want to summarize apps.

To summarize an app, see Summarize the contents of an app in ServiceNow Studio.

Enable the App Summary AI agent

Enable the App Summary AI agent to generate application summaries in ServiceNow Studio.

Before you begin

The App Summary plugin must be installed.

Role required: admin

About this task

After the App Summary plugin is installed, the App Summary AI agent is disabled by default. This default helps avoid unexpected charges, since using the agent is associated with a charge. Enable the agent to make app summary available to end users.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to AI Agent Studio.
2. Select the **AI Agents** tab.
3. Select **App Summary AI Agent**.
4. Select the **Select channels and status** tab.
5. In the **Activation status** section, enable the **Active** toggle.
6. Select **Save and test**.

What to do next

Open an application in ServiceNow Studio to generate an app summary.

Summarize the contents of an app in ServiceNow Studio

Generate a summary of your app using ServiceNow Otto for Creator in ServiceNow Studio. After reviewing the summary, you can use it as a description for your app.

Before you begin

Role required: sn_app_summary.app_summary_user, and admin or delegated_developer with either Delegated admin or Delete application permissions

About this task

If your company has a robust citizen development program and has many developers creating apps, you might have some duplicate apps. Good app descriptions are one of the best and easiest ways to be able to scan your list of apps for duplicates. Using ServiceNow Otto to generate a good app description can enable you to have more governance over app creation.

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Open an application.
3. On the **App details** page, select **Summarize**.

   The App summary modal shows that an app summary is being generated, which may take a few moments.

   The app summary generation times out after two minutes by default. If this occurs, you will receive an error notification. An administrator can adjust the timeout duration.

4. After the app summary appears, review it for accuracy and take one of the following actions.

Action	Result
Rate the summary as helpful	Rating summaries helps enable ServiceNow Otto to create better summaries.
Rate the summary as unhelpful	Rating summaries helps enable ServiceNow Otto to create better summaries.
Copy to clipboard	Copy the summary to use it elsewhere in your app development.
Regenerate summary	Regenerate the summary if the initial attempt wasn't accurate or needed major changes.
Use as app description	If the summary is complete, accurate, and helpful, select **Use as app description**, and

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Action	Result
	the description is populated with the summary.

5. Select **Done**.

Result
After you create an initial app summary, the **Summarize** button becomes the **Summary** button. If you see **Summary**, know that an app summary has already been created and can be edited as needed.

ServiceNow Otto for app summary generation reference

The following roles are required for use with the ServiceNow Otto for Creator app summary generation skill.

Roles

Grant the admin and now.assist.creator roles, or the sn_g_app_creator.app_creator and now.assist.creator roles, to each user that you want to summarize apps.

Change your development experience in ServiceNow Studio

Switch between development environments in ServiceNow Studio using the experience switcher to use the best tool for each stage of your app development.

Before you begin
Role required: admin or delegated_developer

About this task
Use the experience switcher when you need to continue app development in a different environment. For example, create an app in Creator Studio and continue working on it in ServiceNow Studio. The experience switcher provides access to Creator Studio, ServiceNow Studio, and ServiceNow IDE.

Important:

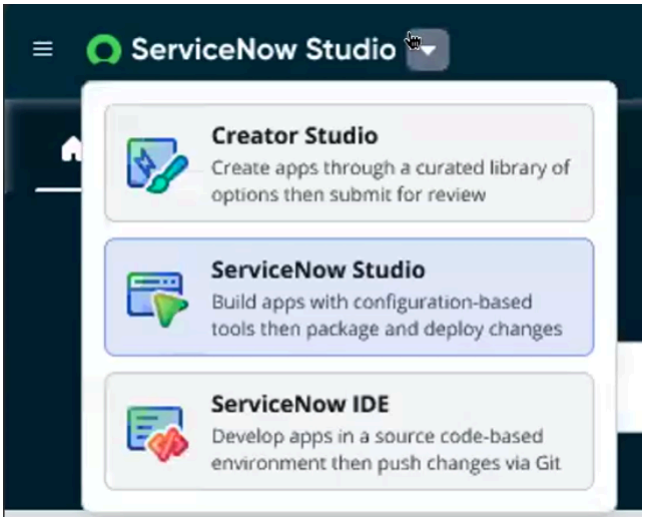
As of version 30.1.1, the experience switcher has been removed from ServiceNow Studio. Each development tool can still be accessed separately through the ServiceNow AI Platform.

As of version 29.2.11, ServiceNow IDE capabilities have been integrated into ServiceNow Studio. For more information, see Building apps in source code in ServiceNow Studio.

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Display the experience switcher by selecting the arrow next to the ServiceNow Studio name.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



3. Select the development environment you want to open.

> **Note:** The development experiences that you can select from depend on your permissions. Contact your administrator if you need assistance.

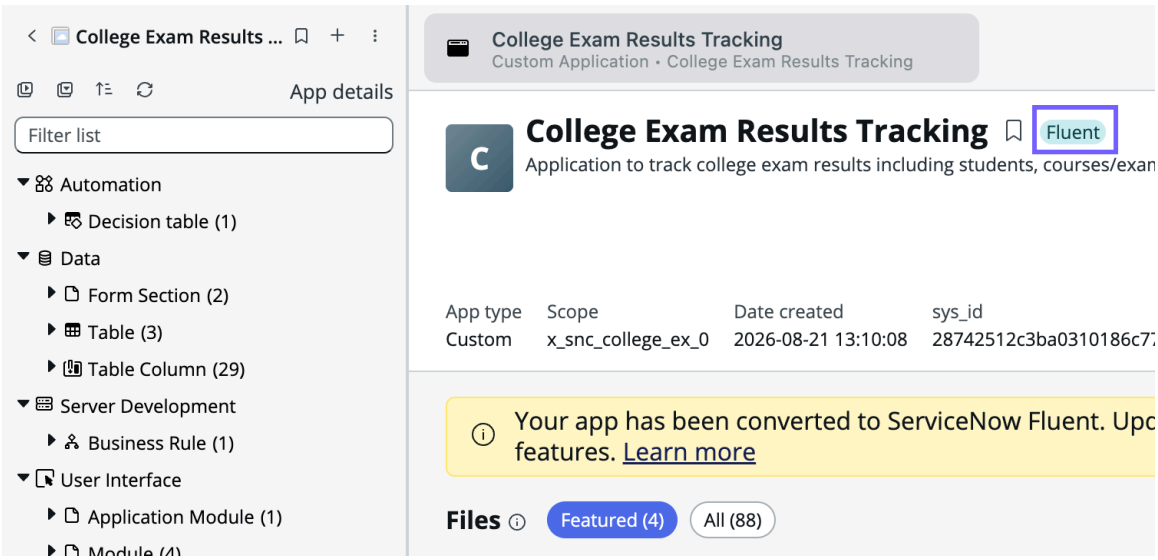
Result
The new development environment opens and you can begin working. For more information, see Creator Studio and Developing applications with the ServiceNow IDE.

Viewing app origination information in ServiceNow Studio

View app origination information on the App details page in ServiceNow Studio to determine where an app was created and which environment to use for editing and deployment.

What does the App details page show?

Each App details page shows whether the app was created in Creator Studio or created in ServiceNow IDE and converted to Fluent. Apps created in ServiceNow Studio do not display a badge. This information determines whether the app can continue to be edited and deployed through ServiceNow Studio, or whether it must be opened and deployed in a different environment.

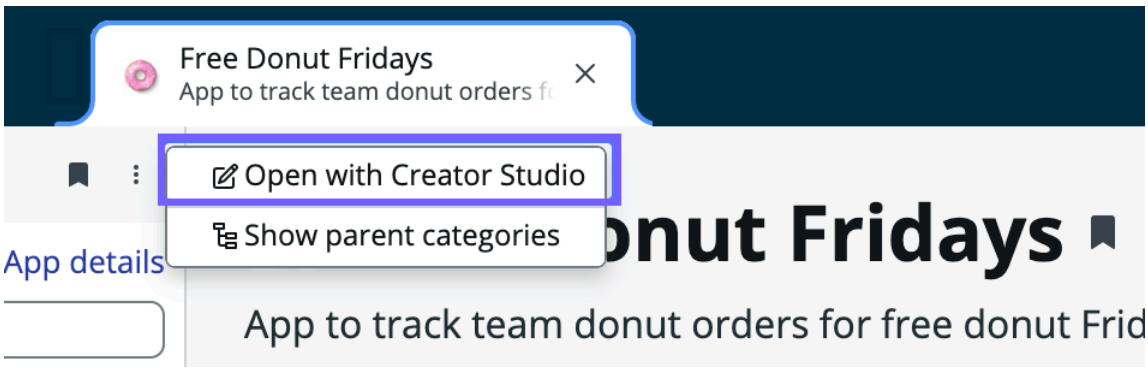


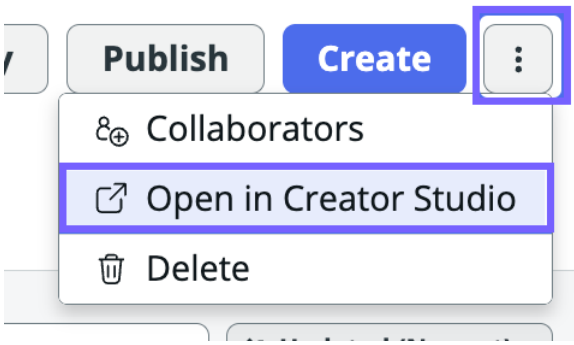
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

How do I edit an app created in Creator Studio?

Only apps created in Creator Studio can be edited in Creator Studio. Creator Studio supports only a subset of the metadata supported in ServiceNow Studio.

Open an app in Creator Studio by selecting the more options icon ⋮ on the Navigator panel or the App details page and selecting **Open with Creator Studio**. For more information, see Building apps with Creator Studio.





How do I deploy an app created in ServiceNow IDE?

Apps created in ServiceNow IDE can be edited in ServiceNow Studio.

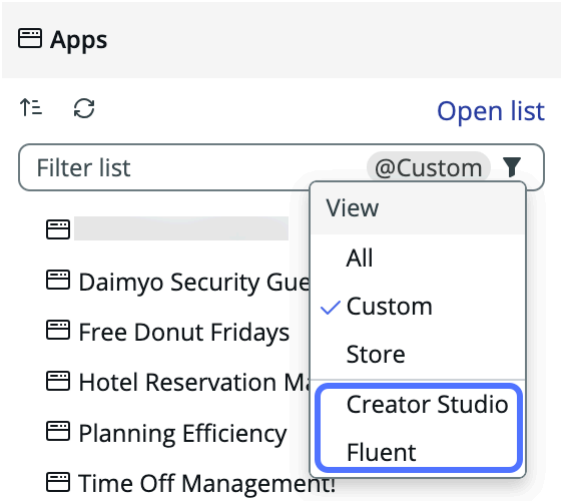
Important: Apps created in ServiceNow IDE or converted to Fluent cannot be deployed from ServiceNow Studio. Deploy these apps using the ServiceNow IDE.

When you see the **Fluent** badge on an app, follow the normal app deployment process used in the ServiceNow IDE. For more information, see Developing applications with the ServiceNow IDE.

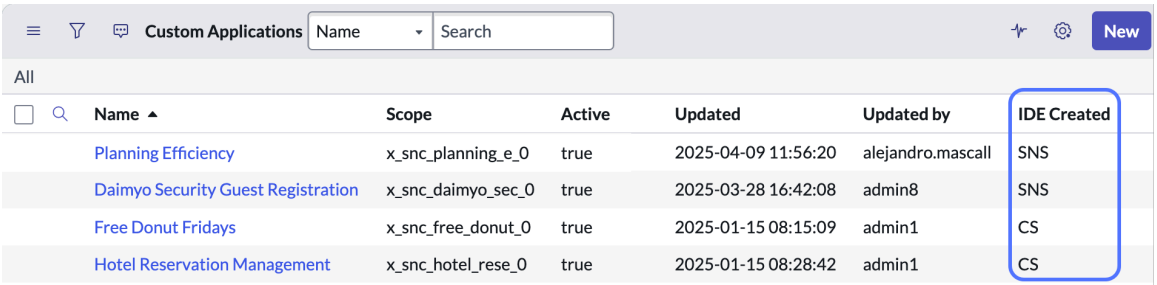
How do I filter apps by origination environment?

Access Creator Studio or Fluent apps using the filter in the Apps section of the Navigator panel. Select the **Creator Studio** or **Fluent** options to display only those apps.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



For more filtering options, select **Open list**. Use the classic UI16 list sorting and filtering options to find the apps you need. The **IDE Created** column shows which internal development environment the app was created in.



Name	Scope	Active	Updated	Updated by	IDE Created
Planning Efficiency	x_snc_planning_e_0	true	2025-04-09 11:56:20	alejandro.mascall	SNS
Daimyo Security Guest Registration	x_snc_daimyo_sec_0	true	2025-03-28 16:42:08	admin8	SNS
Free Donut Fridays	x_snc_free_donut_0	true	2025-01-15 08:15:09	admin1	CS
Hotel Reservation Management	x_snc_hotel_rese_0	true	2025-01-15 08:28:42	admin1	CS

Debug a script in ServiceNow Studio

Use the Script Debugger in ServiceNow Studio to debug business rules and other synchronous server-side scripts during app development.

Before you begin

Role required: admin or delegated_developer

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. On the home page, enter `Script Debugger` in the main search.
3. Select the **Script Debugger** module.
4. Use the Script Debugger to debug scripts.
   Some tabs and features are role-dependent. For more information, see Script Debugger and Session Log.

Metadata record categories in the ServiceNow Studio Navigator

Use ServiceNow Studio to access and work with all metadata types that the ServiceNow AI Platform supports, from automations and integrations to user interface files.

How do I access metadata in the Navigator panel?

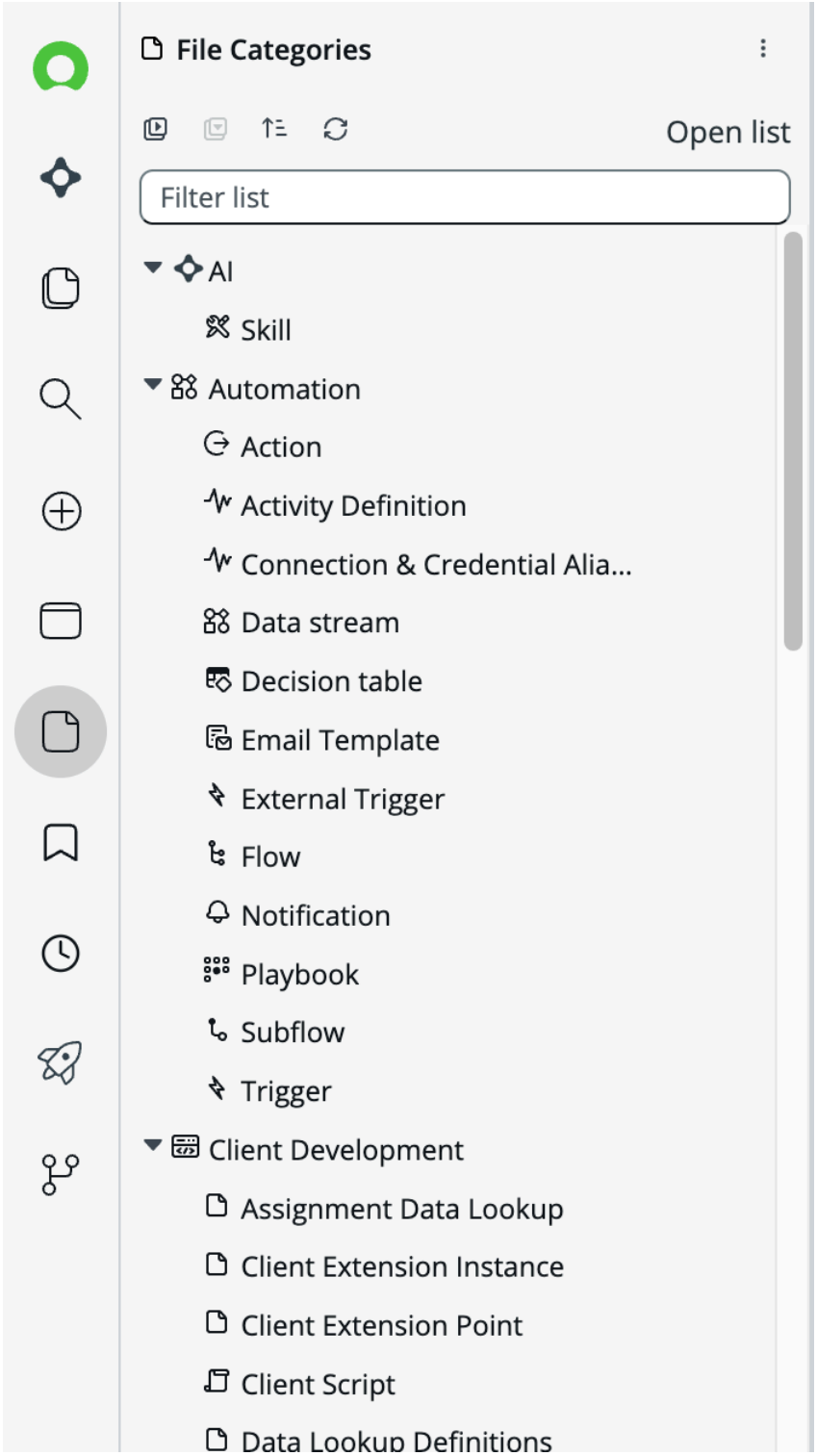
The **File Categories** section of the Navigator panel displays all metadata types accessible to the current user, arranged by taxonomy category. Expanding each metadata file category shows

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

its subcategories. For example, expanding the **Automation** category displays all automation metadata sub-types, such as actions, flows, and playbooks.

Note: For a complete list of each category's file types, see ServiceNow Studio Navigator panel taxonomy.

Metadata sections in the File Categories section of the Navigator

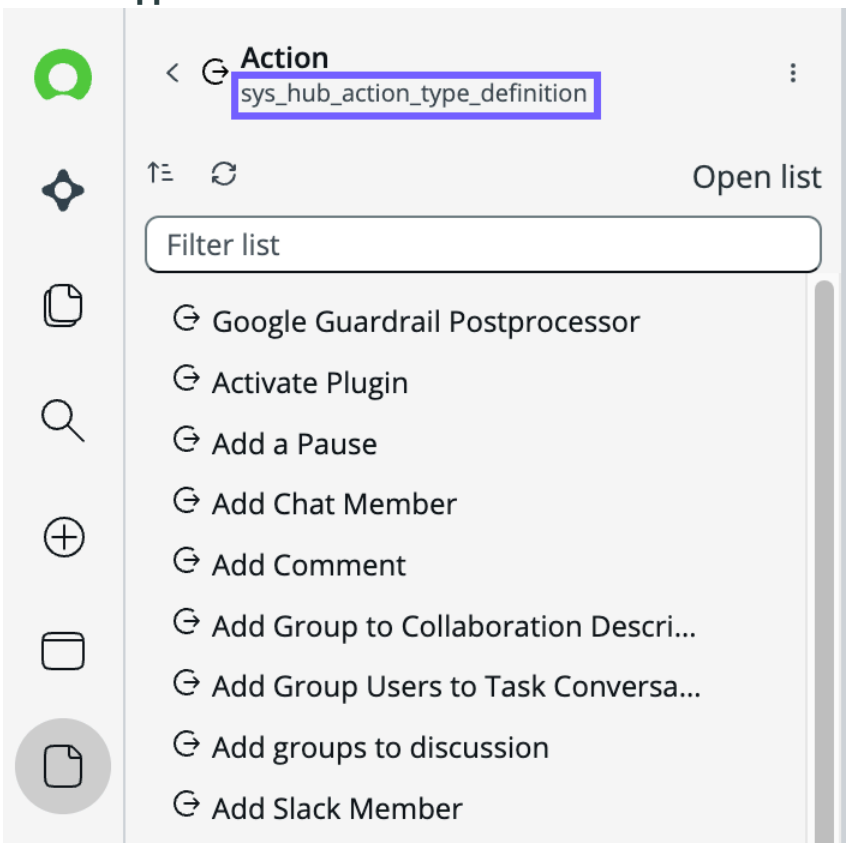


Selecting a sub-type of an expanded file category displays all metadata of that type. For example, selecting **Flow** under the **Automation** file category displays a list of all accessible flows. Select one to open it in Workflow Studio in a tab in ServiceNow Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: The source table for each sub-category of metadata file type appears when you select the category. Select **Open list** to see a complete list of all the files of that type.

Table for app files



For more information on using the Navigator panel to find app files by metadata category, see Find an app or app file using the Navigator panel.

How does metadata relate to platform tables?

Metadata lives as application files in their corresponding tables on the ServiceNow AI Platform. The main metadata table is sys_metadata, but specific types of metadata extend the sys_metadata table. For example, flows and sub-flows use the extended sys_hub_flow table.

For a list of available metadata and corresponding tables that extend sys_metadata, see ServiceNow Studio Navigator panel taxonomy.

What metadata is available by category?

Each file category contains different metadata types for automations, data, integrations, mobile app builder, UI, and other categories. For a list of each file type, its primary table, primary builder experience, and more, see ServiceNow Studio Navigator panel taxonomy.

Note: Group or ungroup the metadata categories in the File Categories menu by selecting the more options icon ⋮ and selecting or deselecting **Show parent categories**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Categories of metadata

Category	Metadata sections
AI	• Agentic Workflow • AI Agent • Skill
Automation	• Automation • Schedules
Data	• Client development • Data • MID Server • Server development
Integrations	• Inbound integration • Outbound integration • Properties
Mobile	• Mobile App Builder • Mobile Card Builder
UI	• Content • UI Builder • User interface
Other	• Natural Language Understanding (NLU) • Reporting • Security

Managing application and record changes

All changes made to applications or metadata records display on the Changes tab of the activity bar in ServiceNow Studio. View and manage changes made to applications tracked in update sets or attached to a Git repository.

Working with different types of applications

Manage custom apps you create in ServiceNow Studio through update sets or by connecting them to a Git repository. Applications can be converted to ServiceNow Fluent and manage the app through Fluent source control.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For more information, see Update sets in ServiceNow Studio and Convert an application to Fluent.

You can customize or extend out-of-the-box/base system ServiceNow applications to function to your specifications. However, because they contain proprietary code, all base system applications must be managed through update sets and XML. Base system apps cannot be converted to Fluent source code or managed through a Git repository.

Important: Only admins can work on Fluent apps. Don't convert the app if delegated developers must continue working on the application.

Update set and Git repository operations

Change operations for apps and metadata records can be performed from the Changes tab. When working with Build Agent, changes from each checkpoint are tracked in an update set. If you want to create a batch update set, select the **Batch** option in the Changes tab, and the platform record for the batch opens in the canvas. For more information about all update set operations, see Update sets in ServiceNow Studio.

Operations available for apps and records connected to Git differ depending on whether the app has been converted to Fluent. For applications and records converted to Fluent, see Using Fluent source control in ServiceNow Studio. For non-Fluent apps and metadata records, see Work with changes in Git.

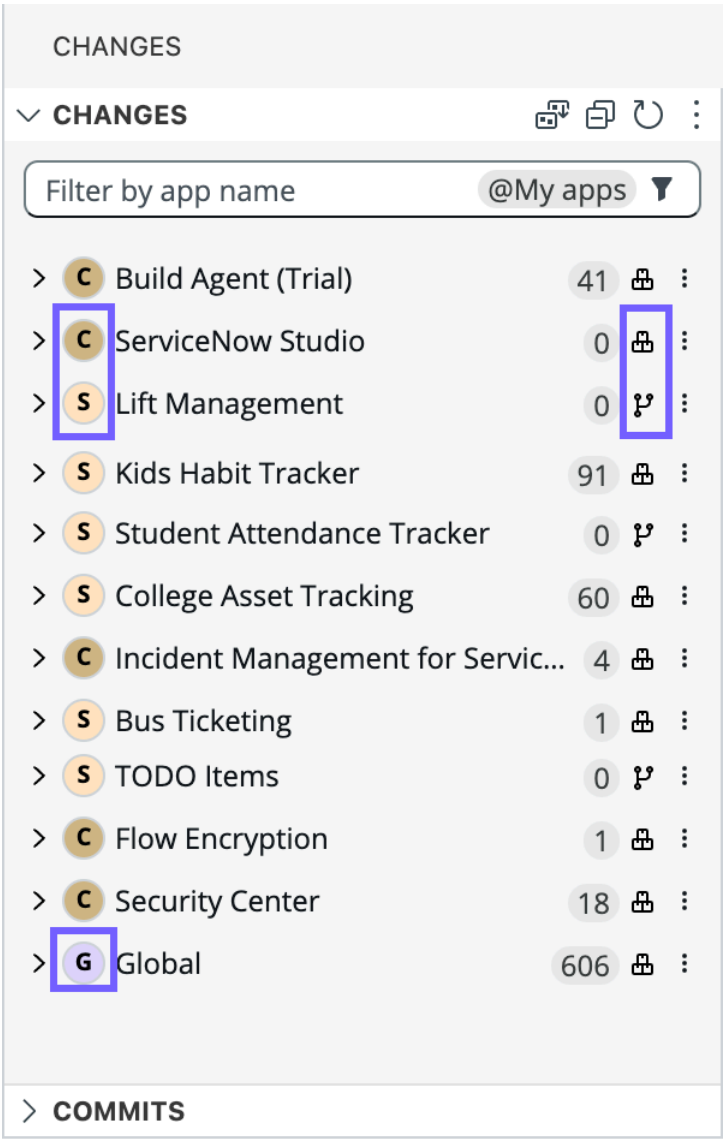
Note: Changes tracked in metadata source control display as update sets, not Git source control files. For more information, see Metadata source control in ServiceNow Studio.

Working with changes

The Changes tab contains information to help you identify changes.

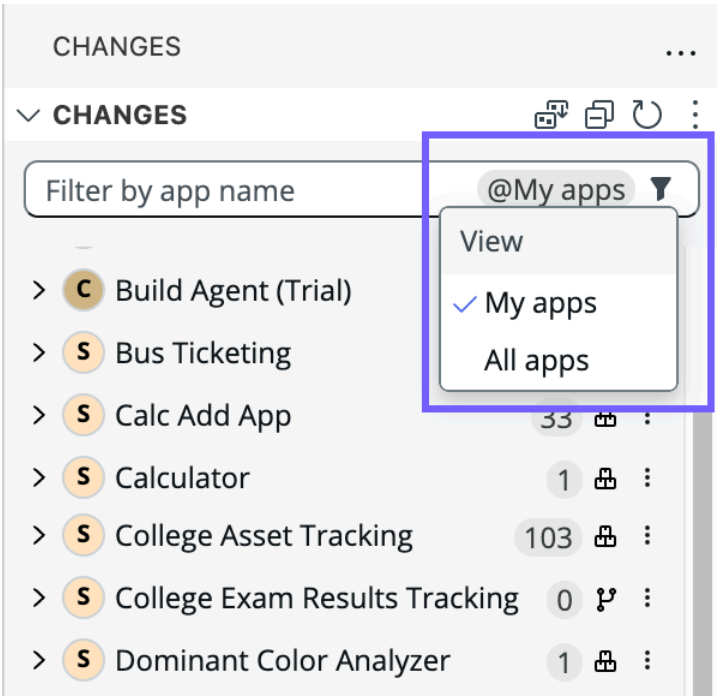
- **C**: Out-of-the-box/Store apps
- **S**: Custom Scoped apps
- **G**: Global apps
- : Changes tracked through a Git repository
- : Changes tracked through update sets

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



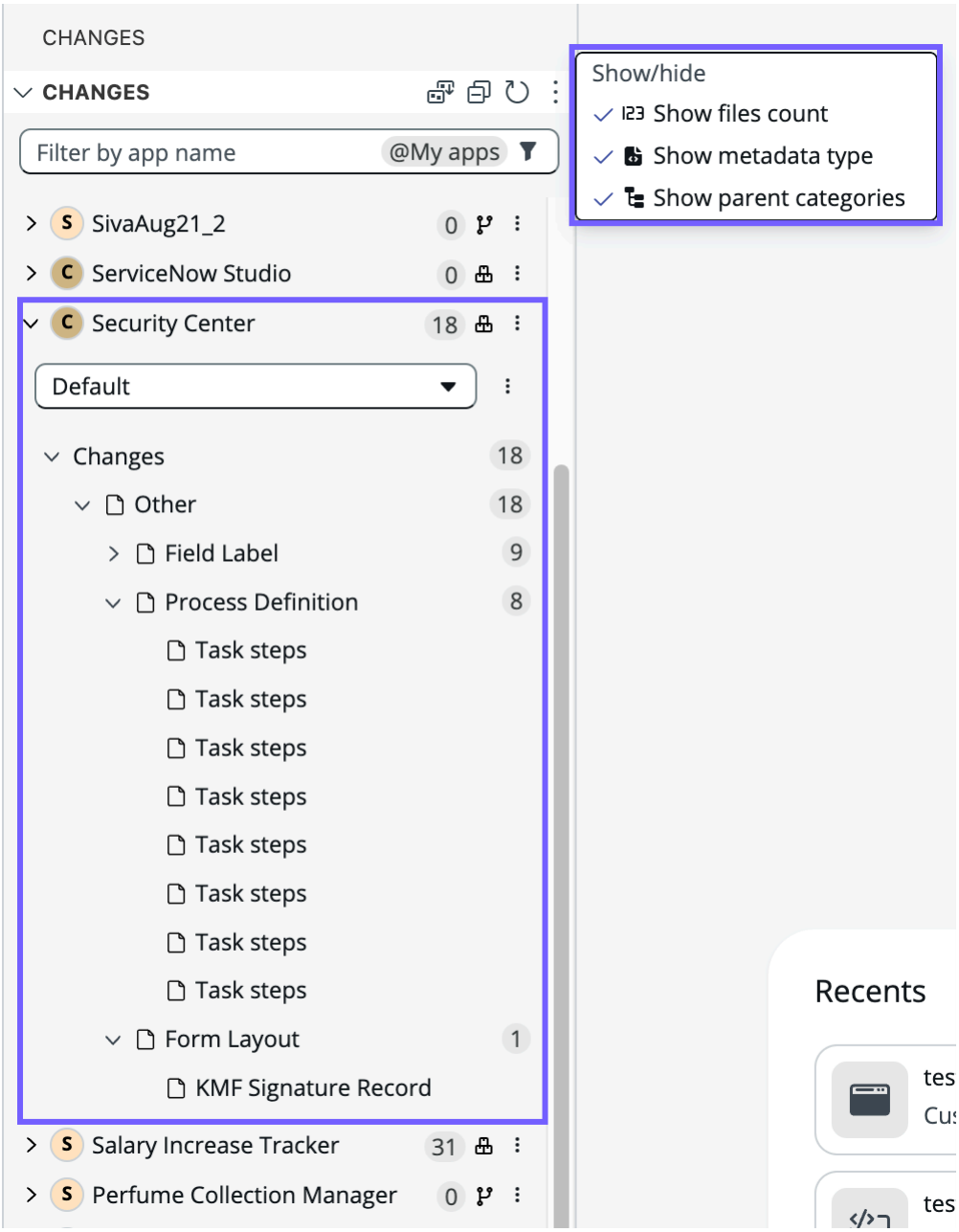
Filter the list of changes by selecting **My apps** or **All apps**. All changed apps appear in the Changes tab, whether tracked through update sets or Git source control.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



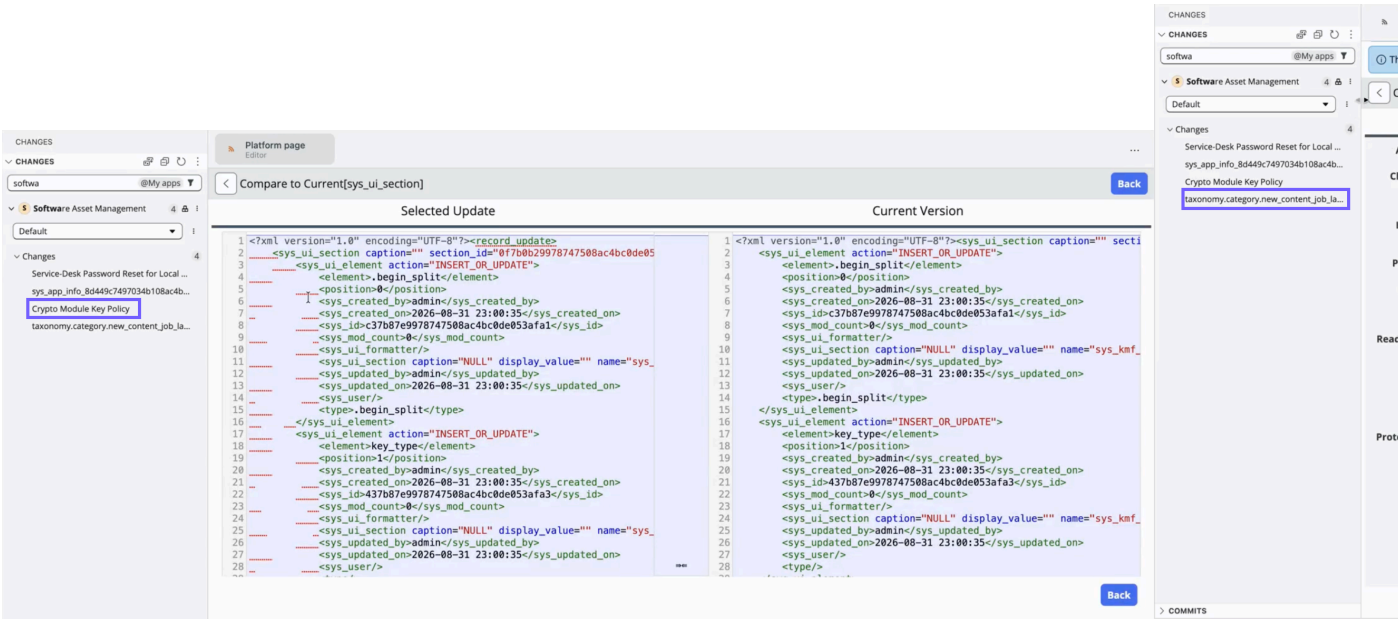
Select the more actions icon ⋮ to access options for showing and hiding file count, metadata category, and parent categories. Select one or all of the options to see more information about each file. Clear all of the options to hide categorizations and just see a list of files.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



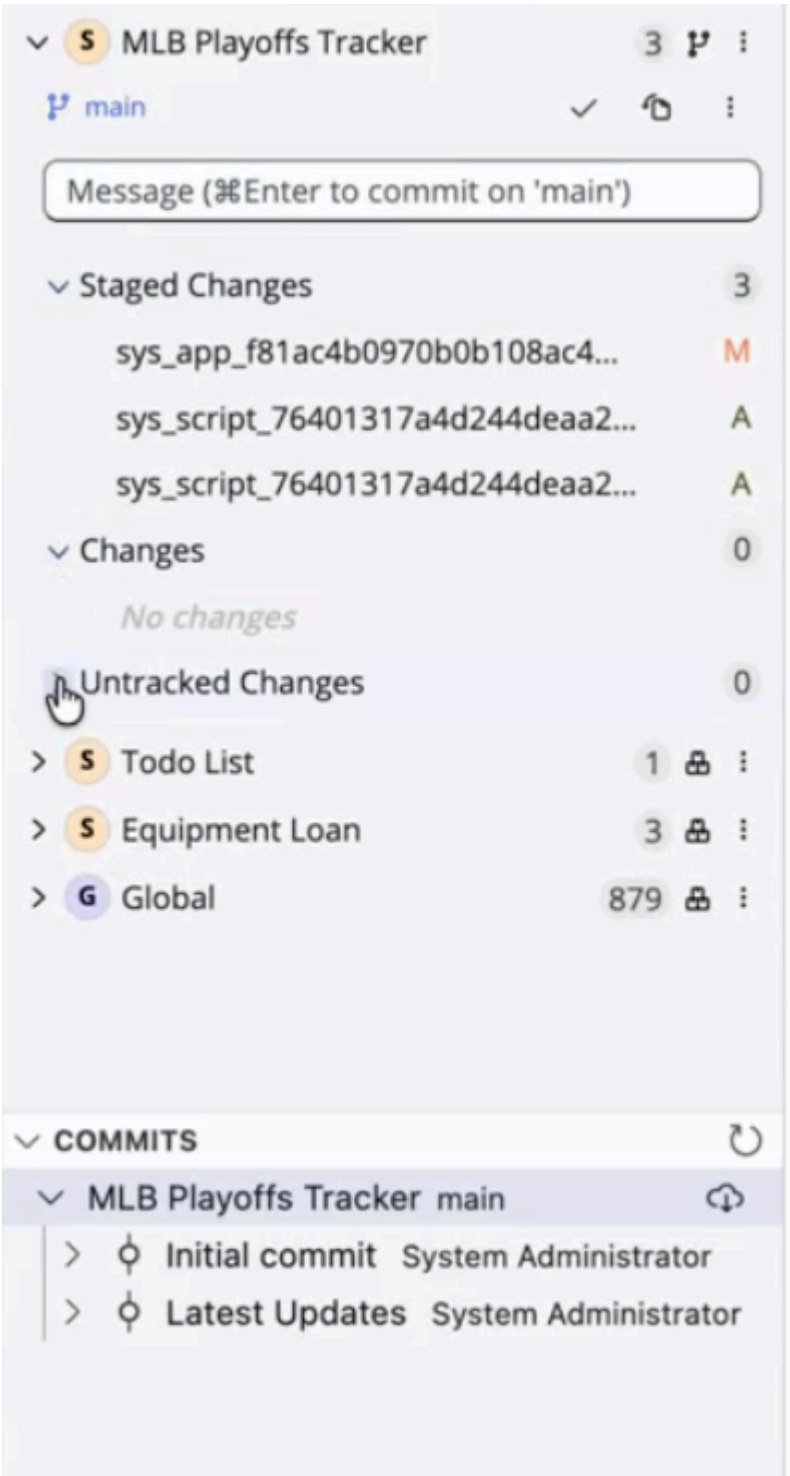
Selecting a changed record tracked under an update set opens it in the canvas with a code or metadata diff viewer. The diff view depends on the type of file that you select.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Selecting an application connected to Git shows staged changes, untracked changes, and recent commits.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

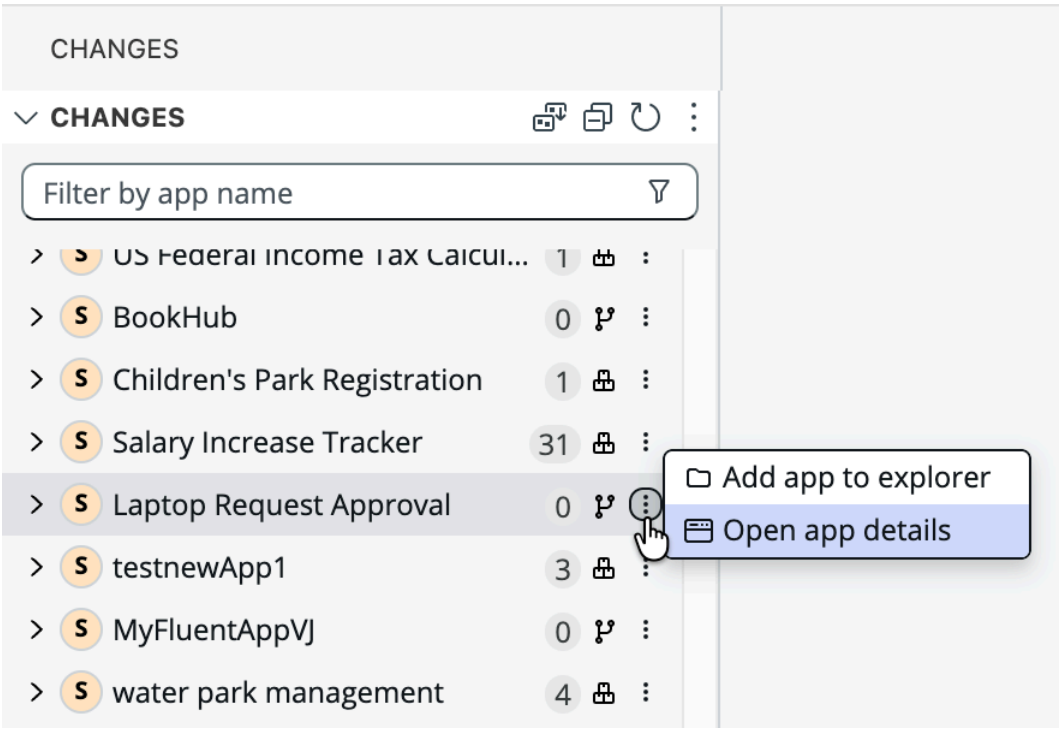


Use the more actions icon on each entry in the Changes tab to access additional options. For applications and records connected to Git, you can add the app to the Explorer and open App details.

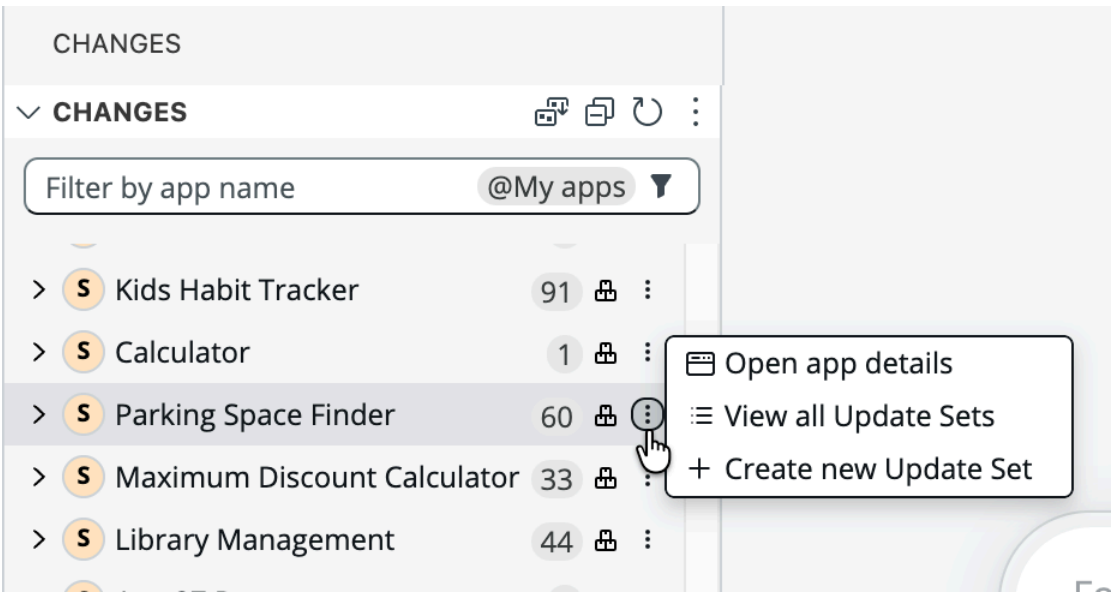
Note: Selecting **Add app to explorer** can produce different results depending on the following scenarios:

- If an app is not a Fluent app, then it is converted to Fluent and added to the Explorer.
- If an app is a Fluent app but is not in the Explorer, it is added to the Explorer.
- If an app is a Fluent app and is in the Explorer, the Explorer tab is focused and a message indicates that the app is in the workspace.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



For apps and records tracked through update sets, you can open app details, view all update sets, or create an update set.



For more information, see Update sets in ServiceNow Studio.

App deployment in ServiceNow Studio

Deploy applications from ServiceNow Studio using several methods, including using deployment requests to move application and global file changes across instances. Deployment options depend on organizational preference and standards.

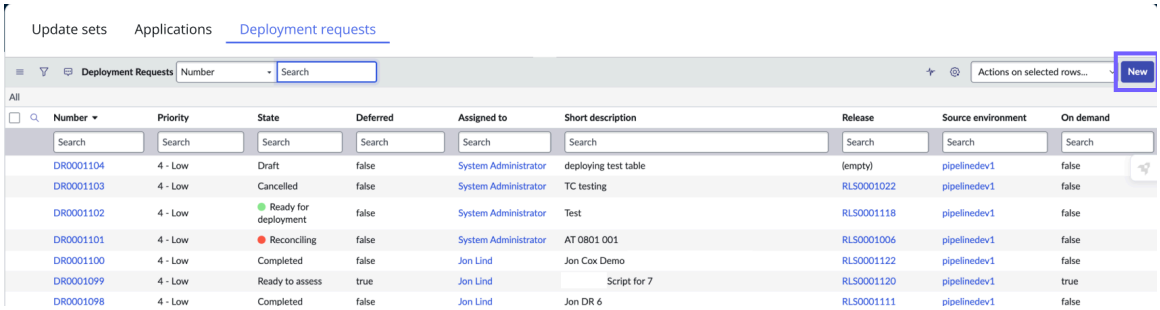
View and manage your deployment pipeline from within ServiceNow Studio, without switching to another application or environment.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

How are deployment requests created?

When a developer selects **Publish** in an application, a deployment request is created and displays on the **Deployment requests** tab in the Deployment tab.

Select a deployment record number to open it in a new tab, or select **New** to create a deployment request and attach update sets to it.



What options are available for app deployment?

Available deployment methods vary depending on if you're on the core UI or the Next Experience.

Next Experience

App Engine Management Center (AEMC): Deploy custom applications built in App Engine Studio, Creator Studio, and ServiceNow Studio across instances. AEMC uses Pipelines and Deployments or ReleaseOps to move applications between instances as they progress toward release. From AEMC, admins can customize pipelines, review and approve or reject deployment requests, and schedule deployments. For more information, see App Engine Management Center.

Core UI (lists and forms)

ReleaseOps: Automates the deployment process using update sets and customizable pipelines. ReleaseOps orchestrates the entire process of deployment, including moving changes between instances, running scans and tests, and releasing changes, while providing transparency into the process for release managers and teams. For more information, see ReleaseOps.

Application Repository: Central location for all scoped applications that are published by all ServiceNow customers. After you develop and test a custom application, you can make the application available to company instances by publishing it to this repository.

Update sets: Group of configuration changes that can be moved from one instance to another. Admins can group a series of changes into a named set and then move them as a unit to other systems for testing or deployment.

Update sets in ServiceNow Studio

Use update sets in ServiceNow Studio to group configuration changes and move them as a unit to other instances for testing or deployment.

Update sets track configuration changes made in ServiceNow Studio so that admins and delegated developers can move those changes to other instances as a single deployable unit. Access update sets from within an individual application or from the **Deployment** tab. Update set interactions in ServiceNow Studio follow the same patterns as on the ServiceNow AI Platform. For more information about tasks you can accomplish with update sets, see System update sets.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Administrators can perform the following actions with update sets to control the full deployment lifecycle.

- Create an update set to store local changes.
- Select the current update set to direct where new local changes are recorded.
- Commit an update set to prepare it for distribution.
- Report on the contents of update sets.
- Compare update sets to determine what differences they contain.
- Merge separate update sets into a single update set.
- Create an external file from an update set.
- Retrieve update sets from remote instances to pull in changes made on other instances.
- Apply retrieved update sets to install those changes on the current instance.
- Back out changes applied from an update set.
- Set system properties related to update sets to configure default update set behavior.
- Promote update sets for deployment via ReleaseOps. For more information, see ReleaseOps.

Delegated developers can perform the following actions with update sets in ServiceNow Studio to manage changes within their assigned applications.

- Create an update set to store local changes.
- Switch between update sets using the update set picker.
- Package changes for deployment using the **Publish** button in any app.

> **Note:** If your app was created in ServiceNow IDE and converted to Fluent, you must switch from ServiceNow Studio back into the ServiceNow IDE for the publishing and deployment process.

Create an update set in ServiceNow Studio

Create an update set in ServiceNow Studio to package app changes for deployment to other instances.

Before you begin

Role required: admin or delegated_developer

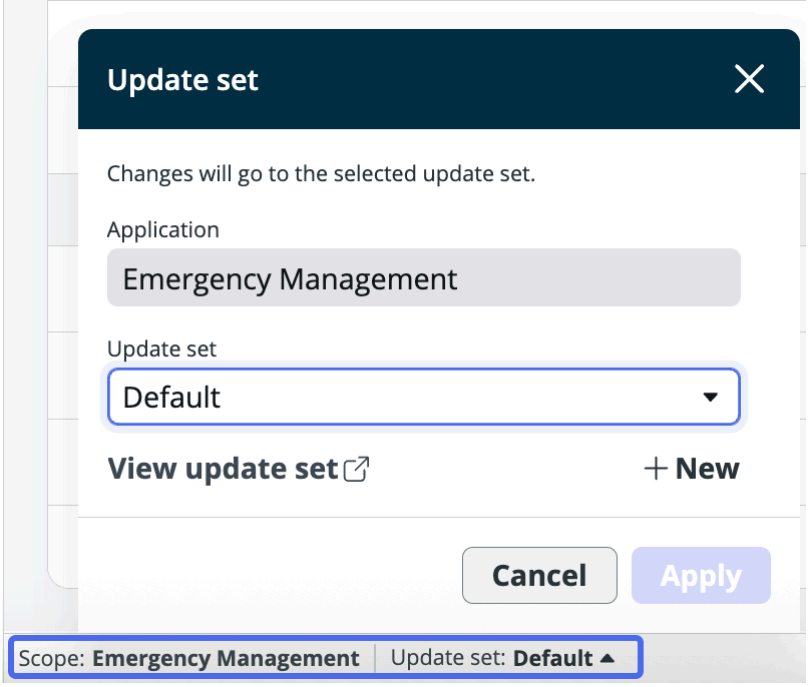
Watch this short video on how to create an update set.

https://player.vimeo.com/video/1155106572?h=620ad6f536&badge=0&autopause=0&player_id=0&app_id=58479

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Open the application that you have changes for that need a new update set.
3. At the bottom-left corner of the canvas, expand the update set dialog by selecting the current update set from the status bar.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



4. Select **New**.
5. Enter a name for the new update set.
6. Select **Save**.
7. Select **Apply**.
   The changes you made in your app go to the new update set.

Mark an update set complete in ServiceNow Studio

Mark an update set as **Complete** in ServiceNow Studio to make the changes available for retrieval by other instances.

Before you begin

Role required: admin or delegated_developer

About this task

Mark an update set as **Complete** only when it is ready to migrate. After an update set is marked **Complete**, do not change the state back to **In progress**. Create a new update set for any remaining changes instead.

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Select the **Deployment** tab.
3. Open the update set record to change.
4. Change the **State** of the update set from **In progress** to **Complete**.
   The update set is marked **Complete** and the changes are available for retrieval by other instances.

Application Repository in ServiceNow Studio

Publish custom applications from ServiceNow Studio to the ServiceNow Application Repository to distribute them to other instances in the organization.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The Application Repository stores and distributes applications between instances within an organization. Each organization can view and manage only its own published applications — applications published by other organizations are not visible.

After developing and testing an app in ServiceNow Studio and selecting **Deploy**, publish the application to the Application Repository to make it available to other instances in the organization. For more information about the publishing process, see Publish app changes to the Application Repository from ServiceNow Studio.

Note: If your app was created in ServiceNow IDE and converted to Fluent, you must switch from ServiceNow Studio back into the ServiceNow IDE for the publishing and deployment process.

For more information, see ServiceNow application repository.

Publish app changes to the Application Repository from ServiceNow Studio

Publish app changes from ServiceNow Studio to the Application Repository (App Repo) so your updates are available to all instances of your application.

Before you begin

Role required: admin or delegated_developer

About this task

The App Repo is one deployment option available in ServiceNow. You can also create update sets to publish changes. If your company has a deployment pipeline configured, you can deploy changes through the pipeline in ServiceNow Studio. For more information about the App Repo, see ServiceNow application repository.

Procedure

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Open the **App details** page for the application you want to update.
   - On the ServiceNow Studio home page, select the **Deployment** tab, select **Applications**, then select your application from the **Name** column.
   - In the Navigator panel, select the Apps icon , select your application, then select **App details**.
3. On the App details page, select **Publish**.
4. In the **New version** field, enter a version number for the changes.
5. In the **Release notes** field, enter notes about the changes you are publishing.
6. Select **Publish**.
   Your changes are submitted for review and approval. After approval, your updates are available to all instances of your application.

Pipelines in ServiceNow Studio

Use pipelines in ServiceNow Studio to automate the propagation and installation of applications from one instance to another. Admins manage deployment requests in the App Engine Management Center (AEMC).

Pipelines are powered by the ServiceNow CICD spoke, which automates processes such as publishing applications to the Application Repository, installing them on target instances, running ATF tests, and running instance scans.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: Deploying scoped apps using pipelines in App Engine Management Center (AEMC) is a premium feature that might require an additional license. For more information, see Installing ServiceNow Studio.

Admins use pipelines to move app changes through instances. For more information about configuring pipelines, see Configure Pipelines and Deployments.

With the App Engine Management Center (AEMC) installed, admins can manage deployment requests created when a developer selects **Deploy** in ServiceNow Studio to initiate the pipeline deployment process. For more information, see Managing deployments using Pipelines and Deployments.

Note: If your app was created in ServiceNow IDE and converted to Fluent, you must switch from ServiceNow Studio back into the ServiceNow IDE for the publishing and deployment process.

ServiceNow Studio reference

Use these reference topics to find details about the Navigator panel taxonomy, supported file types, properties, and collaboration permissions in ServiceNow Studio.

ServiceNow Studio Navigator panel taxonomy

Reference each metadata file type and its corresponding primary table in the File categories tab of the ServiceNow Studio Navigator panel.

Note: Any additional extensions of the sys_metadata table are automatically included in the Navigator panel in the appropriate section.

For each file category, the following sections list the available file types and their corresponding primary table. For more information about navigating to a file's primary table, see Navigate directly to a table in ServiceNow Studio.

Metadata-specific admin roles are existing roles on the ServiceNow AI Platform that provide developers with admin access to a specific metadata type. For example, a script include admin allows a developer to create, read, update, or delete any script include on the ServiceNow AI Platform and perform development on script includes in ServiceNow Studio. For more information, see ServiceNow Studio personas and roles.

AI files

AI files provide ways to use generative AI to enhance user productivity through conversation and proactive experiences.

Important: AI options in the ServiceNow Studio file taxonomy are only available if the AI components have been installed and configured.

AI file types

File type	Description	Primary table	Primary editing experience
Agentic Workflow	Agentic workflows are a business problem that you want to solve. In AI Agent Studio, you must define an agentic workflow	sn_aia_usecase	AI Agent Studio

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

AI file types (continued)

File type	Description	Primary table	Primary editing experience
	and connect it with an AI agent to get a resolution plan. For more information, see Create an agentic workflow.		
AI Agent	AI Agents are entities that mimic human-like intelligence by using large language models (LLMs). AI agents can perform tasks that range from simple automated responses to complex problem solving. By using AI agents, you can reduce the workloads of your live agents and help increase their productivity. For more information, see Create an AI agent.	sn_aia_agent	AI Agent Studio
Skill	Skills are user-defined AI capabilities tailored to meet specific business needs. Custom skills are created using ServiceNow Otto Skill Kit, which allows for a user to define input data, prompts, LLM providers and deployment areas for custom skills. For more information, see Create a skill.	sn_nowassist_skill_config	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Automation files

Automations manage recurring tasks with fewer steps, reducing human input and effort. For example, configure an email notification to send automatically when a record is approved. Add automations such as actions, flows, and other automatic tasks to streamline how apps work.

Automation file types

File type	Description	Primary table	Primary editing experience
Action	Actions automate a repeatable task or operation within a flow as a sequence of related steps. Actions run a sequence of steps to complete the task, and pass data to the flow as outputs. For more information, see Exploring actions.	sys_hub_action_type_definition	Workflow Studio
Activity definition	Activity definitions describe how the activities in your playbook get the data that they need when your playbook runs. Each activity definition contains some basic configuration details, as well as an automation plan and activity experience. For more information, see Activity definitions.	sys_pd_activity_definition	UI16
Connection and Credential Aliases	Connection and credential aliases define an alias that labels a credential or connection record, enabling an app to connect to another system or component. For more information, see Connections and Credentials.	sys_alias	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Automation file types (continued)

File type	Description	Primary table	Primary editing experience
Data stream	Data streams are reusable actions that process a stream of response data within a flow. For example, you can create a data stream action to import a large quantity of employee data from a third-party HR site. For more information, see Data Stream actions and pagination.	sys_hub_action_type_definition	Workflow Studio
Decision table	Decision tables decouple decision logic from your code by creating and maintaining decision rules. Decision tables provide a single point where you can create, view, and modify decisions. For more information, see Exploring Decision Tables.	sys_decision	Workflow Studio
Email Template	Email templates enable administrators to create reusable content for the subject line and message body of email notifications. Admins can add rich text and other items, such as images, to email templates. For more information, see Email templates.	sysevent_email_template	UI16
External trigger	External triggers are reusable triggers that respond to external events through	sys_ih_external_trigger_definition	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Automation file types (continued)

File type	Description	Primary table	Primary editing experience
	webhooks. When an event occurs in the configured third-party application that meets the specified conditions, the trigger is activated. For more information, see Create a saved external trigger.		
Flow	Flows are automated processes that consist of a trigger and a sequence of reusable actions and flow logic. The trigger specifies when to run the flow. The actions perform a sequence of operations on your data. For example, the Visual Task Boards (VTB) Sample Flow creates and assigns a VTB card whenever a priority 1 incident is created. For more information, see Exploring flows.	sys_hub_flow	Workflow Studio
Notification	Notifications alert users when a record changes. For example, you could get a push notification when a request is rejected. For more information, see System notifications.	sysevent_email_action	UI16
Playbook	Playbooks are sets of automated activities that occur based on a trigger. For example, you can create a playbook for your app	sys_pd_process_definition	Workflow Studio

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Automation file types (continued)

File type	Description	Primary table	Primary editing experience
	to send an email when a request is approved. For more information, see Building your first playbook.		
Subflow	Subflows are processes that consist of a sequence of reusable actions and flow logic, data inputs, and outputs. In contrast to flows, subflows do not have a trigger but instead run when called from a playbook, flow, another subflow, or a script. For more information, see Exploring subflows.	sys_hub_flow	Workflow Studio
Trigger	Triggers contain the start conditions for your automation. You can create separate triggers for each flow or create a reusable saved trigger that you can use in multiple flows. For more information, see Building triggers.	sys_hub_trigger_template	Workflow Studio

Client development files

Client development files define operations that occur based on actions users take when working with an app built in ServiceNow Studio.

Client development file types

File type	Description	Primary table	Primary editing experience
Assignment Data Lookup	Assignment data lookups assign a	dl_u_assignment	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Client development file types (continued)

File type	Description	Primary table	Primary editing experience
	record automatically using Data Lookup and Record Matching. For example, automatically set a value in the assigned_to and assignment_group fields for a record when a set of conditions occurs, such as assigning approvals to a group of managers for users below a certain level. For more information, see Defining assignment rules.		
Client Extension Instance	Use a client extension integration to a registered instance of a client extension point that links a UI script with a client extension point. The UI script is included on pages that invoke the client extension point. For more information, see Using extension points to extend application functionality.	sys_client_extension_instance	UI16
Client Extension Point	Client extension points extend the functionality of an application without altering the original application code. Extension points help prevent custom code interactions from breaking after an upgrade when custom code is	sys_client_extension_point	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Client development file types (continued)

File type	Description	Primary table	Primary editing experience
	directly embedded into application code. For more information, see Using extension points to extend application functionality.		
Client Script	Client scripts run JavaScript on the client (web browser) when client-based events occur. For example, a client script could run when a form loads, after form submission, or when a field changes value. For more information, see Client scripts.	sys_script_client	UI16
Data Lookup Definitions	Data lookup definitions are no-code solutions that retrieve attributes from a record on the same table. For example, you can create a data lookup definition to populate an email field automatically when you enter your name. For more information, see Create a catalog lookup definition.	dl_definition	UI16
Priority Data Lookup	Priority data lookups define the impact and urgency of an incident to calculate how it should be prioritized.	dl_u_priority	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Client development file types (continued)

File type	Description	Primary table	Primary editing experience
	For more information, see Define priority lookup rules.		
UI Extension Instance	Use a UI extension instance to create a registered instance of a UI extension point that links a UI macro with a UI extension point. The macro can be called whenever the UI extension point is invoked. For more information, see Creating and adding a UI extension point.	sys_ui_extension_instance	UI16
UI Extension Point	UI extension points add custom content to a UI page without modifying the page directly. Create the UI extension points first, then add them to the UI macros in the base application code. For more information, see Using UI extension points in server-side UI macros.	sys_ui_extension_point	UI16
UI Policy	UI policies dynamically change the behavior of information on a form and control custom process flows for tasks. For example, use UI policies to make the number field on a form read only, make the short description field required, and hide other fields.	sys_ui_policy	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Client development file types (continued)

File type	Description	Primary table	Primary editing experience
	For more information, see Using UI policies.		
UI Script	UI scripts package client-side JavaScript into a reusable form, similar to how script includes store server-side JavaScript. For more information, see UI scripts.	sys_ui_script	UI16

Content files

Content files provide users and systems with information by referring to knowledge articles and other forms of communication. For example, content files can reference knowledge articles, static blocks of text, or external web-based content. Use content files to provide information to apps built in ServiceNow Studio.

Content file types

File type	Description	Primary table	Primary editing experience
Audio	Use audio files to upload sounds and recordings that your app can use on the ServiceNow AI Platform. For more information, see Manage audio files.	db_audio	UI16
Detailed content	Use a detailed content block to display the content of an existing document, such as an incident, knowledge article, or service management request. For more information, see Configure a detailed content block.	content_block_detail	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Content file types (continued)

File type	Description	Primary table	Primary editing experience
Dynamic content	Dynamic content uses scripting or pulls information from the ServiceNow AI Platform into an app. For example, use dynamic content to create a job posting, where the postings are stored in knowledge articles and displayed in the app with a dynamic block. For more information, see Configure dynamic blocks.	content_block_programmatic	UI16
iFrames	Use iFrames to embed a URL on a page within a frame. You can embed external pages or render ServiceNow content. For more information, see Configure iFrames.	content_block_iframe	UI16
Images	Upload and store images on the ServiceNow AI Platform to be used in apps and forms. Reference images from HTML fields by appending the name of the image to the URL of the instance. For more information, see Storing images in the database.	db_image	UI16
Static content	Use static blocks for text that does not change. For example, use a static block	content_block_static	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Content file types (continued)

File type	Description	Primary table	Primary editing experience
	for a site footer with only the company or organization name. For more information, see Configure a static HTML block.		

Data files

Data files store and reference the information that applications consume. Start with a table, or upload spreadsheets and PDFs to get data into the ServiceNow AI Platform for apps to use.

Data file types

File type	Description	Primary table	Primary editing experience
Form	Forms are content pages that display fields and values for a single record from a database table. For more information, see Forms in Table Builder.	sys_ui_form	Form Builder
Form section	Sections control the layout of a form. For example, you can have one column or two. For more information, see Customize your form layout in Table Builder.	sys_ui_section	Form Builder
Many to Many Definition	Many to many tasks define relationships between different tasks. You can implement one-to-one, one-to-many, and many-to-many relationships. For example, users and roles are a many-to-	sys_m2m	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Data file types (continued)

File type	Description	Primary table	Primary editing experience
	many relationship because a user can have multiple roles, and multiple users can have any given role. For more information, see Creating many-to-many task relations.		
Relationship	Relationships define how tables interact with each other. You can create relationships between tables by extending tables, referencing records in another table, creating many-to-many relationships, and joining tables in a database view. For more information, see Table relationships.	sys_relationship	UI16
Table	Tables are the foundation of how the ServiceNow AI Platform stores data. When you view a table as a list, each row is a record, and each column is a field from the record. For example, the Incident table has a record for every customer interaction, or incident. For more information, see ServiceNow AI Platform tables and data and Table Builder. Preferred table admins can configure	sys_db_object	Table Builder

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Data file types (continued)

File type	Description	Primary table	Primary editing experience
	preferred tables for Table Builder. For more information, see ServiceNow Studio personas and roles.		
Table Column	Table columns represent a field from a record. For example, a user record might have a column for first name and a separate column for family name. For more information, see Table properties in Table Builder.	sys_dictionary	UI16

Inbound integration files

Inbound integrations bring information and data onto the ServiceNow AI Platform from another system or source, enabling data flow from external systems into ServiceNow.

Inbound integration file types

File type	Description	Primary table	Primary editing experience
Data Import	Data imports show all records being processed for an import job and the import jobs awaiting approvals. For more information, see Importing data using import sets and Integration Hub - Import.	sn_ihub_integration_instance	Integration Hub
Data Source	Data sources specify how and where to get the data you want to import. For more information, see Data sources	sys_data_source	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inbound integration file types (continued)

File type	Description	Primary table	Primary editing experience
	and Configure a data source.		
Scheduled Data Import	Scheduled data imports specify to import data from data sources using import sets. Transform maps are applied to the imported data before writing the data to the target table. For more information, see Run or schedule a data import.	scheduled_import_set	UI16
Scheduled Data Import	Scheduled data imports specify to import data from data sources. Transform maps are applied to the imported data before writing the data to the target table. For more information, see Run or schedule a data import.	scheduled_data_import	UI16
Scripted REST API	Scripted REST APIs build custom web service APIs for your application. You can define service endpoints, query parameters, and headers for a scripted REST API, and scripts to manage the request and response. For more information, see Scripted REST APIs.	sys_ws_definition	UI16
Scripted Web Service	Scripted web services enable developers to create their own APIs	sys_web_service	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inbound integration file types (continued)

File type	Description	Primary table	Primary editing experience
	on the ServiceNow AI Platform. Third-party applications use scripted web services to access records in ServiceNow tables. For more information, see Web services.		
Table Transform Map	Transform maps contain a set of field maps that determine the relationships between fields in an import set and fields in an existing ServiceNow table, such as Incident [incident] or User [sys_user]. After creating a transform map, you can reuse it to map data from another import set to the same table. For more information, see Transform maps.	sys_transform_map	UI16

Outbound integration files

Outbound integrations send data from the ServiceNow AI Platform to an external system or source.

Outbound integration file types

File type	Description	Primary table	Primary editing experience
Export Definition	Export definitions determine the data to include in an export set. For more information, see Create an export definition.	sys_export_definition	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Outbound integration file types (continued)

File type	Description	Primary table	Primary editing experience
Export Set	Export sets define the data to export and the export target to use. For example, push data from an instance to an external file. For more information, see Create an export set.	sys_export_set	UI16
Export Target	Export targets specify the target file on a MID Server to which the export set data is written. For more information, see Create an export target.	sys_export_target	UI16
REST Message	REST messages store details on how to interact with an external web service through REST. Use REST messages to send requests to a REST web service endpoint by creating a REST message record. For more information, see Create a REST message.	sys_rest_message	UI16
Scheduled Data Export	Scheduled data exports specify a schedule for running export sets. A single export can be scheduled, or regular intervals can be scheduled with support for delta records only.	scheduled_data_export	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Outbound integration file types (continued)

File type	Description	Primary table	Primary editing experience
	For more information, see Schedule an export.		
SOAP Message	SOAP messages store details on how to interact with an external web service through SOAP. SOAP messages define the remote endpoint, web services description language (WSDL), and authentication settings. For more information, see SOAP message.	sys_soap_message	UI16

MID Server files

Work with management, instrumentation, and discovery (MID) Server files to facilitate communication and data movement between a single ServiceNow instance and external applications, data sources, and services.

MID Server file types

File type	Description	Primary table	Primary editing experience
MID Server Application	The ServiceNow MID Server is a Java application that runs as a Windows service or UNIX daemon on a server in your local network. For more information, see MID Server.	ecc_agent_application	UI16
MID Server Capability Value Test	MID Server capabilities define the specific functions of a MID Server within an IP address range. The capability value can be empty, a single value, or a * (wildcard).	ecc_agent_capability_value_test	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

MID Server file types (continued)

File type	Description	Primary table	Primary editing experience
	Use value tests to create capabilities that find devices using values without requiring exact string matching. For more information, see MID Server capabilities.		
MID Server IP Range	MID Server IP ranges specify an IP range or the specific IP address of a target. For more information, see Configure an IP address range for the MID Server.	ecc_agent_ip_range	UI16
MID Server Property	MID Server properties define the behavior of one or more MID Servers. For more information, see MID Server properties.	ecc_agent_property	UI16
MID Server Script File	MID Server script files synchronize to a connected MID Server. For more information, see Attach a script file to a file synchronized MID Server.	ecc_agent_script_file	UI16
MID Server Script Include	MID Server script includes make REST calls to cloud providers. For more information, see CAPI classes in MID Server script includes.	ecc_agent_script_include	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Mobile files

Mobile App Builder builds and manages screens and records that make up workflows within ServiceNow mobile apps. Mobile Card Builder files edit the cards and templates used in applications for iOS and Android.

Mobile file types

File type	Description	Primary table	Primary editing experience
Analytics preview	Analytics previews display previews of data visualization charts and single score reports in your launcher screen's analytics section. Analytics previews verify that your data is tailored for mobile use and communicates the appropriate information for users. For more information, see Create a mobile analytics preview.	sys_sg_chart	Mobile App Builder
Calendar screen	Calendar screens display a calendar interface and records associated with the selected date. Use a calendar screen to display dates relevant to application records. For example, display when tasks are due or when important events take place. For more information, see Calendar screen.	sys_sg_calendar_screen	Mobile App Builder
Card	Cards are predetermined layouts that can show visuals, text, and data in mobile applications. You can define card elements and specify how elements are	sys_sg_view_config	Mobile Card Builder

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Mobile file types (continued)

File type	Description	Primary table	Primary editing experience
	arranged within a card. For more information, see Cards and icons.		
Card template	Card templates are preconfigured layouts or frameworks that determine how information is displayed in mobile application cards. Use the existing Mobile Card Builder card templates or create your own templates. For more information, see Create a card template with Mobile Card Builder.	sys_sg_view_template	Mobile Card Builder
Chart screen	Chart screens display data visualizations created in the Analytics Overview in the analytics section of your launcher screen. Adding data visualizations helps identify trends and turning points through indicator scores and visual representation. For more information, see Chart screen.	sys_sg_chart_screen	Mobile App Builder
Custom Map Screen	Custom map screens create maps that display content for specific records. For more information, see Configure a map screen.	sys_sg_custom_map_screen	Mobile App Builder

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Mobile file types (continued)

File type	Description	Primary table	Primary editing experience
Function	Functions determine what actions users can perform in your mobile application. For example, create a navigation function that enables users to open a record from a list, or move from an employee user profile screen to a manager user profile screen. For more information, see Mobile functions.	sys_sg_button	Mobile App Builder
Input form screen	Input form screens provide interfaces for users to enter information in mobile applications. For example, use input form screens to create or edit records, complete surveys, or any other situation where users must enter information. For more information, see Input form screen.	sys_sg_parameter_screen	Mobile App Builder
Launcher screen	Launcher screens serve as landing pages or home pages. Using a launcher screen, you can access screens in various formats, search, perform quick actions, and find user information. For more information, see Launcher screens.	sys_sg_applet_launcher	Mobile App Builder

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Mobile file types (continued)

File type	Description	Primary table	Primary editing experience
List screen	List screens display a list of records. Records in list screens appear in a card format, showing a limited selection of the information for the record. For more information, see List screen.	sys_sg_list_screen	Mobile App Builder
Map screen	Map screens display a map with locations associated with the records in a data item. For example, map screens can show users where their assets are located, or which job locations they must travel to. For more information, see Map screen.	sys_sg_map_screen	Mobile App Builder
Mobile app config	Mobile app configs create customized mobile experiences for the Now Mobile app and Mobile Agent app. For more information, see Configuring the Mobile Platform.	sys_sg_native_client	Mobile App Builder
Mobile web screen	Mobile web screens open an external URL or a relative URL within your instance. For more information, see Mobile web screen.	sys_sg_browser_screen	Mobile App Builder
Record screen	Record screens display content for a	sys_sg_form_screen	Mobile App Builder

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Mobile file types (continued)

File type	Description	Primary table	Primary editing experience
	specific single record. Configure functions on record screens to enable users to make edits and perform actions. For more information, see Record screen.		

Natural Language Understanding (NLU) files

Natural Language Understanding (NLU) is a model that enables a computer to interpret, analyze, and derive meaning from human language. Create NLU files to build and train an application's NLU model to recognize user input, or utterances, and determine the corresponding user or system actions.

NLU file types

File type	Description	Primary table	Primary editing experience
NLU Model	Natural Language Understanding (NLU) models are the collection of utterance examples and their associated intents and entities that an application uses as a reference to infer intents and entities in a new utterance. For more information, see (Legacy) Natural Language Understanding.	sys_nlu_model	UI16

Properties files

Properties files are configurable parameters that change the behavior of an application without hard-coding values directly into scripts.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties file types

File type	Description	Primary table	Primary editing experience
Message	Messages are the text values used in informational messages, confirmation messages, error messages, and other types of system messages in your application. For more information, see Message table.	sys_ui_message	UI16
System Property	System properties store important values for an application, such as settings or configurations, that scripts may need to reference. Storing values as system properties allows updates in one place without changing each script that references the values. For more information, see What are application properties? and Available system properties.	sys_properties	UI16
System Property Category	System property categories create the page layout for an application's system properties, displaying all application properties in a single location. For more information, see Create a system property category.	sys_properties_category	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Reporting files

Reporting files create and distribute reports that show the current state of instance data, such as the number of open incidents of each priority.

Reporting file types

File type	Description	Primary table	Primary editing experience
Chart Colors	Chart colors assign a consistent color to a grouping or stacking value in reports and dashboards. The color stays the same across reports regardless of the order of the values. For more information, see Chart colors.	sys_report_chart_color	UI16
Color Definition	Color definitions maintain consistency in the platform's look and feel by applying defined colors to various UI components, such as buttons, backgrounds, text, and other elements. For more information, see Define system colors for analytics.	sys_report_color	UI16
Dashboard	Dashboards display performance analytics, reporting, and other widgets on a single screen. Use dashboards to create a story with data that you can share with other users. For more information, see Create and use Core UI dashboards.	pa_dashboards	Dashboard Builder
Metric Definition	Metric definitions measure and evaluate	metric_definition	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Reporting file types (continued)

File type	Description	Primary table	Primary editing experience
	the effectiveness of an application process. For example, a metric could measure the effectiveness of the incident resolution process by calculating how long it takes to resolve an incident. Define metrics and create reports and dashboards using your metrics definitions. For more information, see Metrics and Define a metric.		
Range	Ranges are defined data intervals used in bar and pie charts to segment data into logical groups. For example, create a range to see how many tasks were completed well within the service level agreement (SLA) and how many tasks elapsed during the SLA. For more information, see Report ranges.	sys_report_range	UI16
Report	Reports display data visually to provide insights, track performance, and support data-driven decisions. For example, create and distribute reports that show the current state of instance data, such as the number	sys_report	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Reporting file types (continued)

File type	Description	Primary table	Primary editing experience
	of open incidents of each priority. For more information, see Exploring reporting.		
Scheduled Email of Report	Scheduled emails of reports generate and distribute scheduled reports via email. For more information, see Schedule emails of Core UI reports.	sysauto_report	UI16

Schedules files

Schedules manage the life cycle of an application, from setting up maintenance schedules to defining durations and risks for operations.

Schedules file types

File type	Description	Primary table	Primary editing experience
Blackout Schedule	Blackout schedules are times during which certain activities, such as changes or updates, are restricted to avoid disruptions. Set up blackout schedules to confirm that critical business operations remain unaffected during high-impact or sensitive times, such as holidays, end-of-quarter financial processing, or other key business events. For more information, see Create blackout and maintenance schedules in Change Management.	cmn_schedule_blackout	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Schedules file types (continued)

File type	Description	Primary table	Primary editing experience
Maintenance Schedule	Maintenance schedules are times during which planned maintenance activities, such as changes and updates, should take place. Maintenance schedules usually occur during low-impact times to minimize disruptions to business operations. For more information, see Create blackout and maintenance schedules in Change Management.	cmn_schedule_maintenance	UI16
Relative Duration	Relative durations are a duration type available in ServiceNow Studio that you can select when defining service level agreements (SLAs). Relative durations calculate how much time is available to work on an SLA by defining the amount of time to wait. For example, define a relative duration as three business days by 4pm. For more information, see Define a relative duration and Use a relative duration.	cmn_relative_duration	UI16
Risk Conditions	Risk conditions are sets of rules or criteria that evaluate the potential risks associated	risk_conditions	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Schedules file types (continued)

File type	Description	Primary table	Primary editing experience
	with scheduling activities, such as changes, updates, or maintenance tasks. Define risk conditions and run risk calculations using The Best Practice - Change Risk Calculator. For more information, see Add or modify risk and impact conditions and Risk conditions and calculation.		
Schedule	Schedules are rules that include or exclude time for various actions or tasks. For more information, see Schedules and Define a schedule.	cmn_schedule	UI16

Security files

Security files control who has access to application data and help prevent accidental modification or deletion of data.

Security file types

File type	Description	Primary table	Primary editing experience
Access Control	Access controls, or access control lists (ACLs), restrict access to data by requiring users to pass a set of requirements before they can interact with application content.	sys_security_acl	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Security file types (continued)

File type	Description	Primary table	Primary editing experience
	For more information, see Explore Access Control Lists.		
Public Pages	Public pages allow users to see the application content without logging in. For more information, see Make UI pages public or private.	sys_public	UI16
Role	Roles determine what application access is granted to which users. For more information, see Managing roles and Determine What Roles to Create.	sys_user_role	UI16

Server development files

Server development files manage back-end processes in ServiceNow Studio to verify that data is handled appropriately and securely.

Server development file types

File type	Description	Primary table	Primary editing experience
Business Rule	Business rules are a server-side script that runs when a record is displayed, inserted, updated, or deleted, or when a table is queried. Establish server-side conditions to determine when a business rule script should run and what record operations the business rule applies to.	sys_script	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Server development file types (continued)

File type	Description	Primary table	Primary editing experience
	For more information, see Classic Business rules.		
Data Policy	Data policies enforce data consistency by setting mandatory and read-only states for fields. Data policies are similar to UI policies, but UI policies only apply to data entered on a form through the standard browser. Data policies apply rules to all data entered into the system, including data brought in through import sets or web services and data entered through the mobile UI. For more information, see Data policy.	sys_data_policy2	UI16
Event Registration	Events are special records that the system uses to log when certain conditions occur and to take some kind of action in response. By registering an event, you can define properties about the event and associate the event with the business rule that fires the event. For more information, see Register an event.	sysevent_register	UI16
Extension Instance	Extension instances are registered instances of a	sys_extension_instance	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Server development file types (continued)

File type	Description	Primary table	Primary editing experience
	scripted extension point that links a script include with a scripted extension point. Use an extension instance to define custom logic or methods without affecting original code. An extension instance encapsulates specific logic and functions, making it easier to manage, update, and debug code. For more information, see Using extension points to extend application functionality.		
Extension Point	Extension points designate where custom script logic can be incorporated into your code to integrate customizations and new features without altering existing application code. Data or objects returned by an extension point must conform to requirements specified by the application creator. For more information, see Using extension points to extend application functionality.	sys_extension_point	UI16
Fix Script	Fix scripts are server-side JavaScript that run after a custom application is installed or upgraded. Include fix scripts to make	sys_script_fix	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Server development file types (continued)

File type	Description	Primary table	Primary editing experience
	changes necessary for the data integrity or product stability of an application. For more information, see Fix scripts and Create a fix script.		
Scheduled Script Execution	Scheduled script executions, also known as scheduled jobs, are automated, server-side script logic that execute at a specific time or on a recurring basis. Use scheduled script executions when application processes require script logic to execute based on a time schedule. For more information, see What is a Scheduled Script Execution? and Creating a Scheduled Script Execution.	sysauto_script	UI16
Script Action	Script actions are server-side JavaScript that executes when a particular event is generated. For more information, see Script actions.	sysevent_script_action	UI16
Script Include	Script includes are reusable, server-side JavaScript that define a function or class and execute only when explicitly called.	sys_script_include	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Server development file types (continued)

File type	Description	Primary table	Primary editing experience
	For more information, see Script includes and Script Includes.		
UI Action	UI actions define the behavior of buttons, links, or context menu items in your application, specifying how they interact with the server-side database. For more information, see Defining UI actions and Create a UI action.	sys_ui_action	UI16

UI Builder files

UI Builder files build pages for workspaces generated in ServiceNow Studio, or custom web experiences using Next Experience Components and custom web components.

UI Builder file types

File type	Description	Primary table	Primary editing experience
Components	Components are used in the UI Builder to build pages. Components have an interface that an end user can view and interact with. Components can talk to each other through events and properties. Commonly used components include Heading, Image, List, Form, and Button. For more information, see Customize UI Builder pages using components.	sys_ux_macroponent	UI16
Controllers	Controllers are a type of data resource that includes	sys_ux_controller	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

UI Builder file types (continued)

File type	Description	Primary table	Primary editing experience
	data and event logic and enables component presets. Controllers are added automatically when using a page template. There are two types of controllers: • Data controllers contain data resources and can be manually added to a page. • UI controllers are added to pages when using page templates and cannot be added manually. For more information, see Bind data to UI Builder pages using controllers (advanced feature).		
Experiences	Experiences are a collection of web pages for users to interact with an application. For more information, see Create an experience for UI Builder.	sys_ux_page_registry	UI16
Page collections	Page collections are groups of pages that can be reused in experiences within tabs or modals. For more information, see #unique_350.	sys_ux_extension_point	UI16
UI interactions	UI interactions are reusable flows that combine UI elements	sys_ui_interaction	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

UI Builder file types (continued)

File type	Description	Primary table	Primary editing experience
	and logic into a single unit that you can trigger from component events in UI Builder. They help avoid repeating the same wiring across multiple pages and make designs easier to maintain. For more information, see UI interactions.		

User interface files

User interface files define layouts for pages, modules, and tools that users interact with. Examples include catalog items, guided tours, and themes.

User interface file types

File type	Description	Primary table	Primary editing experience
Application Module	Application modules are groupings of modules as they appear in the application navigator (UI16) or **All** menu (Next Experience). You can refer to an Application menu as simply an application. For more information, see Enable or disable an application menu or module.	sys_app_application	UI16
Assessment Metric	In the Assessments application, metrics are traits or values used to evaluate assessable records. For more information, see Assessment metrics.	asmt_metric	Survey Designer

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

User interface file types (continued)

File type	Description	Primary table	Primary editing experience
Catalog	Catalogs are sections of the Service Catalog where users can order items and services. A catalog is like a portal where users can request catalog items such as service and product offerings. For example, a hardware catalog may have items to request a new keyboard or a new mouse device. For more information, see Exploring Service Catalog.	sc_catalog	Catalog Builder
Catalog Item	Catalog items are forms that describe a good or service available to order in the service catalog. For example, when requesting time off using a catalog item, you may enter your name and requested dates off on the form. For more information, see Service Catalog items.	sc_cat_item	UI16
Context Menu	Context menus for forms provide controls for a list or form based on the table and user access rights. For more information, see Form context menu.	sys_ui_context_menu	UI16
Embedded Help	Embedded Help provides targeted help content to a user in a UI page,	sys_embedded_help_content	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

User interface file types (continued)

File type	Description	Primary table	Primary editing experience
	based on their role. Some embedded help content comes with the base instance. Your organization can add or replace embedded help content. For more information, see Embedded Help.		
Embedded Help Qualifier	Embedded Help qualifiers are identifiers that help a ServiceNow instance identify the correct Embedded Help topic when there could be more than one topic for a UI page. For more information, see Use qualifiers in Embedded Help.	sys_embedded_help_qualifier	UI16
Guided Tour	Guided Tours contain a series of interactive steps that help users complete online tasks within their browser window to train and onboard users working in a ServiceNow app. For more information, see Guided Tours.	sys_embedded_tour_guide	UI16
List	Lists display a set of records from a table, and can be filtered to refine the contents. For example, filter the Task list to show only Unassigned tasks. Each row in a list is a record, and each	sys_ui_list	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

User interface file types (continued)

File type	Description	Primary table	Primary editing experience
	column is a field from the record. For more information, see Lists in the classic environment and ServiceNow AI Platform® list administration.		
List Control	List controls specify which features are available on a list, such as the **New** and **Edit** buttons. For more information, see Configure list controls.	sys_ui_list_control	UI16
Map Page	Map pages display ServiceNow data graphically on a Google map page based on location data that you provide. For more information, see Map pages.	cmn_map_page	UI16
Module	Modules are any links in the application navigator (UI16) or **All** menu (Next Experience) that opens a page in the content frame or in a separate tab or window. For more information, see Modules on the Developer Site.	sys_app_module	UI16
Portal	Portals provide users with access to the services, information, and resources they need to complete	sys_ux_page_registry	UI Builder

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

User interface file types (continued)

File type	Description	Primary table	Primary editing experience
	their work. Work with portals in UI Builder. For more information, see Configure UI Builder portal experiences.		
Record Producer	Record producers are catalog items that create task-based records, such as incident records, from the service catalog. For more information, see Record Producer.	sc_cat_item_producer	Catalog Builder - Record producer
Related List	Related lists appear on forms and show records in tables that have relationships to the current record. For more information, see Related lists.	sys_ui_related_list	UI16
Schedule Page	Schedule pages are records that contain a collection of scripts that enable custom generation of a calendar or timeline display. For more information, see Schedule pages.	cmn_schedule_page	UI16
Service Portal	Service Portal builds a mobile-friendly self-service portal experience for employees or customers. For more information, see Service Portal.	sp_portal	Service Portal

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

User interface file types (continued)

File type	Description	Primary table	Primary editing experience
Style	Styles define properties such as font size, border, and alignment for text that appears in your app. For more information, see Create a Next Experience style and Style - Scoped, Global.	sys_ui_style	UI16
Template	Templates create reusable content for administrators. For example, an email template could have a reusable subject line and message body for email notifications. Form templates simplify the process of submitting new records by populating fields automatically. For more information, see Email templates, Using form templates, and Page templates.	sys_template	UI16
Theme	Themes tailor the visual experience for users by updating the look and feel to match your brand. For more information, see Working with themes in Next Experience.	sys_ui_theme	UI16
Timeline Page	Timeline pages track any activity bounded by two dates, such as change request start and end dates,	cmn_timeline_page	UI16

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

User interface file types (continued)

File type	Description	Primary table	Primary editing experience
	or incident open and close dates. For more information, see Timeline pages.		
UI Page	UI pages can be used to create and display forms, dialogs, lists, and other UI components. Use UI pages as widgets on dashboards. To find the UI pages, navigate to System UI > UI Pages. For more information, see UI pages.	sys_ui_page	Fluent source
View Rule	View rules force a specified view when users access a page or application. For more information, see Create a view rule.	sysrule_view	UI16
Workspace	Workspaces provide agents and managers with tools to answer customer questions and resolve customer problems. Workspaces are primarily used for request and fulfillment processes, such as a service desk to manage tickets. For more information, see Configurable Workspace UI and Workspace Builder.	sys_ux_page_registry	Workspace Builder

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ServiceNow Studio properties

Configure system properties to control ServiceNow Studio application behavior and delegated development deployment. Access ServiceNow Studio system properties by navigating to **All** > **sys_properties.list**.

System properties

System property	Description
sn_devstudio.servicenow_studio_banner.enable	Shows the banner for navigating to ServiceNow Studio from App Engine Studio or legacy Studio. Set to **false** to hide the banner. • Type: true | false • Default value: true • Location: System Properties [sys_properties] table
sn_sns.is_app_summarization_disabled	This property is not included by default in the ServiceNow AI Platform. To disable ServiceNow Otto app summary generation in ServiceNow Studio, create a new property on the System Properties [sys_properties] table named `sn_sns.is_app_summarization_disabled` and set the value to **true**.
sn_udc.smaller_app_files_count_threshold	Determines the number of files an app can contain and still be treated as a small app in the file navigator. Small apps load all files at once in the file navigator; large apps load a subset of the files on open. • Admin: Less than or equal to 2300 files • Delegated developer: Less than or equal to 1150 files

Delegated Development and Collaboration properties

Property	Description
com.snc.dd.manage_update_set_enabled	Enables or disables display of the **Manage Update Set** permission. Delegated developers cannot see the update sets list unless this property is set to **true**. • Type: true | false • Default value: false • To enable the display of this permission, set this value to **true**.
com.snc.dd.publish_to_app_repo_enabled	Enables or disables display of the **Publish To App Repo** permission.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Delegated Development and Collaboration properties (continued)

Property	Description
	• Type: true | false • Default value: true • To disable display of this permission, set this value to **false**.
com.snc.dd.publish_to_app_store_enabled	Enables or disables display of the **Publish To App Store** permission. • Type: true | false • Default value: true • To disable display of this permission, set this value to **false**.
com.snc.dd.publish_to_update_set_enabled	Enables or disables display of the **Publish To Update Set** permission. • Type: true | false • Default value: false, which disables display of this permission. • To enable the display of this permission, set this value to **true**.
com.snc.dd.upgrade_app_enabled	Enables or disables display of the **Upgrade App** permission. • Type: true | false • Default value: true • To disable display of this permission, set this value to **false**.

Source code development properties

Property	Description
sn_glider.default_to_bundled_sdk	Specifies whether to use the bundled version or the latest version of the ServiceNow SDK when creating or converting applications in the ServiceNow IDE. If true, the version of the ServiceNow SDK that's bundled with a version of the ServiceNow IDE is used. If false, the latest version of the ServiceNow SDK is used.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Source code development properties (continued)

Property	Description
	• Type: true | false • Default value: false • Location: Add the property to the System Property [sys_properties] table
sn_glider.enable_ide	Enables access to the ServiceNow IDE. If false, access to the ServiceNow IDE is turned off across the instance. • Type: true | false • Default value: true • Location: Add the property to the System Property [sys_properties] table **Important:** As of version 29.2.11, turning off this property also turns off access to ServiceNow Studio across the instance.
sn_glider.fluent_convert_enabled	Enables converting existing applications that weren't created with the ServiceNow IDE or ServiceNow SDK to support development in source code from the ServiceNow IDE. • Type: true | false • Default value: true • Location: Add the property to the System Property [sys_properties] table • Learn more: Convert an application with the ServiceNow IDE
sn_glider.git.attachment.extension.binary	Defines a custom extension for attachment files with binary content types (for example, `application/octet-stream`) to override the default `gitdata` extension. The custom extension that you specify with this property must also be specified with the *`glide.attachment.extensions`* system property. **Note:** If you reuse an existing extension, confirm that the value you set passes MIME type validations for binary types. • Type: string • Default value: gitdata

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Source code development properties (continued)

Property	Description
	• Location: Add the property to the System Property [sys_properties] table • Learn more: ServiceNow IDE MID Server User [sn_glider.ide_git_user]
sn_glider.git.attachment.extension.text	Defines a custom extension for attachment files with text content types (for example, `text/plain`) to override the default `txt` extension. The custom extension that you specify with this property must also be specified with the *`glide.attachment.extensions`* system property. **Note:** If you reuse an existing extension, confirm that the value you set passes MIME type validations for text types. • Type: string • Default value: txt • Location: Add the property to the System Property [sys_properties] table • Learn more: ServiceNow IDE MID Server User [sn_glider.ide_git_user]
sn_glider.pinnedSdkVersion	Specifies the version of the ServiceNow SDK to use when creating or converting applications in the ServiceNow IDE. • Type: string • Default value: The latest version of the ServiceNow SDK • Location: Add the property to the System Property [sys_properties] table

Platform navigation properties

glide.ui.open_in_studio_button.enabled	This property is not included by default in the ServiceNow AI Platform. To disable the right-click Configure redirect to ServiceNow Studio, create a property on the System Properties [sys_properties] table named `glide.ui.open_in_studio_button.enabled` and set the value to **false**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ServiceNow Studio supported file types using code search

Reference the file types that ServiceNow Studio supports when using code search to locate scripts and metadata across an instance.

Code search in ServiceNow Studio uses the default code search group. The following table lists the file types supported by the default ServiceNow Studio code search.

Tip: To expand the list of supported file types, an administrator can create a new code search group that includes additional file types. For more information, see Expand code search tables.

Note: File types must extend the sys_metadata table to be searchable in code search. Other file types, such as scheduled items (sys_trigger records), are not searchable in ServiceNow Studio.

Supported file types for code search

File type	Code search table name	Description
Access control	sys_security_acl	Determines whether access is granted for a specified operation to a specific entity, such as: • Type of entity being secured • Operation being secured • Unique identifier describing the object
Business rules	sys_script	Customize system behavior. • Business Rules run when a database action occurs (query, insert, update, or delete). • The script can run: ○ before or after the database action is performed (runs as part of the database operation). ○ asynchronously (at some point after the database operation). ○ on display (when displaying the data in a form).
Client scripts	sys_script_client	Change the appearance of forms and display fields based on entered values or other custom display options.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Supported file types for code search (continued)

File type	Code search table name	Description
		• onLoad: The client script runs when the form or page loads. • onChange: The client script runs when a specific field changes and also when the form or page loads. • onSubmit: The client script runs when the form is submitted. Client scripts can also be called by other scripts or modules, including UI policies.
Email templates	sysevent_email_template	Create reusable content for the subject line and message body of email notifications.
Inbound email actions	sysevent_in_email_action	Script how the system responds to inbound emails.
Map pages	cmn_map_page	Display ServiceNow data graphically on a Google map page based on location data that you provide.
Map transforms	sys_transform_map	Import data by defining mapping relationships between tables. Transform maps: • Define mapping relationships between tables. • Can use business rules, other scripts, or other options to import data.
Notifications	sysevent_email_action	Determine how an application communicates with users and alerts them about important application-related events.
Script processors	sys_processor	Provide customizable URL endpoints that execute server-side JavaScript code and produce output such as TEXT or JSON.
Relationships	sys_relationship	Extend tables, reference records in another table, create many-to-many

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Supported file types for code search (continued)

File type	Code search table name	Description
		relationships, and join tables in a database view.
Scheduled script executions	sysauto_script	Define automated, server-side script logic that executes at a specific time or on a recurring basis. Scheduled script executions are also referred to as scheduled jobs.
Script actions	sysevent_script_action	Contain scripts that run when an event occurs, for example: • Approval is canceled • Change is approved • Problem is assigned
Script includes	sys_script_include	Contain reusable functions or classes that run only when called by other scripts, often business rules. Any server script that is complicated or reusable should be a Script Include, especially complicated business rules.
Schedule items	sys_trigger	Contain the back-end data for the System scheduler, where scheduled jobs are created, queued up, and executed. Schedule items can execute scheduled jobs, business rules, inactivity monitors, service level agreement (SLA) calculations, metric events, upgrades, and more. Access schedule item records to troubleshoot scheduling.
Defining UI actions	sys_ui_action	Include the buttons, links, and context menu items on forms and lists. Configure UI actions to make the UI more interactive, customized, and specific to user activities.
UI macros	sys_ui_macro	Contain discrete scripted components that administrators can add to the user interface. UI macros are typically controls that provide inputs or information not provided by existing field types. By default, the

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Supported file types for code search (continued)

File type	Code search table name	Description
		system provides UI macros for a variety of user interface elements.
UI pages	sys_ui_page	Create and display forms, dialogs, lists, and other UI components.
Using UI policies	sys_ui_policy	Define the behavior and visibility of fields on a form. Fields can be one of the following: • Mandatory • Visible • Read only The following apply to usage: • UI policies are always attached to one table. • UI policies often have a condition that must be true for them to run.
UI scripts	sys_ui_script	Store client scripts for reuse. Run only when called from other scripts.
UI style	sys_ui_style	Declare individual CSS styles for a field in a list or form.
Widgets	sp_widget	Describe objects that contain content and can be added to or embedded in portal pages. Use the base system widgets provided with the Service Portal, clone and modify widgets, or develop custom widgets to fit your own needs.

Supported preview file types

Reference the file types that ServiceNow Studio supports when using preview features to confirm or update user-facing page functionality.

The following table lists the file type supported by the default ServiceNow Studio preview.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Supported file type for preview

File type	Preview table name	Description
UI pages	sys_ui_page	UI pages can be used to create and display forms, dialogs, lists, and other UI components. Use UI pages as widgets on dashboards. To find the UI pages, navigate to System UI > UI Pages. This functionality requires a knowledge of HTML or Jelly. You can also create simple AngularJS applications using UI pages.

Collaboration permissions for ServiceNow Studio

Collaboration permissions determine what delegated developers can do when working on an app in ServiceNow Studio. Admins or app owners set these permissions.

For more information, see Collaborating on apps using ServiceNow Studio.

File types custom collaboration permissions

File type collaboration permissions

Permission	Description	Owner default setting	Editor default setting
All File Types	Access to all file types, including additional file types not granted by the other options.	Yes	Yes
Integrations	Access to web service APIs, REST APIs, data sources, and Integration Hub - Import.	Yes	Yes
Reporting	Access to reports and scheduled reports.	Yes	Yes
Mobile builders	Access to build mobile experiences, such as with Mobile App Builder.	Yes	Yes
UI Builder	Access to UI Builder to build complex interfaces.	Yes	Yes

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

File type collaboration permissions (continued)

Permission	Description	Owner default setting	Editor default setting
Workflow	Access to Workflow Editor and Activity Creator.	Yes	Yes
Service Portal	Access to Service Portal editors and tools.	Yes	Yes
Workflow Studio	Access to the Workflow Studio design environment to create flows and actions.	Yes	Yes
Service Catalog	Access to catalog-related file types such as catalog items, record producers, and variables to add catalog items to apps.	Yes	Yes
Tables & forms	Access to model and layout-related file types such as table columns, form layout, and list layout.	Yes	Yes
Decision Tables	Access to decision tables to create decision logic based on multiple if-else rules.	Yes	Yes
Playbooks	Access to the Playbooks design environment to create processes. Editing activity subflows or actions requires the **Flow Designer** permission.	Yes	Yes
Notifications	Access to create automatic email notifications and work with email templates.	Yes	Yes

Security/Entitlement custom collaboration permissions

Manage access control grants access to security management files, such as Access Control Lists and roles.

The default setting for both owners and editors is selected.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Programming tools custom collaboration permissions

Allow scripting grants access to script fields, such as those in Business Rule, UI Action, and Client Script.

The default setting for both owners and editors is de-selected.

Application management custom collaboration permissions

Application management collaboration permissions

Permission	Description	Owner default setting	Editor default setting
Delete application	Access to delete the app.	Yes	No
Manage collaborators	Access to change the developers and their permissions to collaborate on an app.	Yes	No
Source control	Access to source control tool integrations.	No	No
Invite collaborators	Access to invite developers to collaborate on an app.	Yes	Yes

Deployment custom collaboration permissions

Deployment collaboration permissions

Permission	Description	Owner default setting	Editor default setting
Upgrade app	Access to upgrade applications.	No	No
Submit for deployment	Access to request deployment for an app.	Yes	No
Publish app to repo	Access to publish the app to your repo.	No	No
Publish to app store	Access to publish the app to your app store.	No	No

Custom Application record form

The Custom Application form enables you to add or modify information about a custom application to define its behavior.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Custom Application form

Field	Description
Name	[Required] Enter a label for the application. Changing the name does not change any other field value derived from the application name such as the **Scope**, **Menu**, or **User role** fields.
Version	Enter version information for the application. Both the ServiceNow application repository and the store use this value to determine whether updates are available to your application users. **Note:** To publish the application in the ServiceNow Store, the version must conform to the application certification standards.
Application scoping	
Scope	[Read Only] Displays the unique application scope set during the creation process. You can change this value only by deleting and recreating the application with a new value. For more information about the protections offered, see Application scope.
Application administration	Select whether to protect sensitive application data by restricting how users acquire application-specific roles. See Application administration.
Can edit application in Studio	This field is a legacy option that does not apply to the new ServiceNow Studio.
Design and runtime	
JavaScript Mode	The JavaScript standard that the application supports. Select **ECMAScript 2021 (ES12)** to support features in ECMAScript 12th edition or **ES5 Standards Mode** to support features in ECMAScript 5th edition. Select **Compatibility Mode** to support earlier ECMAScript editions. For more information, see Set the JavaScript mode for an application.
Runtime Access Tracking	Select how the application handles script access requests to resources in other applications. Select **None** to authorize all access requests to cross-scope resources without logging them. Select **Tracking** to log and authorize all access requests to cross-scope resources. Select **Enforcing** to log access requests to cross-scope resources but require an administrator to authorize each request.
Restrict Table Choices	Clear the option to allow the application to see tables from other application scopes. Select the option to restrict design choices to only tables in the same application.
Subscription Management	
Licensable	Specifies whether the application is tracked by the Subscription Management application.
Subscription requirement	Not applicable for ServiceNow customers who build custom applications for their own use. Used only by partners who sell and monitor the usage of resellable applications on the ServiceNow Store. Specifies whether the application requires a separate subscription (**Required**) or is monitored only.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Custom Application form (continued)

Field	Description
Subscription model	Not applicable for ServiceNow customers who build custom applications for their own use. Used only by partners who sell and monitor the usage of resellable applications on the ServiceNow Store. Specifies how the Subscription Management application tracks usage. See Types of subscriptions in Subscription Management.
Primary Menu	
Menu	Select the application menu where you want to display modules. For more information about menus and modules, see Create an application menu.
End user access	
User role	Select the user role required to access the application menu. For more information about user roles, see Create a role Create a role.
Short description	Enter a description of the application purpose or usage.
Logo	Select the image the system displays in the applications list and ServiceNow Store.
Guided Setup	Add the Guided Setup to start when the application is installed or upgraded.
Application Files	View configuration records associated with this application in platform feature tables.
Dependencies	View or add tables or applications on which this application depends. The system automatically adds records to this list when you extend tables or when another application creates application files for this application. Add script-based dependencies. See Dependencies for custom applications.
Cross scope privileges	View or create cross-scope privilege records to determine which script operations and targets the system allows to run. See Cross-scope privilege record.
Design Access	View or specify which other applications have design access to tables or records in this application. See Application design access record.

Test Agent

Use Test Agent to streamline the process of authoring, running, and troubleshooting tests autonomously from a single interface. Test Agent generates test coverage for code produced by Build Agent, executes tests, and performs root cause analysis on failures, enabling end-to-end build-to-test workflows within a single development session.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Get started

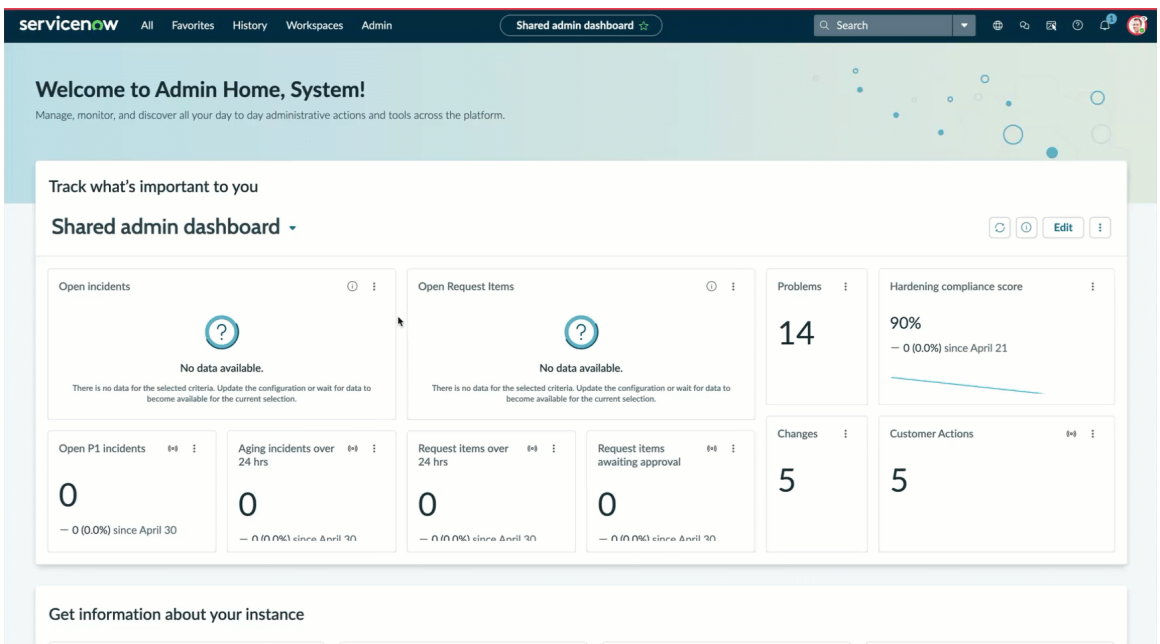
Explore Learn about Test Agent.	Configure Install Test Agent to utilize its capabilities.
Use Use Test Agent to experience one-stop shop on a single interface.	Reference Learn miscellaneous information about Test Agent.

Exploring Test Agent

Test Agent autonomously manages end-to-end test authoring, execution, and troubleshooting from a single prompt.

Test Agent overview

By extending Build Agent, Test Agent uses the same prompt and code context to generate unit and functional ATF tests, run the tests, and automatically triage failures.



Test Agent delivers the following measurable outcomes:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Build and test together: Author code and generate tests sequentially in a single Build Agent session.
- Faster failure resolution: Automated root cause analysis and recommended fixes reduce test failure investigation time. When a test fails, Test Agent performs root cause analysis and either automatically applies safe fixes or provides actionable guidance in the chat panel, enabling issue resolution without leaving the ServiceNow IDE.
- Greater release confidence: Automated tests enforce quality gates and validate code health before promoting to production.
- Seamless test reuse and execution: All generated ATF tests are stored in the standard sys_atf_tests table within the target application scope and can be reused to schedule and run regression tests, just like manually authored ATF tests.

Test Agent can generate ATF tests that use list and related list test steps, including validate related list visibility and apply filter to list tests. List step support extends test coverage beyond form-based interactions to include list view interactions on the ServiceNow AI Platform. For more information on list and related list test steps, see List and Related List.

Test Agent users

AI agents use role masking to determine which users can access them. Ones installed with your applications have specific roles that come included with the application. If you select *`Users with specific roles`* for user access, you must configure the security controls to include these roles. For the instructions to change the security controls, see Define security controls for an AI agent.

Test Agent has the following users.

Users	Description
admin	This role is required to access Test Agent and its capabilities.

Test Agent benefits

Benefit	Feature	Users
One-stop shop experience	Author, execute, and troubleshoot tests and test suites with Test Agent	admin

What to explore next

To learn more about configuring and using Test Agent, see:

- Test Agent access
- Author, execute, and troubleshoot tests and test suites with Test Agent
- Test Agent guidelines

Test Agent access

Learn about the prerequisites required to access and use Test Agent.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you can use Test Agent, your instance must meet specific prerequisite requirements to support test generation, execution, and troubleshooting workflows.

Ensure that the following applications are installed:

- Build Agent: See Install Build Agent for more information.

  Note: If you install Build Agent, Test Agent is installed automatically as part of Build Agent.

- ATF Test Generator and Cloud Runner: Install the ATF Test Generator and Cloud Runner store application from ServiceNow store. You also need to set up the cloud user for seamless execution from the Build Agent interface. See ATF Test Generator and Cloud Runner and Set up cloud user for more information.

These components provide the underlying capabilities required for Test Agent to function correctly.

Author, execute, and troubleshoot tests and test suites with Test Agent

Use Test Agent to significantly reduce the skills and resources needed to troubleshoot test and test suite failures on covered metadata.

Before you begin

Role required: admin

Procedure

1. Navigate to **All** > **App Development** > **ServiceNow IDE**.

   Note: You can also navigate to ServiceNow Studio in place of ServiceNow IDE.

   The ServiceNow IDE or ServiceNow Studio interface shows up depending on your navigation selection.
2. Select the + icon under the ServiceNow Otto window on the right to start a new chat.
3. Enter a prompt in the Build Agent chat panel to create a test/test suite or an application, or to run a new or existing test/test suite.
   You can achieve the following:
   - Write ATF tests/test suites in the global scope by authoring tests/test suites from simple prompts, such as validating that all mandatory fields are completed before submitting the Incident form.
   - Execute existing ATF tests/test suites.

     Note: If you want to execute an existing ATF test/test suite using Test Agent, you're expected to create the ATF test/test suite before this process. You can now use ServiceNow Otto to create an ATF test/test suite. Select the **Create with Otto** button on the ATF tests/test suites list page. If you have Build Agent installed on your instance, you are redirected to the ServiceNow IDE interface. You will see the following message if you don't have Build Agent installed on your instance.

     See Create new test for more information.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Troubleshoot ATF test/test suite failures.
- Use Test Agent, an extension to building or editing apps with Build Agent, to generate ATF tests/test suites for newly built or edited functionality, run those tests/test suites, and troubleshoot failures directly within the ServiceNow IDE or ServiceNow Studio context.

The Test Agent starts the process as mentioned in the prompt.

As you enter a prompt, the interface page displays the associated code for an existing test/test suite or the code generated from the prompt.

4. Review the information in the Test Agent console.

   Note: The content displayed in the Test Agent console depends on the tasks specified in the prompt.

   For example, if you're executing a test/test suite, the process first looks up the sys_id, then builds and installs the latest code. After the build succeeds, it installs the latest changes on the instance and runs the test/test suite in the background.

   Note: If the test/test suite succeeds, the message saying that the test/test suite passed shows up on the console.

5. **Optional:** If you're creating an application, generate tests/test suites based on the business rules and conditions defined in the prompt.
   The Test Agent starts the process as mentioned in the prompt. You can then review the information in the console as mentioned in the previous step. Once the tests/test suites are created, you can run them through another prompt.
   Each test/test suite displays a message indicating whether it passed or failed.
6. **Optional:** Follow the automatic triaging process if the test/test suite execution fails.
   When a test/test suite fails, the Test Agent automatically starts the troubleshooting process. It identifies the step where the failure occurred and analyzes the root cause. The Test Agent then recommends possible fixes. After you select a recommended option, the Test Agent applies the fix, updates the test/test suite code, and reruns the test/test suite automatically. When the test/test suite passes, it provides a summary of the changes made to resolve the failure.

ATF test generation in Build Agent

The Build Agent can generate Automated Test Framework (ATF) tests while you create or edit applications. This capability helps validate your builds with consistent, automated test coverage.

Key benefits

ATF test generation in Build Agent provides these benefits:

- Automated test creation during application development reduces manual test authoring effort
- AI-driven test generation ensures consistent test patterns and improved coverage
- Failed tests are automatically triaged to help you quickly identify and address issues
- User consent gates ensure applications only generate tests when explicitly authorized

How it works

The following information describes how ATF test generation integrates with the Build Agent workflow:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- After successfully building and installing an application, the Build Agent displays a consent prompt asking whether you want to generate automated tests for the application
- If you accept, the Build Agent generates test cases using ATF patterns and best practices
- Generated tests run through quality validators to ensure consistency and coverage
- You can opt out entirely with a feature flag if your team doesn't use automated testing

User workflow

The ATF test generation workflow follows this sequence:

1. You create or edit an application using the Build Agent
2. You complete the build and install process
3. The Build Agent displays a consent prompt: "Generate automated tests for this application?"
4. You choose one of the following:
   - **Yes** — Proceed with test generation
   - **No** — Skip test generation for this build
   - **Remind me later** — Defer the decision until the next build
5. If you select **Yes** and after test generation completes, the Build Agent automatically executes the server-side tests. If the Run UI ATF tests setting is enabled, the client-side UI tests run automatically as well. You can also manually trigger UI tests at any time.

Enable ATF test generation in Build Agent

Enable the Build Agent to generate Automated Test Framework (ATF) tests automatically when you build and install applications.

Before you begin

- The Automated Test Framework (ATF) plugin is installed and activated on your instance
- The Build Agent is configured and available in your development environment

Role required: admin

About this task

When ATF test generation is enabled or have auto-approve setting turned on, the Build Agent automatically offers to create automated tests each time you build and install an application. This reduces manual testing effort and verifies consistent test coverage.

Procedure

1. In the IDE, open the ServiceNow Otto panel on the right side and navigate to **Settings** > **General**.
2. Verify the Build Agent settings for ATF test generation:

   Both settings are disabled by default. Review the following settings in Build Agent:

Build Agent ATF test generation settings

Setting	Default	Purpose
Sync ATF tests with app	Disabled	Generates ATF tests when Build Agent creates an app, and keeps them synced when the app is edited.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Setting	Default	Purpose
Run UI ATF tests	Disabled	Runs the client-side UI ATF tests and the server-side tests. Requires **Sync ATF tests with app** to be enabled. UI test runs are slower.

3. To enable ATF test generation, set **Sync ATF tests with app** to enabled.
   - **Enabled:** Build Agent generates ATF tests when you create or edit applications.
   - **Disabled (default):** ATF test generation is turned off.
4. **Optional:** To run client-side UI ATF tests, set **Run UI ATF tests** to enabled.
   - **Enabled:** Build Agent runs both server-side and client-side UI ATF tests. UI test runs are slower.
   - **Disabled (default):** Only server-side tests are run.

   Note: This setting requires **Sync ATF tests with app** to be enabled.
5. Save your settings.
6. Test the configuration by creating or editing an application in the Build Agent.
   - Complete the build process
   - After installation, verify that the consent prompt appears: "Generate automated tests for this application?"
   - Select **Yes, proceed** to confirm your setting is enabled

Result
After you complete these steps, ATF test generation is enabled in Build Agent. The next time you build and install an application, you will receive a consent prompt asking whether to generate automated tests. The Build Agent creates tests based on your application structure and the configured feature flags.

UI Test Script in Automated Test Framework (ATF)

UI Test Script enables you to create test code for custom user interfaces exposing elements of Testing Library within the Automated Test Framework (ATF). Generate test scripts through conversational interaction with Build Agent and run them alongside other ATF tests.

Key benefits

UI Test Script provides these benefits:

- Agentic authoring through conversational interaction reduces manual test code writing effort
- Test API exposes elements of Testing Library, a widely-used UI testing library with consistent patterns
- Test scripts integrate seamlessly with ATF framework and run in the client test runner
- Line-level success/failure indicators help you quickly identify and fix failing assertions
- Conversational editing allows you to refine test scripts through natural language interaction with Build Agent

How it works

UI Test Script operates in two phases: authoring and execution.

Authoring Phase:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- You interact conversationally with Build Agent to describe the UI test you want to create
- The UI Test Script Skill generates test code using Testing Library syntax
- Generated test scripts can be packaged as a single "UI Test Script" step in ATF, or included alongside existing steps
- You can refine tests by providing conversational feedback through the Build Agent panel

Execution Phase:

- UI Test Scripts run as part of the ATF framework in the client test runner
- The overall test passes or fails at the ATF step level
- Line-level indicators within the test code show which specific assertions succeeded or failed

Test results and feedback

When UI Test Scripts run, you receive results at two levels:

- Step level: The overall "UI Test Script" step passes or fails in ATF based on the cumulative result of all assertions
- Line level: Inside the test code, individual assertions show success or failure indicators, helping you pinpoint which checks passed and which failed

Troubleshooting

The troubleshooting process in UI Test Scripts is manual:

- Review UI Test Script logs to see execution details
- Check test run screenshots to visually verify what the test was attempting
- Modify the test script conversationally through Build Agent if changes are needed

Availability

UI Test Script is available in the following environments:

- ServiceNow Studio (SNS)
- IDE (Integrated Development Environment)
- Standard ATF forms and lists (for non-conversational creation)

Create a UI Test Script in Build Agent

Create test scripts for custom user interfaces using conversational interaction with Build Agent. UI Test Scripts expose elements of Testing Library and run as part of the Automated Test Framework (ATF).

Before you begin

- Build Agent is available in your ServiceNow Studio instance or IDE
- You have access to ATF and Testing Library capabilities

Role required: admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

About this task

UI Test Script enables agentic authoring through conversational interaction. Instead of manually writing test code, you describe your testing needs to Build Agent and it generates Testing Library code for you.

Procedure

1. In ServiceNow Studio or the IDE, open Build Agent.

   Build Agent is the agentic authoring interface for creating tests and test scripts.

2. Navigate to the UI Test Script authoring section.
3. Describe your test requirements conversationally to Build Agent.

   Provide a natural language description of what you want to test. For example:

   - Create a test that fills out a user profile form with name and email, then selects the **Submit** button
   - Test that the error message appears when a required field is left empty
   - Verify that the dashboard loads and displays the correct widgets
4. Review the generated test script code.

   Build Agent generates test code using the UI Test Script API supporting a subset of Testing Library syntax. The code appears in an editor or panel where you can review it for accuracy.

   - Correct element selectors targeting your UI components
   - Appropriate Testing Library methods for user interactions
   - Meaningful assertions that validate expected behavior
5. **Optional:** Refine the test script through conversational feedback.

   (Optional) If the generated script needs changes, provide conversational feedback to Build Agent. For example:

   - Add an assertion for the success message after submit
   - Change the email field selector to use the aria-label attribute
   - Add a wait before checking the result

   Build Agent updates the test code based on your feedback.
6. Package the UI Test Script as an ATF step.

   Choose how to integrate the test:

   - **Single step:** Package the entire UI Test Script as one ATF step
   - **With other steps:** Include the UI Test Script alongside other ATF test steps in the same test case
7. Save the test script.
8. Run the test to verify it works correctly.

   Execute the test through ATF. The test runs in the client test runner and produces:

   - Overall pass/fail result at the ATF step level
   - Line-level success/failure indicators for each assertion in the test code

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Result
After you complete these steps, you have created a UI Test Script that:

- Uses Testing Library syntax to interact with and validate your custom UI
- Runs as an ATF step alongside other tests
- Produces pass/fail results with line-level detail showing which assertions succeeded or failed
- Can be refined further through conversational feedback with Build Agent

If the test fails, review the UI Test Script logs and screenshots to understand what went wrong, then use conversational feedback to update the test script.

Test Agent references

The reference topics provide additional information about Test Agent.

Test Agent guidelines

Leverage the full potential of Test Agent by following these guidelines.

Supported metadata

Test Agent supports the following metadata:

- sys_ui_action
- sys_ui_policy
- sys_ui_policy_action
- sys_ui_policy_rl_action
- sys_script_client
- sys_ui_script
- sys_data_policy2
- sys_ui_action_role
- sys_ui_action_view
- sys_ui_base_theme
- sys_ui_form
- sys_ui_formatter
- sys_ui_macro
- sys_ui_message
- sys_ui_related_list
- sys_ui_section
- sys_ui_view
- sys_ui_title
- sys_ui_style
- sys_ui_theme
- sys_ui_page
- sys_ui_context_menu
- sys_ui_list
- sys_ui_list_control

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- sys_ui_list_control_embedded
- sys_ui_list_script_client
- sys_ui_list_script_server
- sys_ui_related_list
- sys_user_role
- sys_app_application
- sp_widget
- sp_instance
- sys_security_acl_role
- sys_script
- sys_app_module
- sysevent_email_action
- sys_report
- sys_security_acl
- catalog_ui_policy
- catalog_ui_policy_action
- sp_theme
- sp_widget
- sys_sg_universal_link_path_segment
- sys_sg_sections_screen_m2m_section
- sys_sg_location_tracking_details
- sys_highlighted_value_condition
- sys_sg_card_size
- sys_sg_custom_map_config
- sys_sg_empty_state
- sys_aw_form_header_secondary_values
- sys_aw_registered_scripting_modal
- sys_template
- sys_sg_alert
- sys_aw_form_header
- sys_script_include
- sys_scriplet_function
- sys_ui_extension_instance
- sys_playbook_experience_action_assignment_map
- sys_df_data_filtration
- sys_email_address_filt_domain
- sys_atf_step
- sys_atf_test

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- sys_atf_step_config
- atf_input_variable

Exceptions

Test Agent doesn't support the following metadata:

- Flows
- Configurable workspace steps
- Custom UI test steps
- Custom test steps

Scope and availability

Test Agent is available in the following environments and scopes:

Test Agent scope and availability

Dimension	Supported values
Authoring environment	ServiceNow IDE or ServiceNow Studio
Application scope	Global, custom, store
Test types	ATF unit tests, ATF functional tests
Execution target	Cloud runner lanes

Example prompts

- Write an ATF test in the global scope to validate that an employee can order a laptop from the Service Catalog.
- Write an ATF test to validate that internal IT knowledge articles are not visible to self-service users.
- Write an ATF test to validate that all mandatory fields on the Incident form are completed before submission.

Note: The examples above are prompts for authoring ATF tests in the global scope. To run a test, prompt to execute the ATF test by specifying its name or sys_id. To troubleshoot a failure, prompt to triage the failed ATF test by providing the test name or the sys_id of the test result.

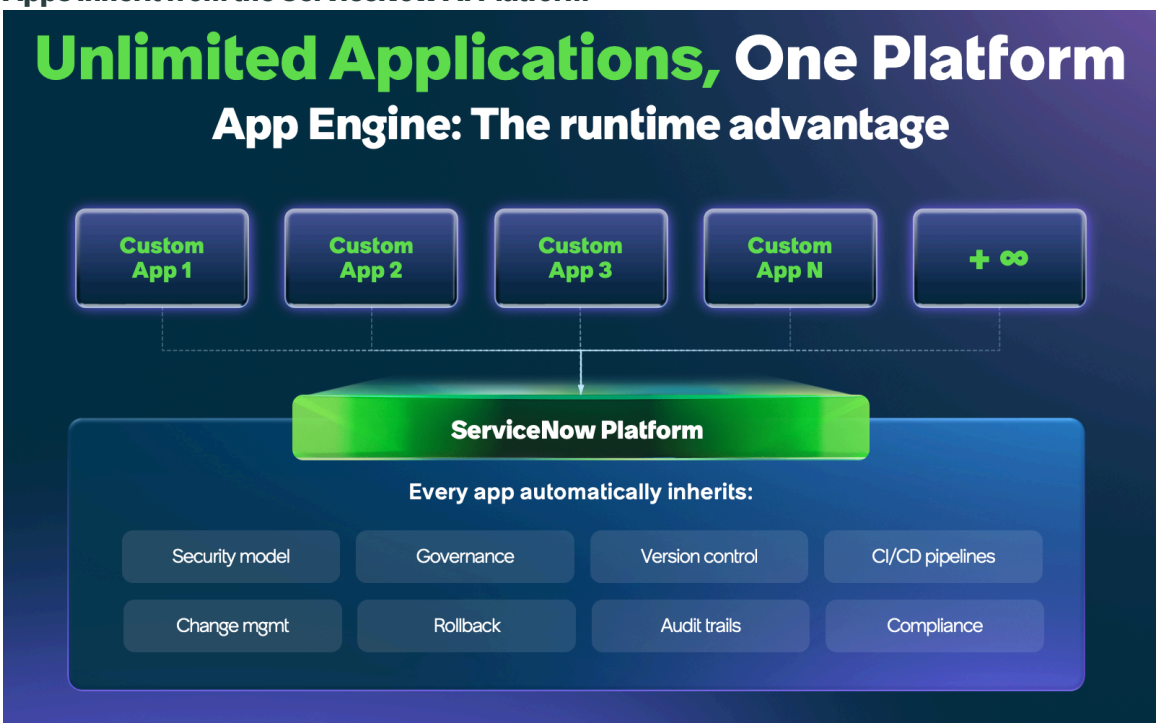
Agentic development on the ServiceNow AI Platform

Agentic development, or vibe coding, is an AI-driven approach to application development. Use agentic development and ServiceNow agentic-assisted app building tools to describe your goals in natural language, and have the ServiceNow AI Platform generate full-stack applications, workflows, and integrations. The ServiceNow AI Platform automatically incorporates governance into the app creation process.

Autonomous app development on the ServiceNow AI Platform is agentic development.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Apps inherit from the ServiceNow AI Platform



Agentic development with the ServiceNow AI Platform collapses the traditional app development lifecycle—from ideation to deployment—into minutes instead of weeks.

Note:

- This document intends to provide a holistic view of agentic development on the ServiceNow AI Platform. It is not a document for a specific product, but instead compiles resources and links off to other products, agents, and skills for you to learn more and get started with agentic development faster.
- Agentic development encompasses a range of approaches to AI-assisted app building:
  - In one approach, developers provide a goal and deploy the generated output without reviewing the underlying code.
  - In another approach, agentic development is an iterative, conversational process in which developers guide AI generation, review the output, and retain responsibility for the generated code.

  ServiceNow provides tools to help you review, refine, and test apps throughout agentic development, including Build Agent and ServiceNow Otto for Creator.

Key features and benefits of agentic development on the ServiceNow AI Platform

Agentic development with the ServiceNow AI Platform features and benefits

Benefit	Feature	Role
Full-stack development	Build Agent autonomously generates full-stack applications while comprehending the ServiceNow data model and leveraging	admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Agentic development with the ServiceNow AI Platform features and benefits (continued)

Benefit	Feature	Role
	the controls and integration capabilities of the ServiceNow AI Platform.	
Built-in security and governance	Governance includes code optimization, ACL enforcement, App Engine Management Center, and security checks for generated scripts.	admin
Context-aware	ServiceNow AI Platform agentic development tools recognize concepts, tables, business rules, and workflows specific to ServiceNow.	now.assist.creator
Versatile tasks	Beyond building new apps, ServiceNow AI Platform agentic development tools can explain code, debug issues, rewrite tables, validate applications, and fix errors.	admin or delegated_developer
Self-healing code	Build Agent can automatically analyze and repair errors without human intervention.	admin or delegated_developer

Why use agentic development on the ServiceNow AI Platform

You can develop agentically with any number of products on the market. However, when you practice agentic development on the ServiceNow AI Platform, you can take advantage of the backend, hosting, and built-in governance. Use the contextual awareness of the ServiceNow AI Platform and all its built-in capabilities as a trusted place to develop and host apps developed agentically.

The ServiceNow AI Platform agentic development tools have direct access to the existing data models, ACLs, business rules, and automation patterns in your instance. When you use ServiceNow AI products to create an application, the tools already have access to your table structures, security setup, and how workflows are configured in your instance. This context depth enables you to generate code that fits your environment rather than generic code that needs adapting.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Additionally, governance is built in to ServiceNow AI Platform agentic development and vibe coding tools, so your applications are secure. For more information, see Governance for agentic development.

Your app doesn't have to be completely developed inside the ServiceNow AI Platform to take advantage of its offerings. A standalone app that's agentically developed can still connect to a ServiceNow AI Platform data model. Your end users don't have to know it's ServiceNow, for example, if the app looks completely different, fulfillers can still use workspaces to track requests from a Service Portal.

Get started

Explore  Explore agentic development on the ServiceNow AI Platform.	Onboard  Browse information on onboarding with agentic development.
Use  Create, test, and deploy applications with agentic development tools.	Reference  Reference information for agentic development, such as prompt examples.

Helpful resources

Some ServiceNow resources that can provide you with helpful information are:

Getting started guide

Build Agent docs

Build Agent

Community Build Agent announcement

Vibe coding at scale is here in the ServiceNow AI Platform

Vibe coding with ServiceNow Build Agent

Build Agent introductory video

App Engine Academy recordings

App Engine Academy playlist on YouTube

Support

Contact Customer Service and Support

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

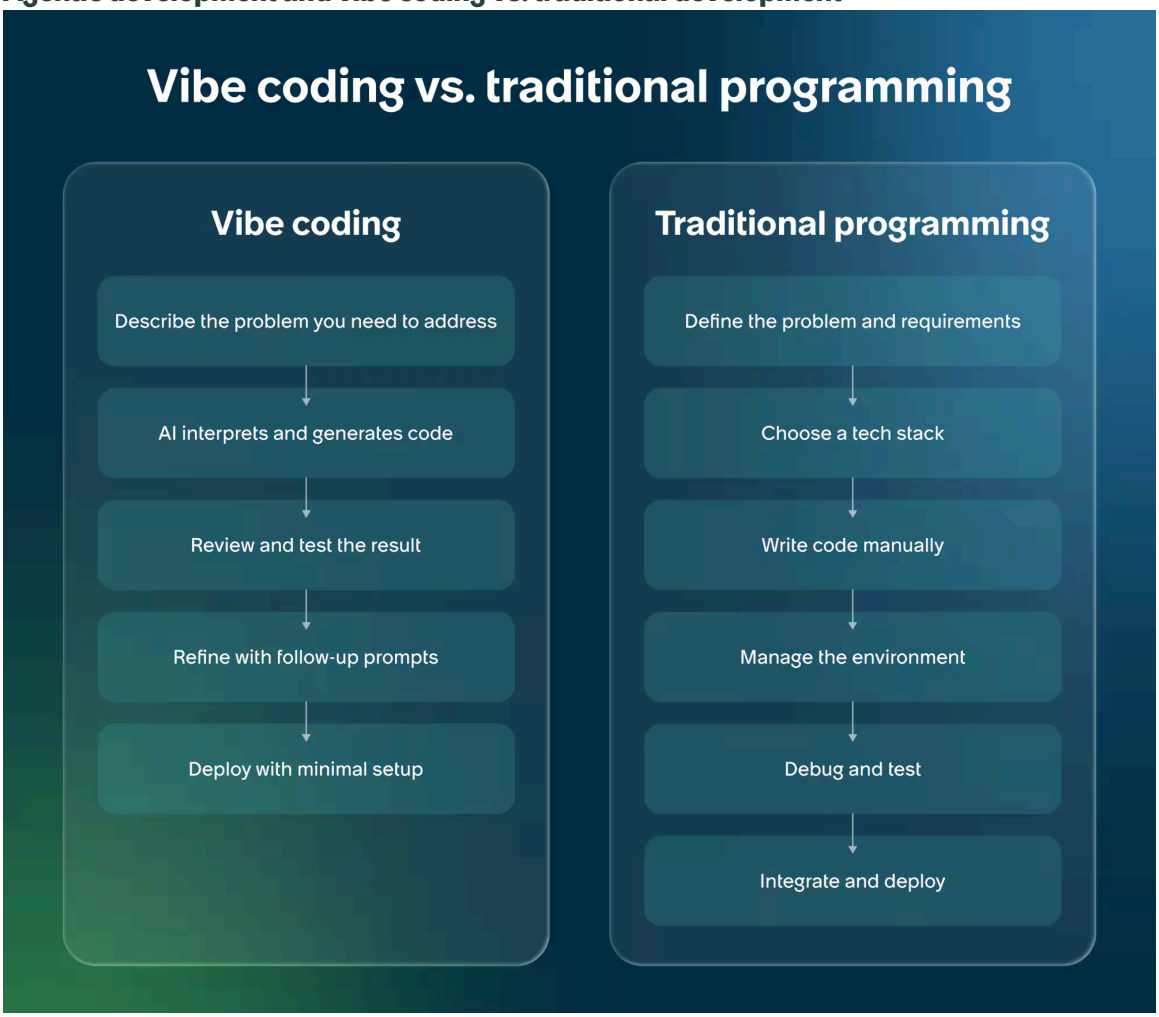
Exploring agentic development

To build apps agentically on the ServiceNow AI Platform, you interact with tools through a multi-turn conversation interface to build apps, write documentation, or brainstorm ideas. Describe what you want in plain language, and the ServiceNow AI Platform generates complete applications including logic, UI, configuration, and tests.

Agentic development overview

The ServiceNow core offerings for agentic development include several products, such as Build Agent, ServiceNow Otto for Creator, built-in security and governance, and an enhanced developer experience.

Agentic development and vibe coding vs. traditional development



Build Agent and agentic development

Build Agent can do the following:

- Convert plain language prompts into production-ready apps.
- Handle UI design, database schema, business logic, integrations, and testing automatically.
- Include audit trails, security controls, and governance for enterprise compliance.
- Build a custom AI agent, agentic workflow, or AI skill for the ServiceNow AI Platform.
- Integrates with ServiceNow Otto for Creator and App Engine for enterprise-grade workflows.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ServiceNow Otto for Creator and using AI to build out apps

While not exclusively an agentic development tool, ServiceNow Otto for Creator can do the following for agentic development:

- Provide AI-powered developer tools embedded in ServiceNow Studio and ServiceNow IDE.
- Generate JavaScript for server-side and client-side logic to enable text-to-code.
- Convert complex scripts into human-readable explanations and documentation.
- Suggest secure and efficient alternatives for code refactoring and optimization.
- Create forms and app scaffolding from natural language prompts.
- Provide a chat panel that offers a multi-turn conversational interface for building apps and querying data.

Security and governance with agentic development

Security and governance enforces enterprise standards for every app with built-in compliance, and includes the following:

- App Engine Management Center enforces compliance with corporate policy through its lifecycle management tools.
- AI Control Tower serves as the central governance hub for all AI activities across the enterprise, including agentic development workflows.
- ServiceNow Vault Console enables you to discover and protect sensitive data across workflows.
- Machine Identity Console helps you govern API connections and authenticate AI agents.

For details on security and governance and agentic development, see Governance for agentic development.

Enhanced developer experience

An enhanced developer experience for agentic development is available in the following ways:

- Build Agent is available in several development products, including ServiceNow Studio and the ServiceNow IDE.
- ServiceNow Fluent is a declarative, TypeScript-based language designed specifically for ServiceNow metadata that enables AI to communicate effectively with the ServiceNow AI Platform.
- Build agentically in your choice of development environments, such as:
  - An isolated sandbox using Developer Sandboxes, which enables safe spaces for testing and collaboration.
  - A Personal Development Instance (PDI), which is a non-production instance.
- Tools like the ServiceNow SDK can work in conjunction with tools like Claude Code to interface with your ServiceNow instance when working locally.
- AI agents can analyze and repair code issues through debugging and automated error-fixing.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Agentic development users

Users for agentic development

User	Description
Admin	Admins can use Build Agent and other agentic development tools to generate an app using the conversational interface.

What to explore next

To learn more about agentic development on the ServiceNow AI Platform, see:

- Onboarding for agentic development
- Agentic development on the ServiceNow AI Platform
- Agentic development reference

Who should use this guide

This guide provides a holistic view of agentic development on the ServiceNow AI Platform. It helps you understand the range of agentic development approaches available, from fully automated code generation to agentic development workflows.

Intended audience

This guide is for a range of people building apps agentically on the ServiceNow AI Platform. The intended audience is:

- ServiceNow developers who want to use AI to accelerate app development on ServiceNow. You may be:
  - New to ServiceNow but experienced with full-stack development
  - Familiar with agentic coding tools and wanting to understand the capabilities of the ServiceNow AI Platform
  - Looking to evaluate how AI can fit into your development workflow
  - Interested in understanding when to use different AI development approaches
- ServiceNow developers who want to incorporate AI into their existing workflows. You may be:
  - Experienced with the ServiceNow platform and wanting to explore AI tooling
  - Looking to improve development speed and efficiency
  - Seeking guidance on general guidelines for agentic development
  - Evaluating which AI tools best fit your use case

Note: Build Agent requires the admin role.

What this guide covers

Agentic development encompasses a spectrum of approaches:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Autonomous code generation: At one end of the spectrum, agentic development involves providing a high-level description and letting AI generate, test, and deploy code without manual review. This approach prioritizes speed and simplicity.
- Agentic development: At the other end of the spectrum, AI serves as a collaborative tool that generates code suggestions, automates repetitive tasks, and accelerates development, while developers review, refine, and maintain ownership of the final code. This approach emphasizes understanding and supportability.

ServiceNow provides tools across the development spectrum. This guide helps you understand:

- What AI development capabilities are available on ServiceNow
- Which tools align with your development philosophy and use case
- How to effectively use AI in your ServiceNow development workflow
- General guidelines for reviewing, refining, and testing AI-generated code

What this guide does not cover

This guide focuses on agentic development tools and workflows. It does not cover the following topics:

- Low-code/no-code development tools like Creator Studio or Catalog Builder
- Traditional ServiceNow development without AI assistance
- Detailed product-specific documentation (links are provided to dedicated product docs)

For comprehensive product documentation, see the dedicated documentation for Build Agent, ServiceNow Otto for Creator, and other ServiceNow development tools.

Understanding your needs

As you explore this documentation, consider the following questions:

- Your technical expertise: Are you comfortable working in an Integrated Development Environment (IDE) with a file system and reviewing generated code? Or do you prefer tools that handle more of the technical complexity?
- Your development philosophy: Do you want to understand and maintain the code AI generates, or do you prefer to let AI handle implementation details while you focus on outcomes?
- Your use case: Are you building custom applications from scratch, extending existing ServiceNow functionality, or automating specific workflows?

Different AI development approaches and tools will be better suited to different needs. This guide helps you identify the right approach for your situation.

What is agentic development

Agentic development uses AI to turn a natural language description of the outcome you want into applications using prompts in a conversation. Autonomous app development is agentic development on the ServiceNow AI Platform.

- In one approach to agentic development, you describe a problem or an application in natural language and have AI generate it without reviewing the underlying code.
- In another approach, AI tools work alongside developers to suggest code, auto-complete functions, and explain existing code, with developers reviewing and guiding the output.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Agentic development approaches



Agentic development encompasses a range of approaches to AI-assisted app building:

- In one approach, developers provide a goal and deploy the generated output without reviewing the underlying code.
- In another approach, agentic development is an iterative, conversational process in which developers guide AI generation, review the output, and retain responsibility for the generated code.

ServiceNow provides tools to help you review, refine, and test apps throughout agentic development, including Build Agent and ServiceNow Otto for Creator.

Though testing and deployment are not always part of an agentic development workflow, they are covered in this document to account for all phases of development. Governance is also included in apps that are built agentically on the ServiceNow AI Platform.

Some agentic development workflows do not include reviewing or adjusting generated code, for example in ServiceNow Studio or the ServiceNow IDE. However, this document includes information on those processes for developers who want to refine generated output further.

Agentic development enables users across skill levels to build apps by prompting, accelerating the app creation cycle.

Governance for agentic development

Agentic development on the ServiceNow AI Platform accelerates application development by using AI to generate code and configurations from natural language prompts. However, speed must not compromise security, compliance, and maintainability.

Governance addresses the following:

- Risk and compliance: AI-generated apps meet enterprise security standards and regulatory requirements.
- Quality assurance: Automated code is validated through testing and review.
- Visibility and control: Prevents shadow IT and enforces lifecycle transparency.

ServiceNow embeds security and governance directly into the agentic development workflow, so AI-generated applications meet enterprise standards by default. Build Agent automatically generates Access Control Lists (ACLs) that enforce role-based access, validates scripts for

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

security vulnerabilities, and applies code optimization during generation. Every app that's developed agentically on the ServiceNow AI Platform includes audit trails, security controls, and compliance checks without requiring explicit prompts for these features.

Note: Build Agent requires the admin role.

Governance in an agentic development workflow

AI-generated code and configurations can be produced faster than traditional development cycles, which compresses the window for review. When you describe an outcome and the AI generates the implementation, the resulting code, scripts, business rules, and access control configurations may not be immediately visible to the developer who prompted them. Governance practices fill that gap by making the outputs of agentic development auditable, testable, and aligned with organizational policy before apps reach production.

Apps built through agentic development on the ServiceNow AI Platform are scoped applications and inherit the same platform controls as any other scoped app, including ACLs, update sets, and application scope boundaries. This means governance is not an additional layer applied after the fact, but part of the development environment from the start. The tools listed in this topic connect those controls to a review and approval workflow.

Human review responsibilities

Automated checks in the agentic development workflow, including ATF tests, security scans, and compliance validation, catch many issues before deployment. However, some decisions require human judgment that automated tools cannot replace.

Review AI-generated output directly before deployment when any of the following are true.

- The app accesses sensitive tables, personally identifiable information, or regulated data.
- The app includes scripts, business rules, or integrations with external systems.
- The app creates or modifies roles, ACLs, or cross-scope privilege records.
- The generated output has not been tested against your organization's specific data or configuration.

For apps generated by Build Agent, check the session summary and any generated documentation to understand what was built before promoting the app from a sandbox to a higher environment. Use AEMC to require approvals at key lifecycle stages.

Governing apps built with AI Control Tower

Apps and agents generated through agentic development are registered as AI assets in AI Control Tower, where AI stewards can track lifecycle progression, review security posture, and assess compliance from a single workspace. Registration happens as part of the development process, so governance visibility is established before an app reaches production.

From the AI asset record in AI Control Tower, stewards can see governance health, evaluation scores, and risk classification for each generated app or agent. Security considerations specific to AI-generated assets are surfaced automatically, including agents with elevated permissions, agents that experience access-related errors, and agents that have been inactive for more than 90 days but still retain active permissions.

If your organization uses AI Risk and Compliance, apps built through agentic development can be evaluated against regulatory frameworks such as the NIST AI Risk Management Framework and the EU Artificial Intelligence Act. Risk classification and compliance posture are available on the Risk and Compliance tab of each AI asset record.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For more information, see AI Control Tower (legacy).

Core governance principles

1. Approval and oversight:
   - Use App Engine Management Center (AEMC) to approve app ideas and manage collaborators.
   - Require governance checkpoints before moving from sandbox in Developer Sandboxes to production.
2. Controlled release management with ReleaseOps implementation practices:
   - Update sets and update set automation for version control.
   - Metadata-as-code pipelines for automated deployments.
3. Secure development practices:
   - Enforce ACLs and role-based access for generated apps, which Build Agent can do.
   - Validate AI-generated scripts for security vulnerabilities.
   - Apply code optimization and review before publishing.
   - Create Cross-Scope Privileges to control which tables, scripts, and resources one scoped application can access from another. Use Cross-Scope Privileges to diagnose and resolve `operation not allowed` errors between scoped apps.
   - While creating agents and skills, Build Agent asks which users and roles it should operate as, as well as which users are allowed to access the agents or skills.
4. Support for testing and validation:
   - Use Automated Test Framework (ATF) for functional and regression testing.

     Note: If you're using Build Agent, it automatically updates failing metadata to resolve ATF test failures, without you needing to manually run ATF tests.

   - Include peer review for critical workflows and integrations.
5. Isolation of work with Developer Sandboxes:
   - Experiment and develop in Developer Sandboxes to avoid impacting production.
   - Align with Git-style branching for concurrent development.

Governance checklist for apps built with AI assistance

1. App idea approved in AEMC.
2. ACLs and security roles applied.
3. Code reviewed and optimized.
4. ATF tests executed and passed.
5. Release pipeline validated.
6. Documentation generated (such as summaries and flow explainers).
7. Compliance and audit logs updated.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Governance tools and resources

Tools and resources for governance

Tool	Description	More information
App Engine Management Center	Governance hub for approvals and monitoring.	App Engine Management Center
AI Control Tower	Monitors AI agent behavior, enforces guardrails, tracks AI-generated code changes, and provides dashboards showing which apps were created by Build Agent, what data they access, and how they comply with organizational policies.	AI Control Tower (legacy)
ServiceNow Vault	Discovers and protects sensitive data across workflows, so AI-generated apps handle confidential information appropriately.	ServiceNow Vault
ReleaseOps Toolkit	Update set automation and metadata pipelines.	ReleaseOps
Automated Test Framework	Automated testing for ServiceNow apps.	Automated Test Framework (ATF)
Developer Sandboxes	Develop in a secure, isolated Developer Sandboxes environment.	Developer Sandboxes
Knowledge Base articles	Data handling and AI usage guidelines.	Knowledge Management

Governance general guidelines

When using agentic development, prompts should not only describe functionality but also embed governance requirements. This helps generated apps comply with security, compliance, and quality standards.

See Example prompts for agentic development for example prompts for governance.

Tool comparison for agentic development

Compare ServiceNow development tools to select the right approach for your agentic development needs.

The ServiceNow AI Platform provides multiple tools for agentic development. Each tool serves different use cases, skill levels, and development philosophies. Use this comparison to select the appropriate tool for your project.

Tool comparison matrix

The following table shows a general comparison of ServiceNow agentic development tools.

Agentic development tools comparison

Tool	Best for	Skill level	Autonomy level	Primary output	Key differentiator
Build Agent	Creating full-stack	Beginner to Advanced,	High (autonomous	Complete applications	Conversational interface that

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Agentic development tools comparison (continued)

Tool	Best for	Skill level	Autonomy level	Primary output	Key differentiator
	applications from scratch with minimal manual intervention • In ServiceNow Studio: metadata-driven development with previews and controlled iterations • In ServiceNow IDE: code refinement and hardening AI-generated applications	depending on where you access: • In ServiceNow Studio: low-code builders and admins • In ServiceNow IDE: Advanced (Pro developers)	generation) to low (developer-controlled)	with UI, backend, and tests: • In ServiceNow Studio: ServiceNow AI Platform metadata (tables, flows, experiences) • In ServiceNow IDE: Code files with syntax highlighting and real-time deploy	generates production-ready apps end-to-end • In ServiceNow Studio: UI-first experience with diffs, previews, and step-by-step guidance • In ServiceNow IDE: VS Code-like experience for manual code review and optimization
ServiceNow Otto for Creator	Service Catalog item creation with variables and workflows, app generation, flow generation, playbook generation, UI generation, and many more agentic development skills	Intermediate	Medium	Catalog items, record producers, order guides For a list of included generative, development, and summarization skills, see Agentic app creation with ServiceNow Otto for Creator.	Specialized for catalog management with text-to-catalog generation and other generative and development skills
ServiceNow SDK	Local development with ServiceNow Fluent and command line interface-based workflows	Advanced (Full-stack developers)	Low (Framework support)	ServiceNow applications	Declarative TypeScript framework that enables AI to communicate with the ServiceNow AI Platform

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Workflow comparison

The following table compares workflows for ServiceNow agentic development tools.

Development workflow by tool

Phase	Build Agent	ServiceNow Otto for Creator
1. Input	Natural language prompt describing full application Options include: • Task or requirements (structured or unstructured) in ServiceNow Studio • Load existing app workspace; describe changes in ServiceNow IDE	Describe Service Catalog item with fields and approval flows
2. Generation	Autonomous full-stack generation (tables, UI, flows, tests) Options include: • Propose files to create/modify with dependency graph in ServiceNow Studio • Edit code/metadata; scaffold new components in ServiceNow IDE	Scaffold Service Catalog item with variables and fulfillment
3. Review	Approve edits; agent builds and deploys Options include: • Review diffs and summaries; approve or adjust in ServiceNow Studio • Manual code review with syntax highlighting in ServiceNow IDE	Review and harden item configuration
4. Refinement	Multi-turn prompts to add features Options include: • Iterative guided, batch, or one-shot generation in ServiceNow Studio • Manual editing with code completion in ServiceNow IDE	Iterative prompts for variables and flows
5. Testing	Automatic Automated Test Framework (ATF) test creation, editing, and running	ATF test generation for catalog submissions

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Development workflow by tool (continued)

Phase	Build Agent	ServiceNow Otto for Creator
6. Deployment	Deploy from Developer Sandboxes with audit trails Options include: • Update sets or Pipelines and Deployments with ServiceNow Studio • Command line interface deploy or Git-based CI/CD in ServiceNow IDE	Standard Service Catalog item publication workflow

Tool access

The following table shows how to access ServiceNow agentic development tools.

Where to access agentic development tools

Tool	Access method
Build Agent	Chat panel in ServiceNow Studio or ServiceNow IDE
ServiceNow Otto for Creator	Embedded in Service Catalog and Workflow Studio
ServiceNow SDK	Local VS Code or the ServiceNow IDE with ServiceNow Fluent SDK

Performance and scalability considerations

The following table shows performance and scaling considerations for ServiceNow agentic development tools.

Tool	Best performance for	Limitations
Build Agent	New applications with < 20 tables	Limited support for cross-product integration
ServiceNow Studio	Iterative metadata updates; low-code workflows	Does not support complex custom code generation
ServiceNow IDE	Complex business rules, script includes, advanced customization	Developer expertise required
ServiceNow Otto for Creator	Service Catalog items (one at a time)	Creates single Service Catalog items, not full applications
ServiceNow SDK	Local development with TypeScript	Limited platform metadata manipulation compared to ServiceNow Studio

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Workflow for agentic development

The agentic development workflow begins with developers providing conversational prompts to the Build Agent tool, which then creates tables, forms, workflows, and automation rules.

Integrated tools like Build Agent and ServiceNow Otto for Creator enhance the experience by offering text-to-code generation, code explanations, and optimization suggestions. Governance and security are embedded throughout the process, ensuring enterprise-grade compliance and scalability.

Note: Agentic development encompasses a range of approaches to AI-assisted app building:

- In one approach, developers provide a goal and deploy the generated output without reviewing the underlying code.
- In another approach, agentic development is an iterative, conversational process in which developers guide AI generation, review the output, and retain responsibility for the generated code.

ServiceNow provides tools to help you review, refine, and test apps throughout agentic development, including Build Agent and ServiceNow Otto for Creator.

Step-by-step general workflow

1. Describe your goal: Provide a clear natural language prompt (for example, `Create an application that handles several types of issues related to cash management. Create a custom data model with tables for each type of request. Add fields that are typical for these types of issues to each table. Request tables should extend the Task table. Create custom states for each request type to track issue resolution.`).

   Tip:

   - If you're using Build Agent, include as much context as possible in your first prompt to enable more robust development. Include roles, data requirements, and success criteria.
   - You can craft a prompt and then feed it to an LLM, such as Claude, and ask for it to be made more specific for ServiceNow.

   For more information on example prompts, see Example prompts for agentic development.

2. Generate the plan: Build Agent interprets your prompt and creates a blueprint, including tables, roles, UI components, and workflows.

   Benefit: Architectural clarity before coding begins. For more information, see Vibe coding: Chatting your way to a new app with ServiceNow Build Agent.

   For larger implementations with well-defined requirements, Autonomous Engineer can replace this step. It conducts an interview, generates a structured plan with work items, and builds all work items in parallel. For more information, see Autonomous Engineer in Build Agent.

3. Build and generate code: The ServiceNow AI Platform autonomously creates full-stack components using ServiceNow Fluent, JavaScript, and ServiceNow metadata.

   Includes: UI pages, data models, business rules, and integration points. For more information, see Vibe coding with ServiceNow Build Agent: The future of AI-powered app development.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

4. Validate and self-heal: If you're using Build Agent, it generates automated tests and fixes errors without manual intervention.

   Governance: Compliance and security checks are applied automatically. For more information, see Governance for agentic development.

5. Refine through prompts: Add features or adjust design iteratively (for example, `Add a Kanban board` or `Improve UI layout for accessibility`).

   For more information, see Vibe Coding: Chatting your way to a new app with ServiceNow Build Agent on Community.

6. Run tests: Run Automated Test Framework to test that the app is working as expected.

   For more information, see Testing agentically developed apps.

7. Deploy securely: Push from a sandbox in Developer Sandboxes with audit trails and enterprise-grade security via Vault Console and Machine Identity Console.

   For more information, see Agentic development and deployment.

General guidelines for agentic development

General guidelines for agentic development on the ServiceNow AI Platform cover prompt writing, context management, compliance validation, and development environment setup.

Agentic development guidelines

- Use clear, focused prompts with as much detail as possible to generate more accurate and relevant output.
- If you're using Build Agent, include as much context as possible in your first prompt to enable more robust development. Include roles, data requirements, and success criteria.
- When switching between two separate applications in Build Agent, specify the new application context.
- Save your successful prompts to repurpose and reuse. For more information, see Example prompts for agentic development.
- Save your session to quickly resume working where you stopped.
- Use the Build Agent chat panel for iterative development.
- Validate compliance using Vault Console to check audit trails and security settings before deployment.
- Develop in a sandbox using Developer Sandboxes for isolation and safety, or use a Personal Development Instance (PDI) or a non-production instance to avoid production deployment.

Onboarding for agentic development

The following checklist helps to ensure your environment is properly configured, that you understand the core tools and general guidelines, and that you adopt secure, compliant workflows. Whether you're new to ServiceNow or transitioning to agentic development, this resource will set you up for success.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Agentic development access and setup

1. Ensure you have credentials for the development instance.
2. Confirm that you have the appropriate license for the tools you want to use, for example for Build Agent or ServiceNow Otto for Creator.

   Note: Depending on your license, you will have access to certain application features, generative AI skills, agentic workflows, and AI agents. For more information, see ServiceNow product tiers.

3. Confirm that the plugins for the tools that you want to use are installed and activated. For more information, see the following topics:
   - Install Build Agent

     Note: You can also connect a supported MCP server to Build Agent, for example, to accelerate the conversion of Figma designs into enterprise-grade applications. For more information, see MCP connections and Build Agent.

   - Install ServiceNow Otto for Creator
4. Install Developer Sandboxes, which is not required for agentic development but provides a helpful instance to experiment with development. For more information, see Developer Sandboxes.

ServiceNow Studio and ServiceNow IDE are automatically available on every instance, so you don't need to confirm their installation.

Prepare your development environment

1. Use Developer Sandboxes to allocate a sandbox for safe, isolated agentic development. For more information, see Developer Sandboxes.
2. Validate compliance policies and security controls.
3. Enable Vault Console and Machine Identity Console for sensitive data protection and API governance.

Learn core features

1. Practice converting natural language prompts into apps using Build Agent.
2. If you're using ServiceNow Otto for Creator tools, familiarize yourself with the following:
   - Text-to-form/App scaffolding
   - Code explanation and refactoring, though Build Agent has a more robust implementation.
3. Explore multi-turn chat for iterative development through conversational prompting.

Visit the App Engine Academy playlist on YouTube for recordings of training sessions.

Get familiar with general guidelines

See General guidelines for agentic development for more information.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

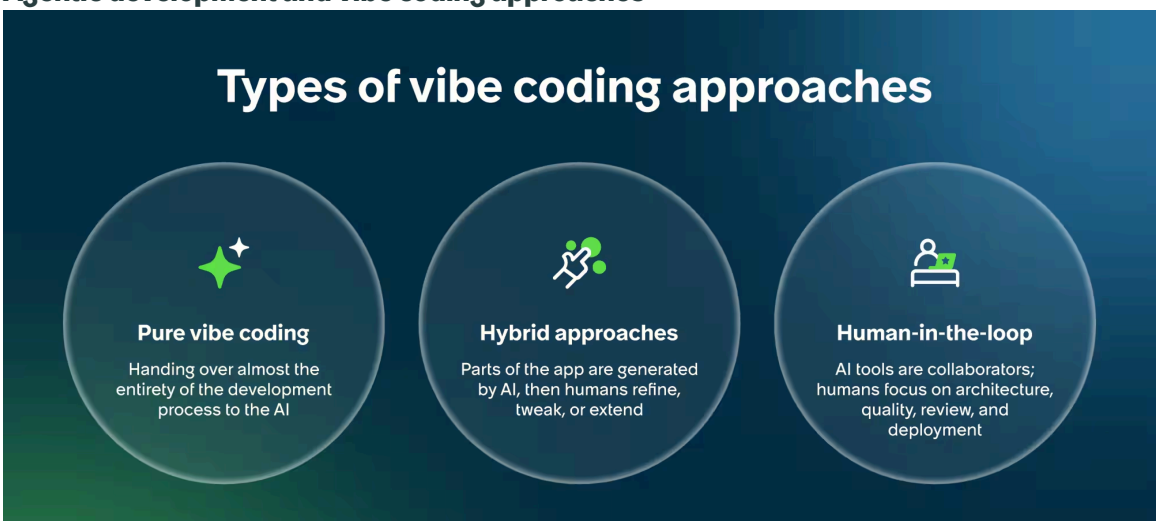
Engage with the development community

Check out ServiceNow Community to connect with other developers and share ideas.

Agentic development on the ServiceNow AI Platform

Use ServiceNow AI Platform environments, products, tools, and skills to develop, expand, test, and deploy apps using a conversational interface. Autonomous app development on the ServiceNow AI Platform is agentic development.

Agentic development and vibe coding approaches



Agentic development in ServiceNow AI Platform environments

You can develop apps agentically with AI tools on the ServiceNow AI Platform in a safe, isolated environment using a non-production instance with Developer Sandboxes, or on a Personal Developer Instance (PDI).

Note: You can develop for ServiceNow locally, outside of an instance using local VS Code and the ServiceNow SDK. However, this topic discusses agentic development on the ServiceNow AI Platform on an instance.

Developer Sandboxes and development

Developer Sandboxes are isolated environments within a non-production instance that enable developers to build and test changes without impacting other work. They solve the problem of shared-instance conflicts by providing the following capabilities:

Parallel development

Each sandbox is independent, so multiple developers can work concurrently.

Full metadata copy

Developer Sandboxes are initialized with the base instance's metadata.

Version control alignment

Supports Git-based workflows for compliance and architectural standards. Update sets are also available.

On-demand provisioning

Developer Sandboxes can be requested and spun up dynamically.

Use Developer Sandboxes to eliminate change conflicts, enable branching and merging strategies, and reduce risk in enterprise-scale development.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Agentic development works with Developer Sandboxes in the following ways:

- Agentic development accelerates app creation, while Developer Sandboxes provide a safe space to experiment with AI-generated changes.
- Every agentic change in a sandbox can be tracked, previewed, and rolled back to maintain governance and compliance.

For more information on Developer Sandboxes, see Developer Sandboxes.

PDIs and agentic development

A Personal Developer Instance (PDI) is a free, standalone ServiceNow instance provided for personal use and experimentation. It is not connected to a customer's production environment and is controlled by ServiceNow. PDIs are empty instances where developers set up apps and data from scratch.

- The ServiceNow AI Platform offers Build Agent trials in PDIs with a 10-prompt limit per month for personal experimentation.
- Try out Build Agent on a PDI to experience agentic development without impacting production environments.
- PDIs support learning prompting patterns, testing AI-generated apps, and refining workflows before moving to Developer Sandboxes or sub-production instances.
- For production-grade work, use Developer Sandboxes instead, which are exact copies of customer instances and support source control integration.

For more information on PDIs, see Personal developer instance guide.

Agentic ServiceNow AI Platform development with Build Agent

Build Agent is the primary AI-powered development tool that works directly with the ServiceNow AI Platform to help you build applications. It generates code based on natural language descriptions, suggests improvements, and automates repetitive development tasks while you retain control of your application.

Build Agent is not a one-shot tool, but enables enterprise-grade agentic development.

- Use Build Agent if you understand and can support what it generates.
- Build Agent is for development on the ServiceNow AI Platform, and requires some knowledge of the ServiceNow AI Platform.
- Build Agent requires you to review and actively participate in the development process.
- Build Agent requires the admin role.

Note: While Build Agent is part of ServiceNow Otto for Creator, it's the primary ServiceNow tool for agentic development, and thus highlighted in its own section of this document.

Key benefits of Build Agent

- Built-in governance to develop securely
- Faster enhancement of existing applications, such as adding new features
- Reduced development backlogs and streamlined deployment of new applications
- Better documentation of existing applications through automated documentation
- Increased developer productivity across skill levels

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Reduction in overall development cost
- Automated testing with Automated Test Framework (ATF) test suite generation and execution

Additional use cases

Some sample use cases for app development with Build Agent include the following examples:

- Rapid prototyping of a new business application
- Creating an HR onboarding workflow
- Adding laptop request to Service Catalog
- Hardening AI-generated code for production
- Refactoring legacy business rules
- Building a mobile-first experience
- Training new developers on the ServiceNow AI Platform
- Extending existing applications
- Designing agentic workflows that chain actions across ServiceNow tables, approvals, and integrations
- Creating custom AI agents tailored to your organization's specific workflows, such as service desk, HR, or IT operations
- Generating skills that extend ServiceNow Otto with domain-specific capabilities

Accessing Build Agent

You can access Build Agent in both ServiceNow Studio and the ServiceNow IDE. For large or complex implementations, you can use Autonomous Engineer to go from requirements to a working implementation without building each artifact manually. For more information, see Autonomous Engineer in Build Agent.

Note: This document provides an overview of how Build Agent sits in the agentic development workflow. For full product documentation, see Build Agent and ServiceNow AI Platform tools.

Get started with agentic development using Build Agent

Use Build Agent on the ServiceNow AI Platform to build apps agentically by executing complex configuration and development tasks through conversational prompts. This approach simplifies editing and creating ServiceNow applications and metadata such as tables, relationships, and access controls, without manual navigation.

Note: Review any AI-generated code before deploying it.

For more information on Build Agent, see Build Agent.

Build Agent and agentic app development

Build Agent is an autonomous AI agent native to ServiceNow that translates plain language instructions into ready-to-deploy applications and metadata, using the platform's domain language and guardrails. It's purpose-built for full-stack creation and editing across tables, flows, UI, and scripts, and it operates as the core engine for agentic development on the ServiceNow AI Platform.

Available in the ServiceNow IDE and ServiceNow Studio, Build Agent appears as a chat panel with a multi-turn conversation interface. Build Agent processes natural language prompts,

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

autonomously generates full-stack applications, manages the entire build process, and responds to feedback. The agent generates all necessary code, organizes files clearly, and manages both the core logic and user interface components of applications.

Build Agent supports the following development tasks:

- Generate and edit enterprise-grade applications: Create full-stack ServiceNow applications from natural language prompts with enterprise-grade governance.
- Accelerate development workflows: Support flow design, catalog item creation, and Automated Test Framework (ATF) test creation.
- Enhance understanding: Explain how to best perform tasks on the ServiceNow AI Platform.
- Perform diverse code tasks: Refactor tables, explain and document code, validate and enhance existing applications, fix errors, run queries, and more.
- Support governance: Work inside platform scopes, roles, and testing workflows rather than as a detached external bot.
- Enable developer learning: Answer ServiceNow development questions, summarize documents, and provide practical examples.

For more information on Build Agent, see Build Agent.

General guidelines for agentic development with Build Agent

To maximize Build Agent effectiveness, use the following practices:

- Design before coding: Think through and document the requirements for your application across the data and UI layers, for example using Workflow Studio or Figma.
- Start with a clear plan: Collaborate with Build Agent to define scope, required tables, and metadata types.
- Instruct with context: Write instructions for what you want to achieve with your application using Markdown in your file system. You can then ask Build Agent to use the file as context for its work.
- Use specific terminology: Treat Build Agent as your development partner. Provide specific, clear instructions using ServiceNow platform terminology such as table names, field names, roles, and artifact types.
- Test early and often: Add sample records, test on the instance, and run ATF tests throughout development.
- Use version control: Use Git for tracking changes and maintaining a clean workspace structure.
- Provide visual context: Give screenshots to Build Agent to troubleshoot UI issues and request changes.
- Extend with third-party libraries: Integrate third-party Node Package Manager (NPM) libraries, such as React JS, for enhanced interfaces.
- Maintain documentation: Keep clear rules and documentation in project folders; use Build Agent to create knowledge base articles.
- Iterate and experiment: Prototype freely, then ask Build Agent for an ideal prompt to recreate the result once your vision is complete.

Where to use Build Agent

Access Build Agent through either of the following options:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Customer instance: Upgrade to a minimum of Zurich and activate Build Agent (Trial) with a 25 calls per month limit.
- Personal developer instance: Request a Zurich or later Personal Development Instance (PDI) on the Developer site with a 10 calls per month limit.

Tools for using Build Agent

You can access Build Agent within both ServiceNow Studio and ServiceNow IDE. Both products offer different experiences for different levels of agentic app development.

Limitations of Build Agent

Use Build Agent for creating standalone apps that have limited interaction with flows, deployment, etc. on the ServiceNow AI Platform. For details, see Limitations of agentic app generation with Build Agent.

Helpful resources

ServiceNow resources for agentic development:

App Engine Academy: Practical Build Agent - Add Gen AI to your existing dev methodology

App Engine Academy on YouTube

Install a trial version of Build Agent

Get Build Agent on the ServiceNow Store

Community article on learning prompting from Build Agent

The fastest way to learn Build Agent prompting? Ask Build Agent.

Vibe coding with ServiceNow Build Agent: The future of AI-powered app development

Community blog on getting started with Build Agent

Build Agent enablement guide

#BuildWithBuildAgent Guidebook on Community

Build Agent and ServiceNow AI Platform tools

Compare how Build Agent behaves in ServiceNow Studio (UI-first, declarative workflows) versus the ServiceNow IDE (code-first, autonomous full-stack development), so you can choose the right environment for your task and audience.

Build Agent is available in both ServiceNow Studio and the ServiceNow IDE, but each environment emphasizes a different development style. ServiceNow Studio provides a guided, UI-first experience that focuses on metadata creation and controlled, iterative changes. The ServiceNow IDE provides a code-first experience with an autonomous agent capable of generating and modifying full-stack applications through conversational prompts.

Audience and intent

Choose the environment based on your skill set and the type of work:

- ServiceNow Studio: Low-code builders and admins who prefer declarative, metadata-driven workflows with previews, diffs, and guardrails.
- ServiceNow IDE: Pro-code developers who need conversational, code-centric generation, advanced customization, and end-to-end build and deploy steps.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Key differences between ServiceNow Studio and the ServiceNow IDE

Area	ServiceNow Studio	ServiceNow IDE
Primary style	UI-first, declarative, metadata-centric	Code-first, conversational, full-stack
Typical users	Low-code builders, admins	Pro-code developers
Interaction model	Guided steps with suggestions, diffs, and summaries; selectable modes (guided, batch, one-shot)	Chat-driven autonomous generation; user approves edits, then build and deploy
Scope of automation	Create or update platform metadata (tables, flows, experiences) with dependency awareness	Generate and edit entire scoped or global apps (UI and backend), explain or repair code, run queries, create documentation
Change control	Strong guardrails; preview via ServiceNow Studio diff surfaces	Approval gates before writing; build and deploy workflow in the ServiceNow IDE
Best fit	Iterative configuration, edits, low-code delivery	Greenfield app creation, deep refactors, debugging, multi-artifact edits
Dependencies	Uses the ServiceNow Studio agentic experience layer and metadata explorers	Relies on the ServiceNow IDE workspace, file and metadata explorers, and build pipeline

Note: You can do conversational checkpoints with Build Agent and roll back to the last conversation checkpoint in both ServiceNow Studio and the ServiceNow IDE.

Typical Build Agent workflow

A general workflow for using Build Agent in either ServiceNow Studio or the ServiceNow IDE is the following:

1. Make sure that everything you need is properly configured in the settings, such as supported MCP server connections. For more information, see Build Agent configuration.
2. Open ServiceNow Studio. Use the central chat area on the home page to start a new conversation, or select the Conversations icon in the Navigator panel to open an existing conversation.
3. Describe what to create or change in natural language.
4. Let Build Agent parse requirements and propose the application and files to create or modify.
5. Build Agent edits code or metadata or scaffolds a new application.
6. Review proposed edits, diffs, and summaries, and approve or adjust before applying changes. Review checkpoints and manual edit update sets. In ServiceNow Studio, view generated app details from the **Apps** tab, and inspect the source code from the **Explorer** tab.
7. Iterate until the desired metadata changes are complete. For more information, see Supported metadata in Build Agent.
8. Prompt Build Agent to create and run Automated Test Framework (ATF) tests to verify that the tests execute as expected. Depending on your configuration, Build Agent may ask you if you

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

want to run ATF tests. If there are failures, auto troubleshooting triages the tests and produces a regression test suite that you can use to monitor app health.

9. Instruct Build Agent to build the application; verify results in the File Navigator or Metadata Explorer.
10. Deploy the application. If you're using source control, you can push to Git. For more information, see Deploying what you built with Build Agent.

For more information, see the following topics:

- Get started with agentic development using Build Agent
- Agentic development app refinement in ServiceNow Studio
- Agentic development app refinement in the ServiceNow IDE

How to choose

Use ServiceNow Studio when you want to do metadata-centric, abstracted low-code development. ServiceNow Studio provides structured, metadata-focused changes with strong previews and guardrails in low-code builders.

Use the ServiceNow IDE when you want file system code-centric development. The ServiceNow IDE provides autonomous, end-to-end generation, complex refactors, and deeper debugging and code explanations.

Notes and limitations

Keep the following in mind when using Build Agent:

- Build Agent generates metadata supported by ServiceNow Fluent. Verify artifact compatibility before approval.
- Feature availability and UI details might differ between monthly releases. Confirm behavior against your instance version.

Autonomous Engineer in Build Agent

Autonomous Engineer is an agentic worker in Build Agent that takes your requirements and generates an implementation plan. It then builds all work items in parallel, without requiring you to construct each artifact manually.

When an application is large or complex, building it artifact-by-artifact through conversational prompts can be slow. Autonomous Engineer addresses this by taking your requirements as input and driving the full implementation lifecycle end to end, from requirements intake through parallel execution of work items.

Note: You must install the Autonomous Engineer app from the ServiceNow Store. Check your entitlements to see whether you can use Autonomous Engineer.

Autonomous Engineer is available in ServiceNow Studio and covers custom applications. For full product documentation, see Autonomous Engineer in Build Agent.

Autonomous Engineer process

Autonomous Engineer follows two sequential phases. You retain control throughout both phases and must review and approve the plan before any artifacts are built.

Planning phase

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Provide your requirements as a natural language description or a file upload, such as a CSV of user stories. Autonomous Engineer interviews you to resolve ambiguous details. It then queries your instance to identify existing tables, roles, fields, and catalog items so it does not duplicate what is already there. The output is a plan with work items structured in an Agile user story format, including acceptance criteria and test criteria, sequenced using a dependency graph. Review and edit each work item before approving the plan.

Execution phase

After you approve the plan, Autonomous Engineer generates a background agent for each work item and builds all work items in parallel. Each agent builds the required artifacts, generates Automated Test Framework (ATF) tests, and runs those tests. Failed tests trigger automatic root cause analysis and retry cycles. When all work items are complete, Autonomous Engineer generates an update set for deployment.

Agent packs

Autonomous Engineer uses agent packs to give background agents product-specific domain knowledge during implementation. For more information, see Agent packs for Autonomous Engineer.

Autonomous Engineer compared to standard Build Agent development

Standard Build Agent development is conversational and turn-by-turn: you prompt, review, and iterate on one artifact or change at a time. Autonomous Engineer shifts that model by working from a plan. It builds multiple work items in parallel in the background, so you interact primarily at the planning and review stages rather than guiding every step.

Use standard Build Agent development when you're building incrementally, exploring ideas, or making targeted changes. Use Autonomous Engineer when you have well-defined requirements for a larger implementation and need to reduce the number of manual prompting steps.

Creating custom agentic workflows, in-app agents, and skills with Build Agent

Use Build Agent to create and deploy custom agentic workflows, AI agents, and skills that add agentic experiences to your ServiceNow custom applications, supported by the built-in governance framework.

The agentic development capabilities in Build Agent extend beyond building applications. You can use the same intent-driven approach to create custom AI agents, design agentic workflows that orchestrate multi-step automated processes, and build skills directly within your ServiceNow environment. You don't need to specify whether you want to build an agent, a workflow, or a skill. Instead, describe the business problem or outcome you want to achieve, and Build Agent determines the appropriate components to generate. Depending on your requirements, it may create an agentic workflow, one or more agents, a set of skills, or a combination of all three.

Before generating any new components, Build Agent scans your instance for existing agents, skills, and workflows that may already satisfy your requirements. If there are matches, Build Agent proposes using the existing components or creating new ones, so you can reuse what's already available before adding to your instance. In each case, Build Agent uses your instance's existing tables, roles, permissions, and automation patterns as context.

All you need to do is describe the business problem you want to solve, and Build Agent determines whether to build an agentic workflow, an agent, or set of skills, and identifies the tools required. Drawing on its understanding of your application context, Build Agent handles the complexity of agent creation for you, with no prior knowledge required and significantly less development time.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

As part of the agent and skill creation process, Build Agent guides you through defining access controls and permissions alongside the agent itself, keeping governance integrated from the start. Once created, you can test and activate the agent in AI Agent Studio, then monitor and govern it in AI Control Tower.

Note: Your entitlements determine whether you have access to agentic workflows, agents, and skills in Build Agent.

Custom in-app agents

Build Agent reads your application's files to understand the context, then plans and creates agents and skills tailored to your app's specific use case. For example, Build Agent can identify manual tasks within your application that are candidates for automation, then plan and build a custom agent to handle them autonomously, no code required. Once created, the agent is available in AI Agent Studio for testing, trigger setup, and further configuration.

Agentic workflows

Build Agent can generate agentic workflows that coordinate sequences of automated actions across your ServiceNow instance. Rather than scripting individual steps, you describe the end-to-end process you want to automate, and Build Agent produces a flow that connects agents, triggers, and actions in a coherent sequence. This context-awareness sets the ServiceNow approach apart from generic AI development tools: Build Agent understands constructs specific to ServiceNow to produce flows that are meaningfully integrated into the ServiceNow AI Platform.

Skills

Build Agent can generate custom skills scoped to your application's data model, without manually authoring skill definitions. You describe what you want the skill to do, and Build Agent creates skills that align with your existing table structure, field configurations, and access controls.

You can create custom skills individually, that is, they don't need to be used as tools for an agent. Think of skills as custom instructions that extend the quality of your agentic development outcomes. For more information, see Exploring AI Skill Kit.

Helpful resources

Some ServiceNow resources that can provide you with helpful information are:

Create agentic workflows, agents, and skills

Details on how to create custom agentic workflows, in-app agents, and skills with Build Agent.

Pro Developer Academy: Vibe Coding AI Agents

Pro Dev Academy session recording on building in-app agents agentically.

Limitations of agentic app generation with Build Agent

Build Agent accelerates agentic app development, but it also has some limitations that you should understand before using it. These constraints span feature coverage, platform compatibility, and governance requirements, all of which affect how and when you can use the tool effectively in agentic development workflows.

General limitations of Build Agent

Be aware of the following limitations when using Build Agent:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- AI-generated code requires human review for ACL correctness, security, platform conventions, and governance compliance. Put peer-review gates, static checks, and validation steps in place.
- You still must work with builders in ServiceNow Studio for app files and metadata types not yet supported in Build Agent, such as certain tables, UIs, or advanced features.

  Note: Build Agent can work on existing applications. You can use Build Agent to enhance base workflows, such as creating business rules on existing tables.

- Security in apps developed with Build Agent is at the application and API level, not at granular record and field level by default. However, you can use agentic development to build security onto the app by making requests. For examples, see Example prompts.
- Limited support for cross-product AI integration.

Build Agent can generate data models and business rules. You should understand where data from your AI-generated apps is stored on the ServiceNow AI Platform.

Build Agent and ServiceNow Fluent

To understand the upper limit of what Build Agent can do, review the ServiceNow Fluent documentation. If you're not familiar with the ServiceNow AI Platform, the ServiceNow Fluent documentation can help you determine what Build Agent can do. For more information, see the following topics:

- ServiceNow Fluent API reference
- ServiceNow Fluent

If ServiceNow Fluent does not support a metadata type, Build Agent cannot update it unless you provide precise platform-specific instructions.

Agentic development app refinement in ServiceNow Studio

ServiceNow Studio connects AI-generated apps to enterprise-grade solutions, where you can review, edit, and enhance your application in a dedicated development environment.

After developing an application agentically with Build Agent or ServiceNow Otto for Creator, ServiceNow Studio provides a powerful, developer-friendly environment for reviewing, customizing, and perfecting your app.

If you're a developer who's new to agentic development, ServiceNow Studio provides a more abstracted approach to code refinement compared to ServiceNow IDE.

For documentation on using Build Agent in ServiceNow Studio, see Build Agent in ServiceNow Studio.

ServiceNow Studio workflow

To review and refine your app from Build Agent in ServiceNow Studio, follow these steps:

1. Open your app in ServiceNow Studio.
2. Use Build Agent to create or edit metadata for your application.
3. Review and edit application files.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Use the **Application Files** tab to see all generated elements and metadata—tables, access controls, roles, record producers, and more.
- Open any file (such as a table or ACL) in a new tab for detailed inspection and editing.
- Confirm that generated tables capture the correct data and adjust field properties as needed.
- Edit access control lists (ACLs) to confirm that the correct roles and permissions are applied.
- Review and modify roles, record producers, and other artifacts for accuracy.

4. Use developer tools in *`ServiceNow Studio`*.
   - ServiceNow Studio provides a tabbed, IDE-like interface for navigation and editing.
   - You can refine code, add business logic, and manage metadata across your app's scope.
   - Use Build Agent for update set management, cross-scope editing, and safe deployment.
5. Iterate with Build Agent.
   - Use the Build Agent panel to edit metadata for your application.
   - Toggle the panel as needed to avoid covering important information.
6. Test and validate the app.
   - Use Build Agent to create and run Automated Test Framework (ATF) tests.
   - Verify that your refined app functions correctly.

When to choose the ServiceNow IDE or ServiceNow Studio

1. ServiceNow IDE: Best for file-centric, code-heavy edits, rapid iteration, and pairing with the ServiceNow SDK for local dev/CI. Suitable for developers familiar with the ServiceNow AI Platform.
2. ServiceNow Studio: Best for metadata-centric review (tables, ACLs, roles) and cross-scope editing with a visual overview of app artifacts; keeps Now Assist in-context for conversational adjustments. Suitable for developers with various levels of experience on the ServiceNow AI Platform.

Agentic development app refinement in the ServiceNow IDE

The ServiceNow IDE provides a VS Code, file-based editing experience with syntax highlighting, code completion, and real-time installing to your instance. Use this approach to harden generated, scaffolded apps into production-grade solutions.

If you prefer to do local development with the ServiceNow SDK and a command line interface (CLI), the ServiceNow IDE also works with the ServiceNow SDK via Git for collaborative workflows, for example for offline iteration.

If you're an experienced ServiceNow developer who's familiar with agentic development, ServiceNow IDE is a suitable starting point for refining your app development.

For full documentation on using the ServiceNow IDE, see ServiceNow IDE.

Agentic development workflow in the ServiceNow IDE

To refine your app in the ServiceNow IDE, follow these steps:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

1. Create a workspace. Use Build Agent to create or edit application metadata. For more information, see ServiceNow Otto for Creator.
2. Create an app.
   - Use Build Agent to create the app.
   - You can choose to edit an existing app.
3. Inspect the code in the File Explorer.
   - Review business rules, script includes, and flow logic.
   - Apply general guidelines for performance and security.
4. Inspect the metadata objects and adjust the data model.
   - Launch the ServiceNow IDE and load the app created by ServiceNow Otto or Build Agent.
   - The ServiceNow IDE Metadata Explorer displays all generated metadata, including tables, scripts, forms, flows, UI components, and ACLs.
   - Verify field types, choice lists, default values, and relationships.
   - Confirm indexes and unique keys are set appropriately for your use case.
5. Refine client-side behavior and user experience.
   - Review client scripts, UI policies, and page layouts.
   - Adjust data bindings, events, and conditional rendering for clarity and accessibility.
   - If you need to make changes, use Build Agent.
6. Validate security and roles.
   - Review ACLs to confirm that the correct roles and conditions are applied.
   - Tighten permissions and remove any overly permissive logic introduced during scaffolding.
7. Use Build Agent to generate and run Automated Test Framework. For more information, see Test what you built with Build Agent.
8. Move to local development with the ServiceNow SDK for deeper refactoring.
   - Use the ServiceNow SDK command line interface (CLI) for build, deploy, and fetch operations, version control, and CI integration.
   - Use the Source Control extension within the ServiceNow IDE to move changes between local versions and the ServiceNow IDE. For more information, see Using source control in the ServiceNow IDE.

General guidelines for the ServiceNow IDE

1. Structure first, micro-optimize second: Lock down tables, relationships, and ACLs before polishing scripts.
2. Use AI where it helps: Keep the Build Agent panel handy to draft snippets or explain legacy code, then review and harden the output manually in the ServiceNow IDE.
3. Prefer declarative where possible: Reduce custom code by using platform configuration (UI Policies, Flow Designer, conditions) for maintainability.
4. Automate validation early: Use Build Agent to create ATF tests for critical paths (create/read/update/delete, role-based access, approvals) before making broad changes.
5. Version and promote safely: When you transition to packaging and deployment, align IDE work with update-set or pipeline practices so your changes can be promoted cleanly across environments.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ServiceNow Fluent and agentic development

ServiceNow Fluent is a development framework and SDK for ServiceNow that connects traditional coding with agentic workflow automation.

With ServiceNow Fluent, you can:

- Build complex workflows and applications using familiar code editors (like VS Code or the ServiceNow IDE)
- Get real-time feedback, including TypeScript type checking and code validation
- Use descriptive, declarative language to define ServiceNow application metadata
- Seamlessly connect to ServiceNow instances for instant deployment and testing
- Use your Git provider of choice as the authoritative source of truth for your code

ServiceNow Fluent works with agentic development and Build Agent. You can generate an app or flow using conversational prompts, then refine, extend, and harden it using the ServiceNow Fluent code-centric tools. Both pro-code and low-code developers can collaborate and iterate, using the strengths of agentic and modern development.

Important: Build Agent and Autonomous Engineer only create metadata supported by ServiceNow® Fluent. For more information, see ServiceNow Fluent. For the latest API reference, see https://servicenow.github.io/sdk/

When to choose the ServiceNow IDE or ServiceNow Studio

1. ServiceNow IDE: Best for file-centric, code-heavy edits, rapid iteration, and pairing with the ServiceNow SDK for local dev/CI. Suitable for developers familiar with the ServiceNow AI Platform.
2. ServiceNow Studio: Best for metadata-centric review (tables, ACLs, roles) and cross-scope editing with a visual overview of app artifacts; keeps Now Assist in-context for conversational adjustments. Suitable for developers with various levels of experience on the ServiceNow AI Platform.

Agentic app creation with ServiceNow Otto for Creator

ServiceNow Otto for Creator provides core skills for rapid conversational app development. Use the integrated tools to generate scripts, optimize code, scaffold applications, and automate testing through natural language prompts.

Note: While ServiceNow Otto for Creator is not a vibe coding product suite, you can use its AI capabilities to quickly enhance app development.

ServiceNow Otto for Creator is the ServiceNow collection of generative AI skills for building apps on the ServiceNow AI Platform. Use it to generate apps, flows, code, tests, UIs, analytics, catalog items, RPA bots, and more from natural language prompts directly in offerings like ServiceNow Studio.

Some skills in ServiceNow Otto for Creator have a conversational interface and can be used for vibe coding and agentic development. For example, the catalog item generation skill in Catalog Builder. For more information, see AI Authoring for Catalog Builder.

Core capabilities

For complete information on using ServiceNow Otto for Creator, see ServiceNow Otto for Creator.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The following metadata-driven tools are available for using AI to develop apps:

- Conversational development: Use Build Agent, which is part of ServiceNow Otto for Creator, to create apps and metadata through a conversational interface.
- Application generation and scaffolding: Create tables, experiences, roles, and initial data model from a prompt.
- Flow generation, recommendations, and summarization: Generate multi-step flows, get next step suggestions, and auto-summarize what a flow does.
- Code generation: Prompt-based server-side and client-side scripts with AI suggestions for script includes, business rules, and GlideRecord queries.
- Catalog item generation: Produce Service Catalog items from text directions.
- UI generation: Build experiences by describing the desired interface.
- Data visualization generation (Platform Analytics): Conversational chart and indicator creation.
- Playbook generation: Outline automated processes from natural language.
- RPA bot generation: Create desktop automations from text prompts inside RPA Desktop Design Studio.
- App, table, and flow summaries: Generate explainers and summaries to support onboarding and application understanding.

Where to start

1. Welcome guide and skills and prompting: See the Community Welcome Guide and Skill Prompt Guides for Creator. For more information, see the following resources:
   - Community Welcome Guide on Community
   - Skill Prompt Guides on Community
2. Store app and release notes: Install/upgrade ServiceNow Otto for Creator skills from the ServiceNow Store.
3. Onboarding topic in this document: See Onboarding for agentic development.

ServiceNow Otto for Creator and agentic development on the ServiceNow AI Platform

ServiceNow Otto for Creator is the ServiceNow AI Platform toolkit for using AI to generate and refine apps. Instead of switching between generic IDEs and separate agents, ServiceNow Otto for Creator provides embedded skills aligned to platform constructs (such as tables, ACLs, flows, and catalog items). You can describe a solution through conversation, then inspect and test it with platform tools (such as ATF, ACLs, and the Metadata Explorer).

General ServiceNow Otto for Creator workflow

1. Enable the skills: Configure the triggers, settings, and display locations for ServiceNow Otto for Creator skills to enable generative AI capabilities across the ServiceNow AI Platform. For more information, see Activate an AI skill.
2. Describe the app: For example, you can request to “Create a cross-training app with instructor signup, session scheduling, approvals.” Then the application generation scaffolds tables, roles, and experiences.
3. Add process logic: Use flow generation for approvals and escalations and Flow recommendations to suggest next steps.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

4. Add custom code logic: Prompt code generation for business rule and script include logic (such as GlideRecord queries), and then review and refactor as needed.
5. Expose services: Catalog item generation creates request items for sessions/resources.
6. Automate outside the platform: RPA bot generation to fetch instructors' availability from a desktop calendar or copy rosters.
7. Explain and test: Use app, flow, and table summarization to document; generate tests with the test-generation skill; run ATF.

Agentic development with ServiceNow Otto for App Engine

ServiceNow Otto for App Engine enables you to embed AI functionality into an application that you built on the ServiceNow AI Platform.

While ServiceNow Otto for Creator skills and Build Agent help you build the app, ServiceNow Otto for App Engine helps you embed AI features into the experience so requesters and fulfillers have access to AI-powered interactions.

ServiceNow Otto for App Engine enables you to leverage relevant ServiceNow AI Platform capabilities to enhance custom apps with GenAI and AI agents. Some of its most important features are the following:

- AI Skill Kit, which you can use to create custom skills for ServiceNow products and ServiceNow Otto for App Engine. For more information, see AI Skill Kit.
- AI Agent Studio, which enables you to create, manage, and test AI agents and agentic workflows. For more information, see AI Agent Studio overview.
- The custom app record summarization skill, which generates AI summaries for the contents of records in custom applications and tables. For more information, see Custom app record summarization skill.

Common capabilities for ServiceNow Otto for App Engine include:

- AI Search and knowledge summarization inside the app experience.
- AI Agent Studio and agentic workflows that guide users and fulfillers through tasks.
- Generative flow actions and in-app assistance that reduces manual effort.
- Content-aware responses that adhere to platform security and data access models.

For more information on ServiceNow Otto for App Engine, see ServiceNow Otto for App Engine.

With ServiceNow Otto for App Engine, you have access to ServiceNow Otto AI assets, such as Platform skills, AI agents, and agentic workflows, as well as the tools for developing custom AI assets using AI Skill Kit and AI Agent Studio.

For a list of all the tools available in ServiceNow Otto for App Engine, see Tools included with ServiceNow Otto for App Engine.

Testing agentically developed apps

As part of the agentic development process on the ServiceNow AI Platform, one of the key features is the automatic generation of Automated Test Framework (ATF) tests.

When you use agentic development, the ServiceNow AI Platform can:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Scaffold the data model, business logic, and user interface based on your prompts.
- Generate ready-to-use ATF tests that validate the functionality of your new applications or flows.
- Incorporate quality assurance and code resilience from the start, reducing the need to address them after initial development.

This means you can go from idea to a working, testable app much faster. After the initial generation, you can refine the app and its tests in ServiceNow Studio or the ServiceNow IDE, and run the ATF tests to verify that the app functions correctly.

Note: Build Agent can create, edit, and run ATF tests. For more information, see Test what you built with Build Agent.

ATF has a troubleshooting agent to debug test failures on specific metadata. For details, see ATF troubleshooting agent.

For full documentation on Automated Test Framework, see Automated Test Framework (ATF).

Agentic development and deployment

After you generate and expand an app on the ServiceNow AI Platform, the deployment process is tightly integrated into the workflow.

Workflow for deploying an agentically developed app

After development, review, and testing are complete, a typical deployment workflow includes the following steps:

1. Collaborative design: Business owners and IT collaborate on requirements and ideas using their preferred tools.
2. AI-driven app development: Build Agent, Autonomous Engineer and ServiceNow Otto process files, chat history, and diagrams to generate and implement app updates.
3. Review and testing: Teams preview updates, make revisions, and run rounds of performance and readiness testing.
4. Developer review: A developer reviews the AI-generated changes, compares versions, and confirms the changes are ready for deployment.
5. Deployment approval: The project is handed off to a deployment manager, who initiates the deployment approval process.
6. Autonomous checks: AI agents automatically scan for issues (such as sensitive data exposure or model integrity problems) and remediate them before deployment.
7. Final deployment: After all readiness scans and approvals, the new app is deployed securely and efficiently.

Deployment options

Build Agent and Autonomous Engineer support the following deployment methods for apps created and edited with agentic development:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Git-based source control integration: ServiceNow supports Git-based workflows for version control and CI/CD.
  - You can push scoped apps to Git repositories, enabling branching, merging, and automated deployments. ServiceNow supports bring-your-own Git integration, such as GitHub or Bitbucket.
  - You must be on Brazil Patch 1 to use source control in ServiceNow Studio.
  - For more information, see Integrating source control with the ServiceNow IDE and Fluent source control in ServiceNow Studio.
- Update sets and application packaging: Standard ServiceNow deployment uses System Update Sets to track changes.
  - Advanced guidance includes packing update sets into scoped applications for easier transport and installation across instances, for example using Application Repository (AppRepo).
  - For information on update sets for Build Agent, see Update sets and Build Agent.
  - For information on update sets for Autonomous Engineer, see Update sets and Autonomous Engineer.
  - For more information on System Update Sets, see System update sets.

Workflows for moving apps through higher instances

After you create an app using Build Agent or Autonomous Engineer, you have several options to move the app to the test instance.

1. Wrap the entire scoped application in an update set. The workflow is as follows:
   a. Go to the **Custom Applications** list, select an app and swap to its scope.
   b. Convert the app to AppRepo.
   c. Publish the update set with demo data.
   d. Put the update set in a deployment request for ReleaseOps, or follow your standard update set process for deployment.
2. Publish the app to AppRepo:
   - You can use a Git-based process or update sets to publish to AppRepo.
   - Scoped apps and apps that are ready for testing can be published to the AppRepo for distribution across environments.
   - After an app is in AppRepo, you can move it through a ReleaseOps pipeline. If ATF tests are included in the pipeline, they automatically run.
   - Register and entitle apps before publishing.
   - For more information on Application Repository, see ServiceNow application repository.

Additional deployment tools and capabilities

The ServiceNow AI Platform has additional deployment tools that include the following tools:

- App Engine Management Center (AEMC):
  - After developing an app, submit it to AEMC for governance checks.
  - AEMC validates ACLs, roles, and compliance settings before deployment.
  - Use ReleaseOps pipelines to move apps through environments with ATF tests and approval gates.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- AEMC provides dashboards for monitoring deployments and managing app versions throughout the lifecycle.
- For more information on AEMC, see Using the App Engine Management Center.

- ReleaseOps:
  - Move changes from development to production through multiple instances using customizable playbooks.
  - Automate preview, commit, and validation of update sets before deployment.
  - Run Automated Test Framework (ATF) tests as part of the pipeline to validate quality.
  - Deploy changes immediately or schedule releases for controlled rollouts.
  - Enforce checks, scans, and approvals before production deployment.
  - For more information on ReleaseOps, see ReleaseOps.
- ServiceNow SDK:
  - Use the ServiceNow SDK to move applications to and from your instance to your local machine. You can integrate the ServiceNow SDK with your off-instance CI/CD process if you have one.
  - Install the ServiceNow SDK locally and use the command line interface (CLI).
  - Authenticate to a ServiceNow instance from the ServiceNow SDK.
  - Push to or install an application on the authenticated instance from your local environment.
- Automated Test Framework (ATF)
  - Tests can be generated by Build Agent and executed in ServiceNow Studio or ServiceNow IDE to confirm functionality after changes.
  - For more information on ATF, see Automated Test Framework (ATF).

Agentic development reference

Find reference material for agentic development on the ServiceNow AI Platform, including a list of tools and prompt examples.

ServiceNow Otto skills: Developer-oriented list

The ServiceNow AI Platform offers a range of skills that support efficiency in each development workflow.

If you're an experienced developer, this list can help you identify which tasks AI automates on your behalf. Each skill represents a previously manual task, a complex business rule, or a classification system that AI automates. Architect the data flow and governance for these skills, rather than writing the repetitive logic they replace.

See Generative AI skills for a complete list of ServiceNow Otto skills.

Application and artifact creation

Application and artifact creation skills

Skill	Description
App generation	Generates a starter ServiceNow application from a natural-language description.
Catalog item generation	Creates a service catalog item from intent.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Application and artifact creation skills (continued)

Skill	Description
Experience generation	Generates UI and experience components from a prompt.
Mobile card generation	Creates mobile UI cards for ServiceNow mobile experiences.
Create a demand	Creates demand records conversationally.
Story generation	Generates stories from higher-level planning items.
Target generation	Creates targets from planning or strategic context.
Write planning item	Generates planning items from text input.

Workflow, flow, and automation

Workflow, flow, and automation skills

Skill	Description
Flow generation	Builds Workflow Studio flows from natural-language requirements.
Flow generation with images	Builds flows using image-based context where supported.
Flow recommendations	Suggests next components while building a flow.
Flow summarization	Summarizes what an existing flow does.
Playbook generation	Creates guided playbooks for structured work.
Playbook generation with images	Creates playbooks enhanced with images.
Playbook recommendations	Recommends playbook steps or structure.
Spoke generation	Generates Integration Hub spokes for integrations.
Event handler generation	Creates event-handling logic from intent.
RPA bot generation	Generates robotic process automation bots.
LEAP installer	Assists with installing and configuring automation components.

Code and script assistance

Code and script assistance skills

Skill	Description
Code Assist generation	Generates code from natural-language prompts.
Code Assist edit	Edits or refactors existing code.
Code Assist autocomplete	Provides inline code completion.
Code Assist summarization	Explains what a piece of code does.
Client script summarization	Summarizes client-side scripts.
Hide Code Assist API definition	Controls visibility of Code Assist APIs.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Code and script assistance skills (continued)

Skill	Description
New column data generation	Generates data for newly added table columns.

Testing, quality, and risk

Testing, quality, and risk skills

Skill	Description
Test generation	Generates test cases from expected behavior.
Change request risk explanation	Explains risk factors in change requests.
Risk assessment summarization	Summarizes risk assessments.
Risk event summarization	Summarizes risk events.
Security incident quality assessment	Evaluates the quality of security incident handling.
Approval recommendation	Recommends approvals based on record context.

Summarization and reading compression

Summarization and reading compression skills

Skill	Description
App summary generation	Summarizes an application's purpose and structure.
Complete record generation	Generates a full human-readable record summary.
Document summarization	Summarizes attached document content.
Docs summarization	Summarizes documentation content.
Doc generation	Generates documentation text.
Work notes analysis	Analyzes work notes and highlights key themes.
Sidebar discussion summarization	Summarizes sidebar discussions.
Chat summarization	Summarizes chat conversations.
Email recommendation	Generates recommended email responses.
Resolution notes generation	Generates resolution notes.
Release notes generation	Generates release notes.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Case, incident, and service operations

Case, incident, and service operations skills

Skill	Description
Incident summarization	Summarizes incidents.
Incident assist	Assists with incident resolution.
Incident sentiment analysis	Analyzes sentiment in incidents.
Case summarization	Summarizes customer or employee cases.
Suggested steps generation	Recommends next actions.
Chat reply recommendation	Suggests chat replies.
Activity response generation	Generates responses to case activity.
Work order task summarization	Summarizes field service work order tasks.
Government case summarization	Summarizes public-sector cases.

Knowledge and content intelligence

Knowledge and content intelligence skills

Skill	Description
KB generation	Generates knowledge base (KB) articles.
Knowledge content recommendation	Recommends relevant knowledge content.
Article optimization	Improves quality and clarity of knowledge articles.
Potential knowledge gaps	Identifies missing knowledge content.

Analytics and data exploration

Analytics and data exploration skills

Skill	Description
Analytics exploration	Explores analytics data via natural language.
Exploration summarization	Summarizes analytics explorations.
Analytics query generation	Generates analytics queries from intent.
Analytics insight generation	Produces insights from analytics datasets.
Analytics follow-up generation	Generates follow-up analytical questions.
Analytics hidden insight generation	Surfaces non-obvious insights.
Dashboard and visualization export	Exports dashboards and visualizations.
Data visualization generation	Generates charts and visualizations.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Configuration, CMDB, and architecture

Configuration, CMDB, and architecture skills

Skill	Description
Configuration item (CI) summarization	Summarizes configuration items.
Manage duplicate CIs	Identifies and manages duplicate CIs.
Service mapping candidates	Suggests service mapping candidates.
Diagram change analysis	Analyzes changes in architecture diagrams.
Business application insights	Generates insights on business applications.
Register a business application	Creates business application records.
Register a digital integration	Registers digital integrations.

Security, risk, and compliance

Security, risk, and compliance skills

Skill	Description
Correlation insights generation	Generates correlation insights for security incidents.
Post-incident analysis	Summarizes incidents after resolution.
Common control objective creation	Creates control objectives.
Control objective impact analyzer	Analyzes impact of control objectives.
Regulatory alert summarization	Summarizes regulatory alerts.
Vulnerable item deduplication	Deduplicates vulnerability items.
Suggest vulnerability solutions	Recommends remediation actions.

Finance, procurement, and supply chain

Finance, procurement, and supply chain skills

Skill	Description
Invoice case summarization	Summarizes invoice-related cases.
Invoice data extraction	Extracts structured data from invoices.
Purchase order summarization	Summarizes purchase orders.
Procurement case summarization	Summarizes procurement cases.
Supplier summarization	Summarizes supplier records.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Finance, procurement, and supply chain skills (continued)

Skill	Description
Negotiation summarization	Summarizes sourcing and negotiation activity.

Example prompts for agentic development

Use these example prompts to build data models, implement security, create business rules, and generate user interfaces when building an app with ServiceNow AI Platform agentic development tools.

To learn more about prompting, see this Community article on The fastest way to learn Build Agent prompting? Ask Build Agent.

Prompting overview and tips

Prompts are counted each time you submit a message to Build Agent. If Build Agent asks a clarifying question and you respond, that response counts as a prompt. Approving a plan that Build Agent presents does not count as a prompt. To get the most value from each prompt, draft your message in a text editor before you submit it.

For more guidelines on prompting, see General guidelines for Build Agent.

Prompt reference

Use the following prompts as starting points for common Build Agent goals. Replace the bracketed placeholders with your specific details.

Prompt reference by goal

Goal	Prompt
Plan	`I want to build [app idea]. Create a detailed plan including tables, fields, roles, and business rules. Wait for my approval before creating any files.`
Prototype	`Build me a [app name] application with [key requirements]. Include basic roles, flows, and a working UI.`
Brainstorm	`I need to build something for [business problem]. What tables, flows, and catalog items would I need?`
Discover	`List the metadata types you can create or edit in this scope.`
Coach	`For [metadata type], coach me on an ideal prompt and common pitfalls.`
Extend	`Open [app name]. Add a field called [field] to the [table] table. Update the ACLs so only [role] can edit it.`
Extend out-of-box	`Add a custom field to the incident table called [field]. Create a business rule that [logic]. Show the plan before applying.`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Prompt reference by goal (continued)

Goal	Prompt
Learn	`Explain [ServiceNow concept] to me. What is it, when do I use it, and show me an example.`
Debug	`I'm seeing this error: [paste exact error]. This happens when: [describe scenario]. Before suggesting fixes: 1) Help me reproduce this consistently. 2) Review the debugger output and logs. 3) Check for access or configuration issues. 4) Examine if this is a known issue.`
Document	`Create a README with setup instructions and a user guide for this application.`
Test	`Create ATF tests to validate [specific functionality] works correctly.`
Security	`Add role-based access so only [role] can edit; others are read-only. Show the plan and ACL changes before applying.`
Diagram	`Generate a Mermaid ER diagram showing all tables, fields, and relationships in this application.`
Flow	`Create a flow that triggers when [condition] and sends an email notification to [recipient] with [details].`
Ideal prompt	`Now that you've built this app, what's the ideal prompt to recreate it from scratch?`

Tip: UI policies handle the common cases declaratively: show, hide, require, or lock fields based on conditions. For example, "when Category is Hardware, show the Asset Tag field and make it required" produces a UI policy that applies the rule without scripting.

Get started with prompting

To get started with Build Agent, try one of the following approaches.

- Prototype a simple app: Describe a straightforward application in one or two sentences. Review the plan that Build Agent presents, then approve it to start the build. This approach lets you learn the workflow with low risk.
- Document an existing app: Open a custom scoped app and ask Build Agent to summarize its architecture, data model, roles, and key components. This produces useful documentation with no risk to the app.
- Extend an existing app or base system table: Add a field, a business rule, or an ACL to an app you're familiar with. Build Agent supports global-scope tables such as Incident, as well as custom and ServiceNow Store apps.

Starter prompts

Use the following prompt patterns to get started with common Build Agent tasks. Replace the bracketed placeholders with your specific details.

Extend safely

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
Open [app name]. Before making changes, document the
current state. Then add [feature]. Return a plan for
approval. After changes, generate ATF tests and a README.
```

Prototype

```
Build me a [app name] application with [requirements].
Include basic roles, a working UI, and flows for
[automation]. Then generate ATF tests and documentation.
```

Extend base system

```
Add a custom field called [field] to the incident table.
Create a business rule that [logic]. Show the plan and
ACL changes before applying.
```

Document

```
Summarize the app architecture, data model, roles, and
key UI components. Don't make any changes.
```

Additional prompt suggestions

After you have the foundation of an app created, save your session and keep prompting to refine and extend your app. For example, you can create Automated Test Framework (ATF) tests, generate release notes, and prepare summaries. Some additional prompt ideas include the following examples:

- Improve generated user interfaces
- Create ATF tests
- Create QA and UAT test plans
- Create a summary of all application functionality
- Add additional industry-specific use cases
- Create an announcement
- Create a user guide
- Generate release notes
- Draft change management plans
- Create training materials for end users
- Build data migration plans
- Generate workflow optimization suggestions
- Prepare executive summary for stakeholders

Example Build Agent prompt to build an app

Prompt: `Create an application that handles several types of issues related to cash management. Create a custom data model with tables for each type of request. Add fields that are typical for these types of issues to each table. Request tables should extend the Task table. Create custom states for each request type to track issue resolution.`

Example Build Agent prompt to add security to an app

Prompt: `Create a new role and define permissions on the request tables for two types of users:`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

1. Requesters who submit issues and can only view their own requests.
2. Fulfillers who can view and edit all requests.

Example Build Agent prompt to add business rules to an app

Prompt: Create business rules for each request that is triggered by state changes in the workflow. Design the business rules to work together to create business processes aligned to industry best practices for each type of issue.

Example Build Agent prompt to generate different UIs for an app

Prompt 1: Create an easy to use UI for users with the cash management fulfiller role to view all issues, update and complete them.

Prompt 2: Create a UI for cash management requesters to submit issues and view or edit issues that they have previously created.

Example prompts for governance

Prompting with governance requirements helps produce secure and compliant applications on the ServiceNow AI Platform. By embedding governance requirements directly into your prompts, you help the AI produce results aligned with your organizational standards for security, compliance, and quality.

For more information on governance, see Build Agent governance.

Tip: If you have explicit security requirements, include them in your initial prompt.

Example: Prompt to include security roles and ACLs

Create an Employee Onboarding app with HR and IT roles, enforce ACLs so only HR can edit personal data.

Example: Specify data sensitivity

Generate a table for medical records with encryption enabled and restrict access to compliance officers.

Example: Request audit and logging

Build a workflow for expense approvals and include audit logging for all status changes.

Example: Mention testing requirements

Create a catalog item for laptop requests and generate ATF tests for submission and approval flows.

Example: Call out compliance standards

Develop a leave management app that meets GDPR requirements and includes consent tracking.

Example: Ask for documentation

Generate a summary of the app and flow logic for governance review.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Run

Deploy, run, and govern custom applications at-scale with unlimited table capacity, embedded AI agents, centralized management for developers and apps, and faster query performance.

App Engine Management Center

The App Engine Management Center (AEMC) is a centralized location for admins and developers to govern and manage the full life cycle of custom applications, from idea submission to deployment and continuous monitoring.

When your applications are ready for release, use ReleaseOps in AEMC to automate deployment across your pipeline. ReleaseOps moves changes between instances, runs scans and tests, and releases changes to production, so you can deploy custom applications with greater predictability and less risk.

For more information, see App Engine Management Center.

ReleaseOps

ReleaseOps orchestrates the entire deployment process, including moving changes between instances, running scans and tests, and releasing changes, while giving release managers and teams transparency into each step. ReleaseOps increases the predictability and reliability of deployments and reduces the risk of releasing unwanted changes to production.

For more information, see ReleaseOps.

ServiceNow Otto for App Engine

ServiceNow Otto for App Engine brings AI capabilities to your custom applications at runtime. Once an application is deployed, users can access embedded AI agents, skills, and agentic workflows to complete tasks directly within the applications they use every day. Because these AI assets are part of the application, they scale with your users and can help to improve productivity and efficiency across your organization.

For more information, see ServiceNow Otto for App Engine.

App Engine Management Center

The App Engine Management Center (AEMC) provides an innovative experience for you to manage the full life cycle for custom app development. You can use AEMC to handle application intake, configure pipelines, oversee developer productivity, and view insights about custom applications across your development environments, all in one centralized location.

The App Engine Management Center (AEMC) is a comprehensive tool designed for IT administrators and developers to handle the full life cycle of custom applications within your ServiceNow environment. AEMC serves as a single, centralized location for managing app development tasks, starting from idea submission all the way to deployment and continuous monitoring. AEMC helps to make the development process more efficient, strengthen governance, and offer clearer insights into custom app development throughout your organization.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Get started

Explore  Learn about AEMC concepts and features.	Configure  Configure AEMC, Application Intake, and Pipelines and Deployments.
Use  Use AEMC to manage app development and deployments.	

Additional resources for the App Engine Management Center

App governance product hub on ServiceNow Community

ServiceNow Community

Build and automate: App Engine

Govern low-code app development at scale with App Engine Management Center

Creator Toolbox

Creator toolbox | App Engine Management Center

Demo library

Govern low-code app development at scale with App Management Center

Exploring the App Engine Management Center

Learn about how the App Engine Management Center (AEMC) can help to centralize and streamline your application development and governance at scale.

App Engine Management Center overview

The App Engine Management Center (AEMC) is a powerful tool that innovates and unifies the application development, deployment, and governance experience. AEMC contains all of the features that you need to manage application development within your organization, from idea submission to deployment and then monitoring the app. AEMC supports development by enabling admins to manage requests for new apps, delegated development and collaboration permissions, and assign licenses for Developer Sandboxes. Admins can also oversee the deployment process for custom apps across their organization, with setup and control for several deployment options, including App Engine pipelines, ReleaseOps, and standalone pipelines.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

App Engine Management Center product tiers

There are tiered versions of the AEMC application. The base platform version of AEMC is available for free from the ServiceNow Store with access to standard set of AEMC features. A premium version of AEMC requires a paid subscription and has more features. Check your entitlements to determine whether you have access to AEMC platform or subscription features.

AI features in AEMC

Note: Depending on your license, you will have access to certain application features, generative AI skills, agentic workflows, and AI agents. For more information, see ServiceNow product tiers.

AEMC contains AI features that enable you to regenerate release notes. For more information, see Generate release notes.

ReleaseOps in AEMC

ReleaseOps is now integrated into AEMC. ReleaseOps automates and enhances the process of deploying changes, customizations, and custom applications on the ServiceNow AI Platform. To learn more about ReleaseOps, see Exploring ReleaseOps.

Developers working with update sets in version 28.1.2 of ServiceNow Studio and later can trigger ReleaseOps deployments directly within the development environment.

In development environments where full support isn't available yet, such as App Engine Studio and Creator Studio, you can still take advantage of ReleaseOps features in AEMC by migrating your existing App Engine pipeline to ReleaseOps. For more information about migrating to ReleaseOps, see the following resources:

- Migrating App Engine pipelines to ReleaseOps
- Migration tasks

App Engine Management Center users

Users

User	Description
Citizen and pro developers	• Apply for citizen development to submit ideas for new applications. • Submit collaboration requests to enable relevant stakeholders to be involved in the application development process. • Submit applications for review.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Users (continued)

User	Description
App Engine admins	• Configure pipelines that are tailored to an organization's unique needs. • Manage which user groups have access to development tools by reviewing application intake and collaboration requests. • Manage application deployment requests and testing, verifying that applications are ready to be moved to staging and production instances.

App Engine Management Center workflow

The workflow for managing app development and deployment through an App Engine pipeline using the App Engine Management Center is outlined in the following list:

1. Developers submit ideas for new applications to build.
2. App Engine admins review and approve application intake requests.
3. Developers and collaborators build custom applications using App Engine product offerings.
4. Once an app is ready, developers submit the app for review.
5. App Engine admins review and approve deployment requests, moving the app through their organization's unique pipeline.
6. App Engine admins review test suite logs to verify that the application is ready to be deployed to production instances.
7. App Engine admins approve apps for deployment to production instances.

What to explore next

To learn more about configuring and using App Engine Management Center, see:

- Configuring AEMC
- Using AEMC

App Engine Management Center user interface

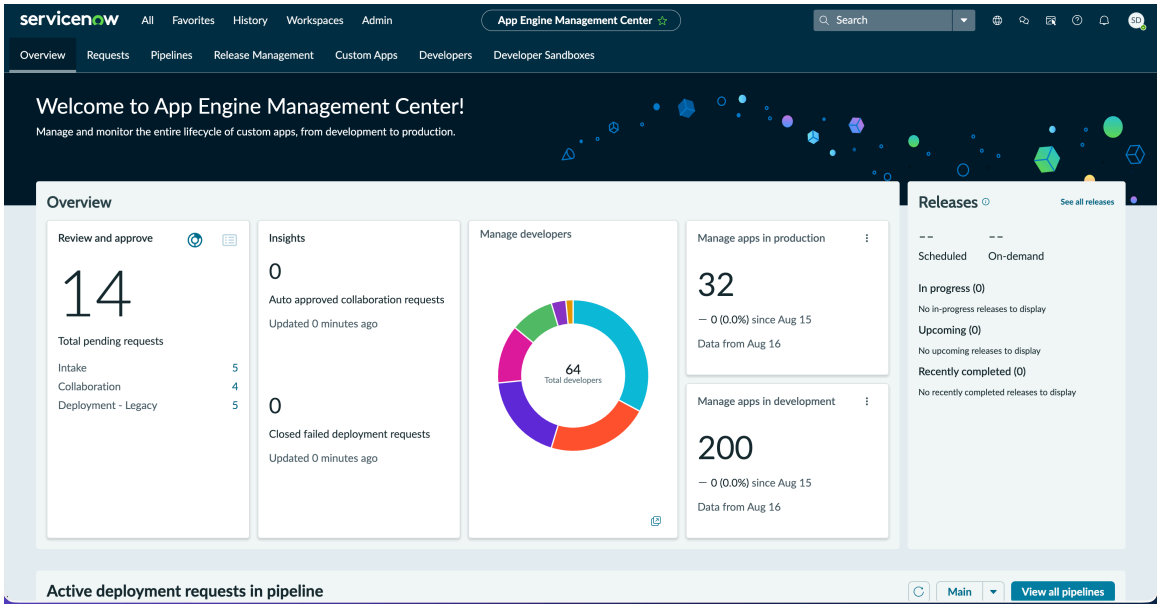
Learn about the App Engine Management Center user interface.

The App Engine Management Center provides a centralized experience for all of your application development, deployment, and governance tasks.

AEMC Overview page

The AEMC **Overview** page provides a high-level view of all intake, app, collaboration, and deployment requests for both App Engine and ReleaseOps deployments. You can also see insights into your developer and application metrics, and metrics about the releases that are in progress, upcoming, and already completed.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Navigating AEMC

From the AEMC **Overview** page, you can navigate to additional views: **Requests**, **Pipelines**, **Release management**, **Custom apps**, **Developers**, and *Developer Sandboxes* by selecting the view in the primary navigation bar.

AEMC primary navigation bar

Page	Description	Resources
Requests	Complete list of each request type broken down by approval status. Requests include: • Intake requests • App requests • Collaboration requests • Deployment requests for ReleaseOps deployments • Deployment requests - Legacy for App Engine deployments	Managing requests using AEMC
Pipelines	High-level view of the number of open deployment requests in each instance and within each pipeline. Selecting a pipeline or an environment in a pipeline displays all active deployment requests.	Configure Pipelines and Deployments

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

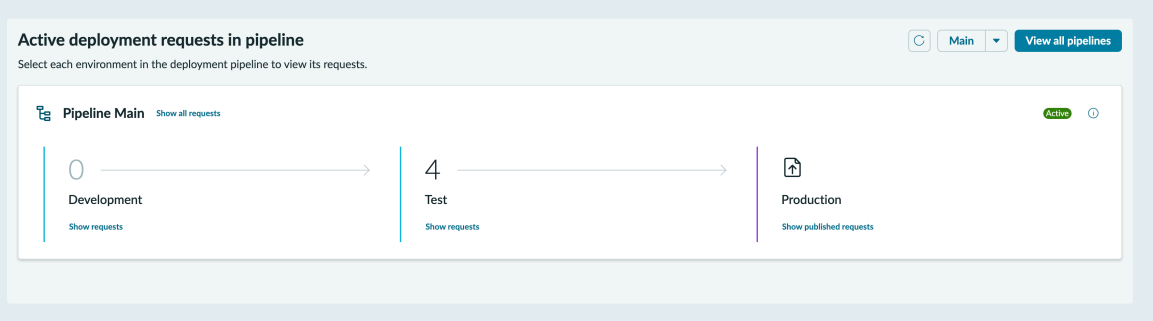
AEMC primary navigation bar (continued)

Page	Description	Resources
	Note: If you view deployment requests in a production instance, all **Closed - Published** deployment requests for that environment are listed.	
Release management	Available starting with version 28.2.1 of AEMC. Complete list of ReleaseOps releases and details about each release. When you select a release number, you can view the deployment requests associated with the release and monitor progress.	ReleaseOps
Custom apps	Charts and graphs illustrating the status of your active applications, and a list of all custom apps in development or production. You can view details for individual apps, including usage information, deployment history, and collaborators.	Manage custom apps using AEMC
Developers	Charts and graphs illustrating total and active developers, and developers by department. You can view details for individual developers, including the apps for which they are collaborators and their request history.	Managing developers using AEMC
Developer Sandboxes	Enables the assignment of Developer Sandboxes licenses to different instances.	Assign Developer Sandboxes packs

Active deployment requests

The Active deployment requests in the pipeline section of the AEMC Overview page shows the number of apps at each deployment stage within each of your pipelines. You can select which pipeline is displayed or select **View all pipelines**. For the selected pipeline, you can select **Show requests** or **Show published requests** next to the corresponding sections to see an overview of each group of requests.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



For a full picture of all your pipeline deployment requests, access the Pipelines page in AEMC. View all of your pipelines, quickly access each deployment request record, and filter each pipeline section to see only the requests that match your criteria.

Deployment process in AEMC

Learn about the deployment process in AEMC.

Deployment overview

The deployment process in AEMC is made up of several parts that make deploying changes possible. The following sections describe each part and how they work together to support the larger deployment process within AEMC.

Starting with version 28.2.1 of AEMC, ReleaseOps is integrated into AEMC. See the ReleaseOps deployment section for more information.

Submitting an application for deployment

The deployment process begins on a development instance when a developer submits an application in App Engine Studio, Creator Studio, or ServiceNow Studio. When an application is submitted for deployment, a deployment request is created, which enables admins to review and approve the application changes. For more information, see Deployment requests in AEMC.

Pipelines

Once an application has been submitted and a deployment request has been created, the application then begins to move through the pipeline that has been specified for the organization. A pipeline is the deployment structure that applications follow within AEMC. A pipeline consists of several instances, typically a development instance, test instance, and production instance, as well as the testing suites and frameworks that an organization has set up to verify that the application is stable and compatible before deploying to production. As the application is reviewed and tested, App Engine admins approve the application to move through pipeline environments until the application is finally deployed to a production or target instance.

With AEMC, you can configure pipelines that are tailored to the specific needs of your organization. AEMC supports any number of instances within each pipeline and as many pipelines as are needed for your organization. For example, if your organization requires that changes move through additional non-production instances prior to deploying to production, you can configure your pipeline so that it reflects each environment that you need. For more information about configuring pipelines, see Configure Pipelines and Deployments.

Change Management integration

You can integrate an existing Change Management program with your app deployment processes to add oversight into your deployments and have apps deploy according to a scheduled Change window. For more information, see Manage deployment requests.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Application Repository

AEMC uses Pipelines and Deployments to deploy applications through different instances. Pipelines and Deployments uses the Application Repository to manage these deployments. For more information about the Application Repository, see ServiceNow application repository.

Each application can either be deployed using the Application Repository or System Update Sets. If you have used update sets in the past, but want to switch to using the Application Repository, you can do a one-time conversion to deploy the app using the Application Repository instead. All apps don't have to follow the same deployment. For more information, see Convert custom applications to upgrade from the application repository and System update sets.

ReleaseOps deployment process

Starting with version 28.2.1 of AEMC, ReleaseOps is integrated into AEMC. Whether you are exclusively using ReleaseOps to manage your deployments, or you have migrated your App Engine pipelines to ReleaseOps, the deployment process differs from the legacy Pipelines and Deployments deployment process in AEMC.

If you are deploying exclusively using ReleaseOps, the deployment process for a scheduled release is as follows. is as follows:

1. The release manager creates a release, which is associated with a pipeline.
2. Developers make changes through update sets. Once they mark an update set **Complete** and promote the update set for deployment, the update set is added to an existing deployment request or to a new deployment request, which is targeted to a release.
3. When the update sets within a deployment request are functional and ready to be deployed, a developer sets the deployment request state to the Ready to Assess.
4. The assessment playbook runs, during which Automated Test Framework (ATF) test suites and instance scans are run on the changes in the deployment request.
5. If there are failures during assessment, deployment tasks are created that the tester can sign off on or redirect to the developer to address.
6. Once the deployment tasks have been addressed, the deployment request is set to the Ready for Deployment state.
7. On the scheduled date for the release, the release playbook runs. All deployment requests in the Ready for Deployment state move through the pipeline to production.

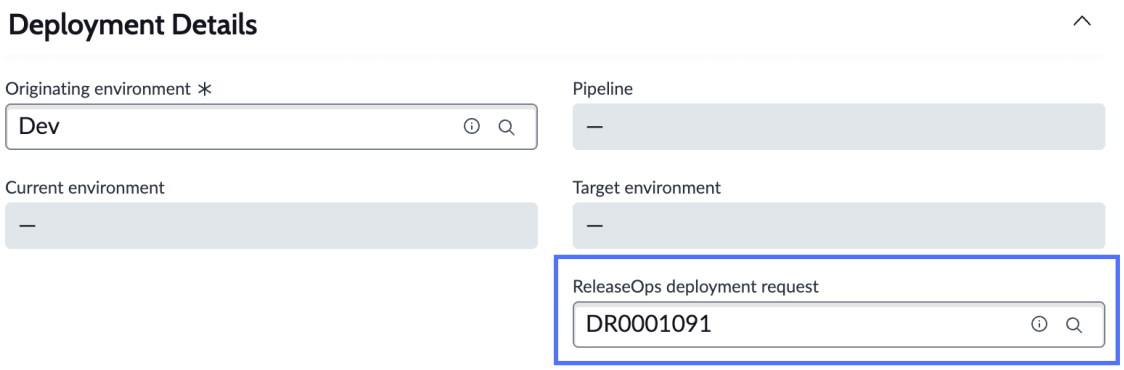
For more information about deploying with ReleaseOps, see Exploring ReleaseOps.

If you have migrated to ReleaseOps, the deployment process is as follows.

1. For applications being deployed via the Application Repository, developers submit applications from App Engine Studio, Creator Studio, and ServiceNow Studio using the standard process.
2. An App Engine deployment request is created and sent to an App Engine admin for approval. At the same time, a corresponding ReleaseOps deployment request is created.

   You can view the corresponding ReleaseOps deployment request record on the App Engine deployment request.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



3. Open the corresponding ReleaseOps deployment request and either attach it to an existing release or set the deployment request to on-demand. Then save the record.
4. An App Engine admin approves the App Engine deployment request from the App Engine pipeline, which hands off deployment to ReleaseOps.
5. An update set with install instructions is created and attached to the ReleaseOps deployment request.
6. Deployment requests in both the App Engine and ReleaseOps pipelines are synced, but ReleaseOps orchestrates the deployment.

To learn more about migrating to ReleaseOps, see:

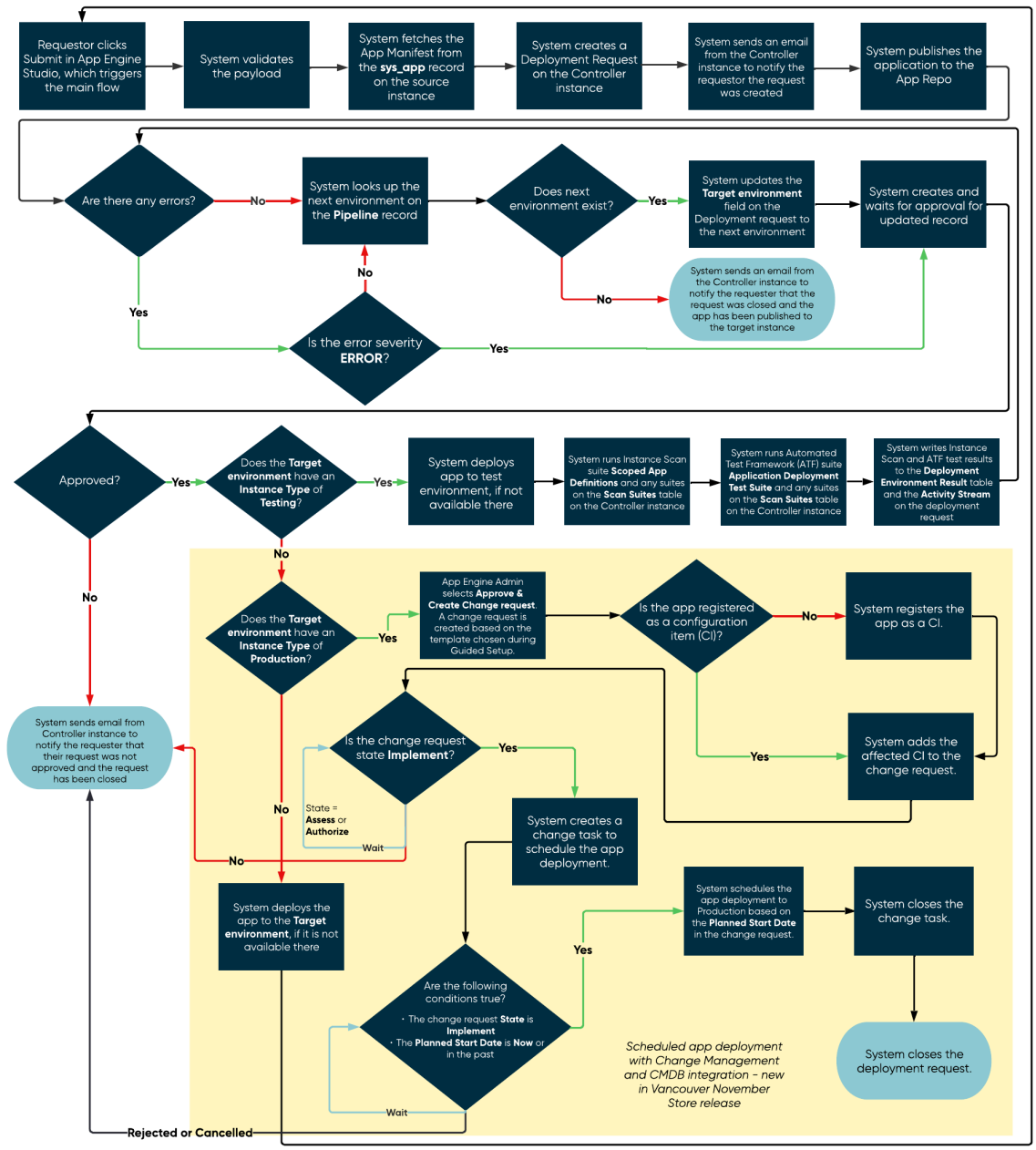
- Migrating App Engine pipelines to ReleaseOps
- Migration tasks

Pipelines and Deployments workflow version 24.1.2

As you manage requests for app deployment in App Engine Management Center (AEMC), use this workflow to understand how app deployments move through your pipelines in version 24.1.2, released in November 2023.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Pipelines and Deployments workflow



In this workflow:

1. The requester selects **Submit** in App Engine Studio, Creator Studio, or ServiceNow Studio, which triggers the main flow.
2. The system performs the following tasks behind the scenes:
   a. Validates the payload.
   b. Fetches the App Manifest from the **sys_app** record on the source instance.
   c. Creates a deployment request on the controller instance.
   d. Sends an email from the controller instance to notify the requester that the request was created.
   e. Publishes the application to the Application Repository.
3. The system performs different actions depending on if there are errors or not during publication.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. If there are errors in the app publication, and the error severity is **Error**, then the system creates and waits for approval for the updated record.

b. If there are no errors, or if there are errors, but the error severity isn't **Error**, then the system looks up the next environment on the Pipeline record.

i. If the next environment doesn't exist, the system sends an email from the Controller instance to notify the requester that the request was closed and the app has been published to the target instance. This action ends the workflow.

ii. If the next environment does exist, the system updates the **Target environment** field on the deployment request to the next environment. Then, the system creates and waits for approval for the updated record.

4. The system performs different actions depending on if the new record was approved.

a. If the record isn't approved, the system sends an email from the Controller instance to notify the requester that the request wasn't approved and the request has been closed. This action ends the workflow.

b. If the record is approved, and if the **Target environment** is **Testing**, then the system performs the following actions:

i. Deploys app to test environment, if the app is not available there.

ii. Runs the Scoped App Definitions Instance Scan and any other suites on the Scan Suites [[scan_suites]] table on the Testing instance.

Note: The Scan Suites table should be populated on the Controller instance.

iii. Runs the Application Deployment Test Suite Automated Test Framework (ATF) suite and any suites on the Scan Suites [scan_suites] table on the Testing instance.

iv. Writes Instance Scan and ATF test results to the Deployment Environment Result table and to the Activity Stream on the deployment request.

v. Returns the workflow to step 3, where it checks for errors.

c. If the record is approved, and the **Target environment** is **Production**, the app begins the scheduled deployment process with Change Management integration.

i. The App Engine admin selects **Approve & Create Change request**. A change request is created based on the template chosen during Guided Setup.

ii. The system performs different actions depending on if the app is registered as a configuration item (CI).

1. If the app is not registered as a CI, the system registers the app as a CI and then adds the affected CI to the change request.

2. If the app is registered as a CI, the system adds the affected CI to the change request.

iii. The system performs different actions depending on if the change request is in the **Implement** state.

1. If the change request state is not **Implement**, and the state is not **Assess** or **Authorize**, the system sends an email from the Controller instance to notify the requester that their request was not approved and has been closed. This ends the workflow.

2. If the change request state is not **Implement**, and the state is **Assess** or **Authorize**, the system waits for the state to be **Implement**.

3. If the change request is in the **Implement** state, the system creates a change task to schedule the app deployment.

iv. If the change request state is **Implement** and the **Planned Start Date** is not **Now** or in the past, the system waits for those two conditions to be true

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

v. If the change request state is **Implement** and the **Planned Start Date** is **Now** or in the past, but the request is **Rejected** or **Cancelled**, the system sends an email from the Controller instance to notify the requester that their request was not approved and has been closed. This ends the workflow.

vi. If the change request state is **Implement** and the **Planned Start Date** is **Now** or in the past, the system schedules the app deployment to Production based on the **Planned Start Date** in the change request. The system closes the change task, and then closes the deployment request. This ends the workflow.

d. If the record is approved, and the **Target environment** isn't **Testing** or **Production**, then the system deploys the app to the target environment, if it isn't available there.

The workflow starts over when a requester selects **Submit** again in App Engine Studio, Creator Studio, or ServiceNow Studio.

Deployment requests in AEMC

AEMC enables you to deploy changes to production through deployment requests. Learn more about what deployment requests are and where to view them in AEMC. Starting with version 28.2.1 of AEMC, you can also manage ReleaseOps deployment requests from AEMC.

App Engine deployment requests

The following information applies to App Engine deployment requests. For information about ReleaseOps deployment requests, see the ReleaseOps deployment requests section.

When a developer submits an application in App Engine Studio, Creator Studio, or ServiceNow Studio, a deployment request is created. A deployment request is a record to track the review of submitted applications.

A reviewer can deploy the app to a test environment, reject it, or publish it, all using the App Engine Management Center. If you integrate an existing Change Management program with AEMC, and your app is ready to deploy to production, you can create a change request and deploy the app within a specified change window.

1. View details of the deployment request and the application being deployed.
2. Select **Approve & Create Change Request** when the app is ready for deployment. This action begins the Change Management processes that you configure using Guided Setup.

   Note: If you don't have an existing Change Management program integrated with Pipelines and Deployments, select **Approve & Deploy App** to move the deployment to the next stage.

3. View and edit details of the deployment on the **Scheduled Deployments** tab.
4. View details of the change request on the **Change Request** tab. Depending on the role you have, you may be able to change some details of the request.
5. Watch the **Activity** stream for test results, change request progress, and other deployment details.

Note: If you submit deployment requests without upgrading all instances in the pipeline, your existing pipeline continues to be used.

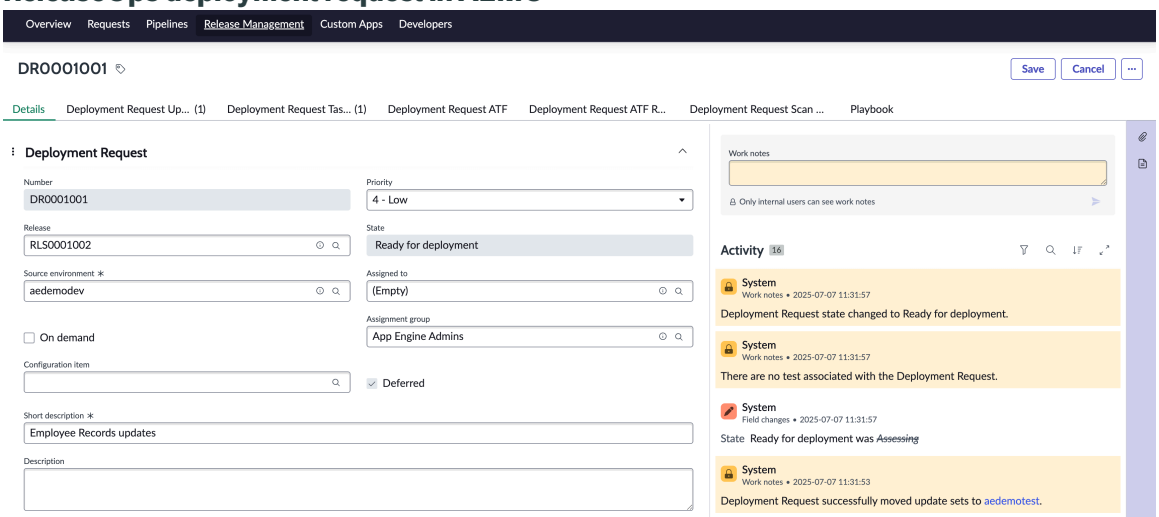
For more information on reviewing a deployment request, see Deployment Request form in the Pipelines and Deployments app and Manage deployment requests.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ReleaseOps deployment requests

Starting with version 28.2.1 of AEMC, you can also manage ReleaseOps deployment requests from AEMC. ReleaseOps deployment requests are created when deploying using ReleaseOps or when you have migrated an App Engine pipeline to ReleaseOps.

ReleaseOps deployment request in AEMC



A ReleaseOps deployment request references the release the deployment request is associated with. For more information about ReleaseOps releases, see Releases in ReleaseOps.

ReleaseOps deployment requests also contain the following information:

- The update set or update sets with the changes intended to deploy to production
- Any deployment request tasks, which are generated if a deployment request does not pass the assessment stage
- The Automated Test Framework (ATF) test suites that run during the deployment request assessment
- The results of the Automated Test Framework (ATF) test suites
- The Instance Scan suites that are run during the deployment request assessment

Testing applications in AEMC

Testing an application helps to ensure the viability of the changes on the production instance. Learn about how testing works in AEMC and the different kinds of tests that automatically run during the deployment process.

Testing an application

Before you publish a submitted application, test it in a non-production instance. To begin testing, an admin must open the deployment request and select **Approve**. The pipeline record is read and determines the next state for the request.

The goal of testing the application is to ensure the viability of the production instance. When the app has transitioned to the Testing state, ServiceNow Automated Test Framework tests run if you have enabled the appropriate properties.

If the application doesn't pass testing, then you reject the deployment request.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Deploying to a test environment

When you deploy or install an application to a test environment, two jobs are automatically executed:

- Application Deployment Test Suite
- Scoped App Definitions instance scan

These tests can be useful to the administrator for diagnosing issues before an app is deployed. For more information about what goes on during an application deployment, see Pipelines and Deployments workflow version 24.1.2.

Jobs that automatically run during deployment

Job	Description
Application Deployment Test Suite	A set of Automated Test Framework (ATF) tests. The suite consists of a single test called **Log**. When the test is run, scan results are logged to the Test Results [sys_atf_test_result] table. **Note:** The following two system properties must be enabled to run the Application Deployment Test Suite: • sn_atf.runner.enabled • sn_atf.schedule.enabled If they aren't enabled, only the instance scan runs. For more information, see Enable Automated Test Framework (ATF) properties.
Scoped App Definitions instance scan	Instance scans aid in diagnosing health issues on a non-production instance, and are useful for addressing best practices. For more information, see Instance Scan. The instance scan checks all tables in the app that extend Task [task].

The admin can view the results of the scans for troubleshooting purposes by selecting the **Deployment Environment Results** tab in the deployment request.

Simply open the record of the instance. The results are split into three tabs:

- **ATF Results**
  - Test suite name
  - Test URL
  - Success and failure counts
  - Overall error count
- **Instance Scan Results**

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

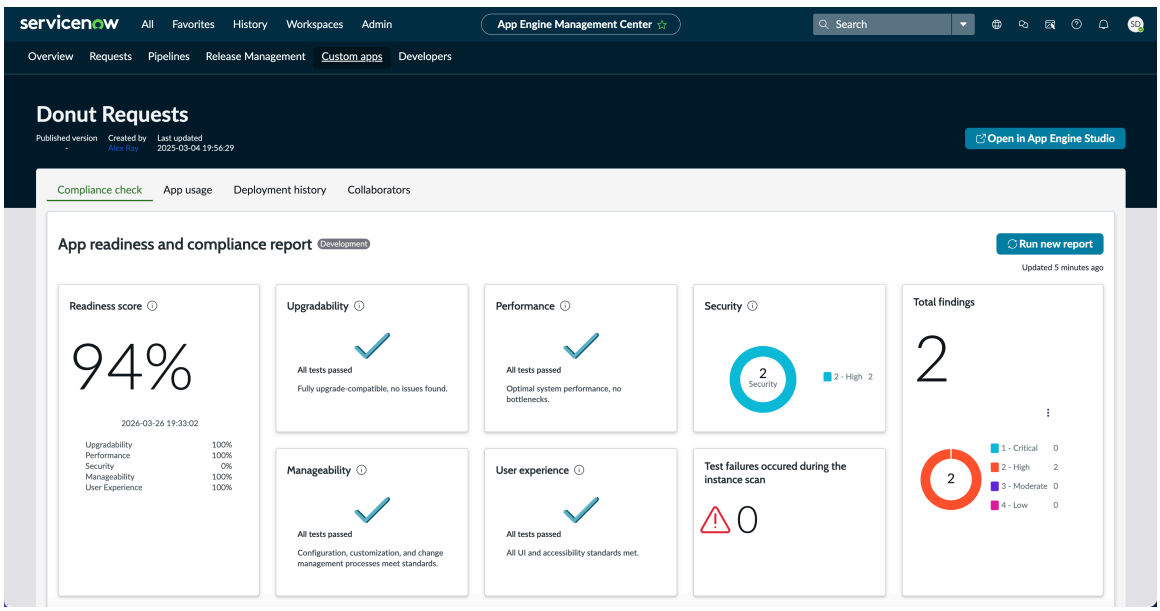
- Scan suite name
- Scan URL
- Finding count for the scan

- **Results (JSON)** includes unformatted JSON code of the ATF and instance scan results, as well as any errors identified during the scans. This JSON can also be found in the **Notes** related list.

App Readiness and Compliance Report

App Engine Admins can use the App Readiness and Compliance Report dashboard to check if the apps they're creating are ready to go live.

App Engine Admins can use the App Readiness and Compliance Report dashboard within the App Engine Management Center (AEMC) to verify that the apps they're creating are fully prepared for a live launch. This dashboard provides detailed insights from the Product Instance scans and helps to identify any issues or requirements that must be addressed before the app goes live, facilitating a smooth and compliant deployment process.



Testing and Validation Categories for the Readiness Score

The readiness score evaluates the following categories of testing and validation:

Readiness Score

The readiness score serves as an indicator for the overall validation checks. It provides a comprehensive assessment of how well a system or product meets the required standards across various categories.

Manageability

Checks for effective management and maintenance practices, including configuration, customization, and change management.

Security

Identifies vulnerabilities by assessing access controls, password policies, encryption, and security compliance.

User Experience

Evaluates UI performance, app responsiveness, and accessibility to provide a smooth user experience.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Upgradeability

Confirms readiness for ServiceNow upgrades by validating deprecated features and potential breakpoints.

Performance

Identifies system bottlenecks, optimizing database performance, script efficiency, and load responsiveness.

Test failure count

Shows the number Instance scan checks that failed to run.

Note: You can create or modify checks, for instance scans in the **All** > **Instance Scans** > **Suites** > **Best Practices- Parent**.

Run a new app readiness report

To run a new app readiness report, select the **Run new report** button in the dashboard. After generating a report, you can see the readiness status of the app.

Migrating App Engine pipelines to ReleaseOps

Starting with version 28.2.1 of AEMC, you can migrate your existing App Engine pipelines to ReleaseOps. Learn more about the migration process and benefits of migrating to ReleaseOps.

When you migrate to ReleaseOps, you enable ReleaseOps to handle the deployment process in the background without disrupting your existing App Engine pipeline. Developers submit applications in their development environments as before. But with migration enabled, a corresponding ReleaseOps deployment request is created at the time of application submission, which enables ReleaseOps to orchestrate the deployment process. To learn more about the deployment process, see Deployment process in AEMC.

Benefits of migration

Migrating an App Engine pipeline to ReleaseOps enables you to take advantage of ReleaseOps features without disrupting your existing App Engine pipeline or deployment process. This approach is helpful if you want to:

- Evaluate ReleaseOps before fully adopting it.
- Continue using the Application Repository for deployments.
- Deploy with ReleaseOps in environments where full support isn't available yet, such as App Engine Studio and Creator Studio.

Migration process overview

To migrate to ReleaseOps, you must have ReleaseOps installed and configured on each instance in your App Engine pipeline. Once you have configured ReleaseOps and the system property to enable migration, developers initiate deployment as usual. However, with migration enabled, deployment is handled by ReleaseOps in the background. The following list outlines the entire process for migrating to ReleaseOps:

1. Install ReleaseOps on each environment (instance) in your App Engine pipeline.
2. Complete ReleaseOps setup by configuring a new ReleaseOps ecosystem and configuring multi-instance management.
3. Configure the system property to enable migration to ReleaseOps.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

4. Map deployment request states if you're using custom deployment request states in your ReleaseOps pipeline. Base system ReleaseOps deployment request states will automatically map onto the corresponding App Engine deployment request states.
5. If you want to add additional conditions to enable ReleaseOps migration, you can set up custom conditions in Workflow Studio. For example, you might want only deployments for certain applications to migrate to ReleaseOps. So you can set up a condition that checks if the deployment request is from that specific application.

Once you have migrated to ReleaseOps, ReleaseOps handles the deployment process in the background. To learn more about the deployment process after migrating to ReleaseOps, see Deployment process in AEMC.

Get help with App Engine Management Center

To get help with App Engine Management Center, your ServiceNow instance, plugins, permissions, and more, watch a short video to contact the ServiceNow admin who works in your company.

Get help now from your Now Support administrator

The ServiceNow admin in your company can help you with things like activating plugins, kicking off upgrades, unlocking accounts, updating user permissions, and much more.

https://player.vimeo.com/video/1056112088?h=1a5cd3b245&badge=0&autopause=0&player_id=0&app_id=58479

Helpful resources

Some ServiceNow resources that can provide you with helpful information are:

App governance product hub on ServiceNow Community

ServiceNow Community

Support

Configuring AEMC

Learn about the configuration process for AEMC.

AEMC configuration overview

To start using AEMC, you must install AEMC on your instances and configure AEMC and related features, such as Application Intake, Pipelines and Deployments, ReleaseOps, and Developer Sandboxes.

The following list outlines the process for configuring AEMC and related features:

1. Install AEMC on each instance that will participate in your AEMC deployment pipeline (for example: development, test, production etc.).
2. Configure the App Engine Management Center by completing guided setup.
3. Configure Application Intake to set up who can submit requests for new applications and what questions appear on the application intake form.
4. Choose from and configure one of the three deployment options:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Configure Pipelines and Deployments
- Configure ReleaseOps in AEMC
- Configure a standalone environment

5. Test App Engine Management Center functionality on a non-production instance to verify that AEMC is working as expected, before you start deploying changes to production.

Configure the App Engine Management Center

Use the App Engine Management Center (AEMC) guided setup to step through the initial configuration of the Application Intake and Pipelines and Deployments applications. The Application Intake guided setup is optional, but if you want to use AEMC, the Pipelines and Deployments guided setup is required.

Before you begin

Role required: admin

Ensure that AEMC is installed on your production instance and all of your non-production instances (development, test, and the like).

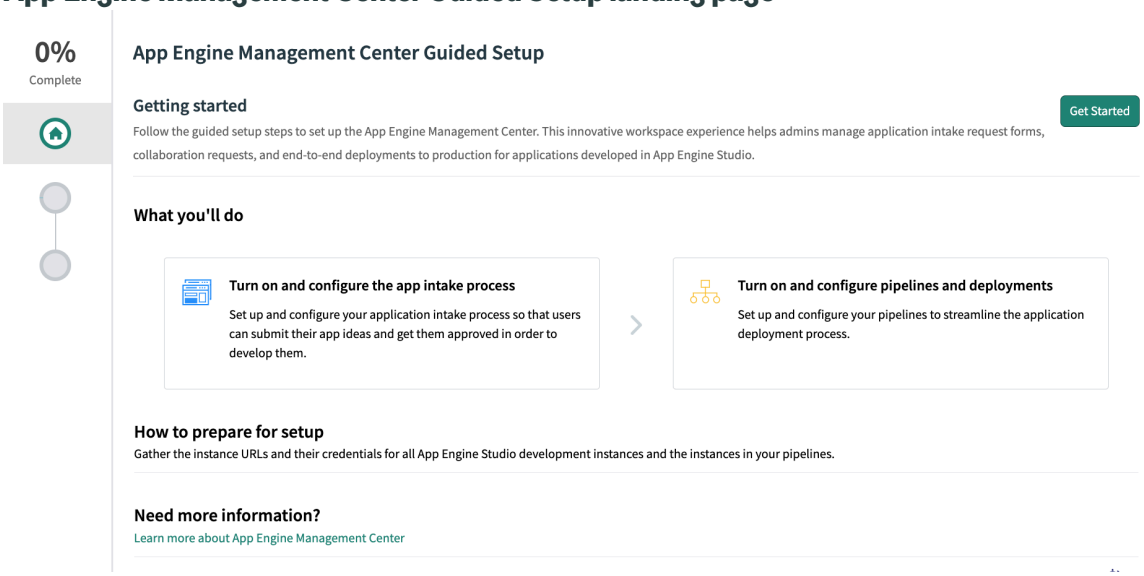
About this task

Essentially, the AEMC guided setup contains the setup steps for both the Application Intake and Pipelines and Deployments applications. If you have already configured one or both of these applications, you can skip those steps within the guided setup.

Procedure

1. On your production instance, navigate to **All** > **App Engine** > **Administration** > **Guided Setup**.
   The landing page provides information on the two applications that you must configure, Application Intake and Pipelines and Deployments.

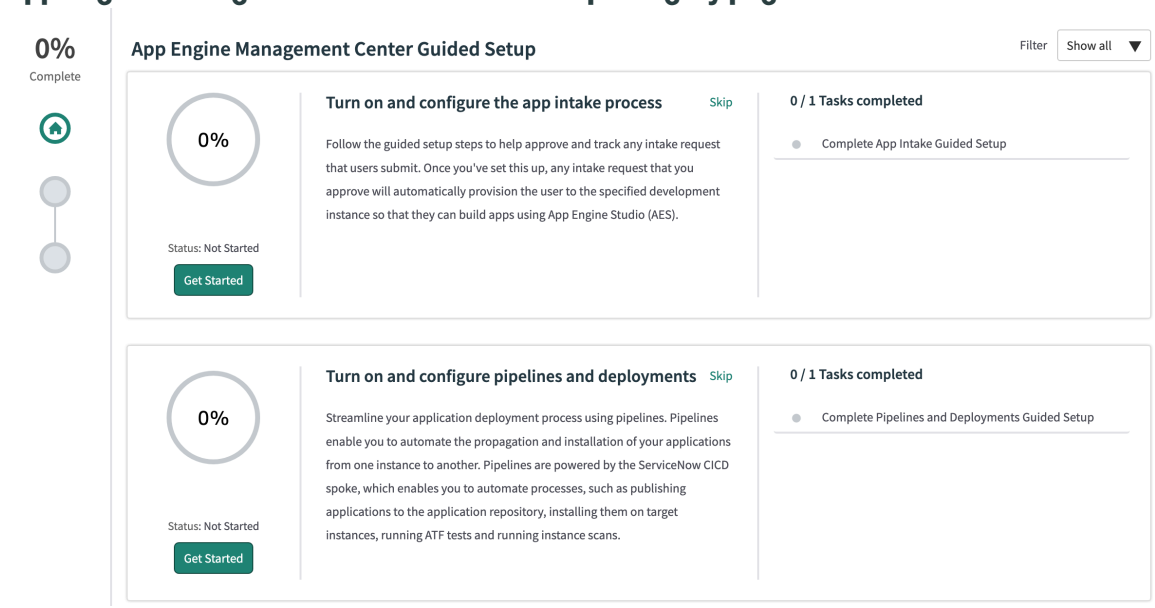
App Engine Management Center Guided Setup landing page



2. Initiate guided setup by selecting **Get Started**.
   The App Engine Management Center guided setup category page provides a list of different categories of tasks that you must complete before you can use AEMC.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

App Engine Management Center Guided Setup category page



Note: If you have previously started any of the guided setup tasks, and then exited without completing them, the **Get Started** button is labeled **Continue**.

3. Select the first **Get Started** button to initiate the Application Intake Guided Setup.
   There are several Application Intake configuration tasks you must complete.
   a. Activate the Apply for Citizen Development catalog item.
   b. Add users to the App Engine Admin group.
   c. Create development environment records.
   When you've completed all of the tasks in this category, the Category screen reopens.
4. Select the next **Get Started** button to initiate the Pipelines and Deployments Guided Setup.
   There are several Pipelines and Deployments configuration tasks you must complete.

   On your production instance:

   a. Configure environment credentials.
   b. Configure your pipeline environments.
   c. Configure your pipeline.
   d. Add users to the App Engine Admin group.
   e. Add ATF and instance scan suites for testing.

   Note: You can add ATF and instance scan suites for testing only if you've already created them and set up your testing instance. If you have not done this yet, you must come back to this step.

   f. Enable Change Management integration.
   g. Configure properties to integrate Change Management.

   On your testing instance:

   a. Configure environment credentials.
   b. Configure your controller instance.
   c. Enable Automated Test Framework (ATF) properties.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

d. Configure Automated Test Framework (ATF) suite.

e. Configure Instance Scan suite.

On your other, non-production instances:

a. Configure environment credentials.

b. Configure your controller instance.

When you have completed the tasks in the second category, the Category page reappears. You have completed guided setup for App Engine Management Center.

5. **Optional:** Configure additional properties used to control system behavior in AEMC.

Test App Engine Management Center functionality on a non-production instance

Test App Engine Management Center (AEMC) on a non-production instance to confirm that everything is working as expected before moving to production.

Before you begin

Role required: admin

About this task

To test AEMC and pipelines before you proceed to a production environment, pick two non-production instances for testing purposes. Follow the Pipelines and Deployments guided setup as if one of those non-production instances is a production instance. Because the system doesn't verify that the instance type on the environment record is actually a production instance, you can set the instance type to **Production** for the second non-production instance.

Procedure

1. Begin the Pipelines and Deployments Guided setup tasks as detailed in Pipelines and Deployments configuration tasks.
2. When you begin the tasks to configure your pipeline environments, select one of your non-production instances and designate it as a production instance in the pipeline environment record.
   For more information on the usual process, see Configure your pipeline environments.
3. Test AEMC and your pipeline functionality.
4. When you're done testing, change the **Instance type** of the non-production pipeline environment record to the correct type of instance.

Configure Application Intake

Use the App Engine Studio (AES) Application Intake guided setup to step through the initial configuration of the Application Intake application. Detailed instructions for each step are provided in subsequent sections of the product documentation.

Before you begin

Before you can use Application Intake to submit application ideas, you must ensure that the App Engine Studio application is installed.

Role required: admin

About this task

Application Intake guided setup provides a sequence of tasks that help you configure the Application Intake app on the ServiceNow AI Platform. For more information on each task, see Application Intake configuration tasks.

For general information about guided setup, see Guided Setup.

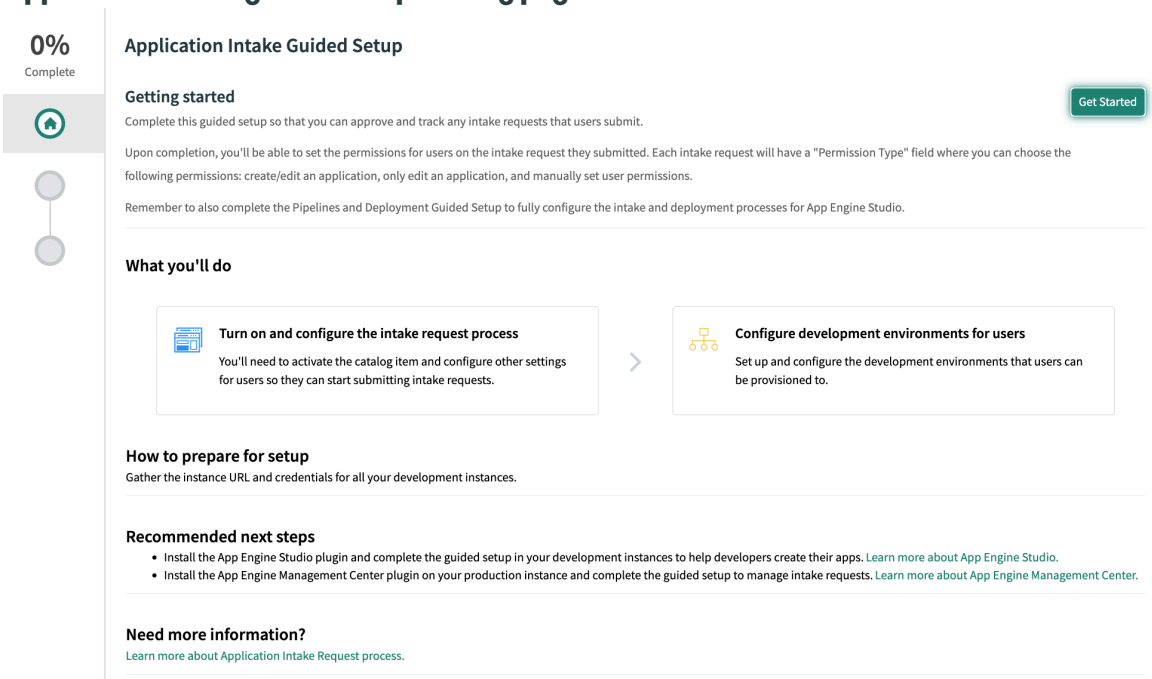
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **App Engine** > **Application Intake** > **Guided Setup**.

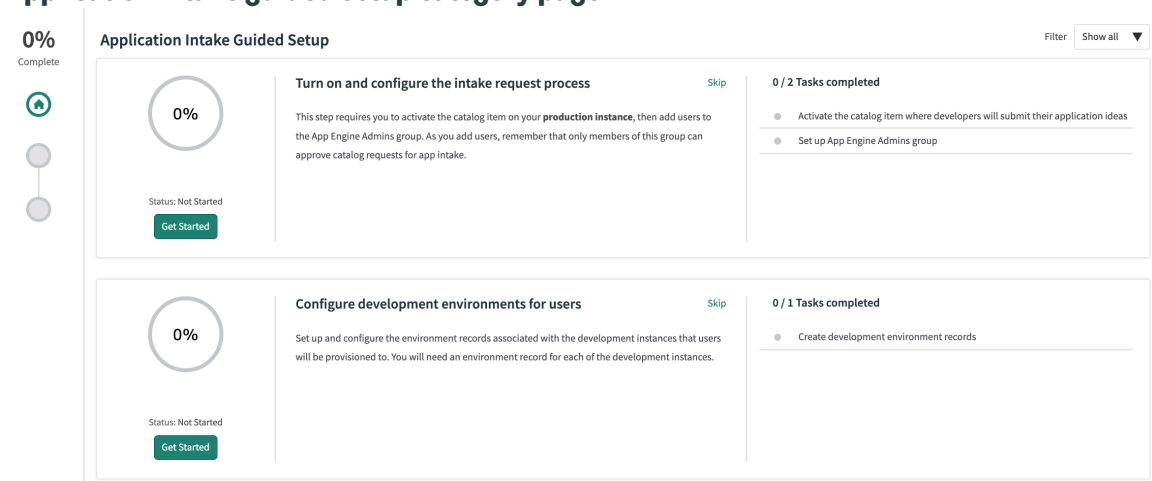
   The landing page provides information on the different categories, tools, and user access.

Application Intake guided setup landing page



2. To initiate guided setup, select **Get Started**.
   The Application Intake Guided Setup page provides a list of different tasks you must complete before you can use Application Intake to submit requests.

Application Intake guided setup category page



> **Note:** If you have previously started any of the guided setup tasks, and then exited without completed them, the **Get Started** button is labeled **Continue**.

3. Select the first **Get Started** button to initiate the Application Intake Guided Setup.
   There are several Application Intake configuration tasks you must complete.
   a. Activate the Apply for Citizen Development catalog item.
   b. Add users to the App Engine Admin group.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

When you've completed all of the tasks in this category, the Category screen reopens.

4. Select the next **Get Started** button to begin performing tasks for configuring development environments for your users.
   On your production instance, Create development environment records for each development instance that you want to provision users to. This process allows your production instance to connect to your development instances.

   Note: If these records have already been set up in the Pipelines and Deployments Guided Setup, you can skip this step. When you have completed the tasks in the second category, the Category screen reappears.

   Congratulations! You have completed guided setup for Application Intake.

Application Intake configuration tasks

As you work through the App Engine Studio (AES) Application Intake guided setup, you must perform different configuration tasks.

Activating and configuring the intake request process, and setting up development environments

The tasks for configuring Application Intake are listed below, along with links to detailed instructions for completing them.

Application Intake configuration tasks

Task	Description
Activate the catalog item where developers will submit their application ideas	Activate the Apply for Citizen Development catalog item so that citizen developers can submit their ideas for applications. For detailed instructions, see Activate the Apply for Citizen Development catalog item.
Set up the Apply for Citizen Development catalog item	For quick setup and use, use the predefined Apply for Citizen Development catalog item in Catalog Builder. To customize your app intake form, either update the Apply for Citizen Development catalog item, or use the AES App Intake template as a starting point and add questions and fields as needed. For detailed instructions, see Customize the App Intake form in Catalog Builder.
Add users to the App Engine Admins group	Add users to the App Engine Admin group. After the email address of the App Engine Admin group is configured, members can receive notifications for app development-related requests, including application intake requests. **Note:** If you already added users on the production instance using AES guided setup or Configure Pipelines and Deployments guided setup, you can skip this step. For detailed instructions, see Add users to the App Engine Admin group.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Application Intake configuration tasks (continued)

Task	Description
Create development environment records	On your production instance, create an environment record of type "Development" for each development instance that you want to provision users to. This will allow your production instance to connect to your development instances. **Note:** If these records have already been set up in the Pipelines and Deployments Guided Setup, you can skip this step. For detailed instructions, see Configure your pipeline environments.

When you have completed Application Intake guided setup, proceed to Pipelines and Deployments guided setup to fully configure App Engine Studio.

Activate the Apply for Citizen Development catalog item

Enable citizen developers to submit their ideas for applications by activating the Apply for Citizen Development catalog item. This is the first step in configuring the App Engine Studio (AES) Application Intake app.

Before you begin

Role required: admin

Procedure

1. Navigate to **All** > **App Engine** > **Application Intake** > **Guided Setup**.
   For more information, see Configure App Engine Studio.
2. Either start the process or continue with a process in progress.
   - To start the process, select **Get Started**.
   - If you previously completed one or more of the tasks, select **Continue**.
3. In the **Turn on and configure the intake request process** section, select **Get Started**.
4. In the **Activate the catalog item where developers will submit their application ideas** section, select **Configure**.
5. In the Apply for Citizen Development catalog item record, select the **Active** check box.
6. Select **Update**.
7. Close the browser window and return to the Application Intake guided setup.

Customize the App Intake form in Catalog Builder

Create a custom app intake experience for your organization by editing the fields and questions on the App Engine Studio (AES) App Intake form in Catalog Builder.

Before you begin

Role required: admin

About this task

The Out of the Box variables associated with this catalog item are set to read-only. To edit the App Intake form, you must create a copy of the existing **Apply for Citizen Development** catalog

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

item. Then, disable the original catalog item, and continue editing the **Copy of Apply for Citizen Development** catalog item. For more information, see Copy a catalog item.

Procedure

1. Navigate to **All** > **Service Catalog** > **Catalog Builder**.
2. On the **Catalog Items** tab, select the copy of the **Apply for Citizen Development** item you created.
3. Select **Edit**.
4. Define the questions on the App Intake form on the **Questions** tab.

Action	For more information
Insert a new question	Create a question for a catalog item in Cata
Edit an existing question	
Change the order of the questions	Change the order of the questions
Define dynamic behavior for an existing question	Define dynamic behavior for an existing App Intake form question
Deactivate or remove a question	Deactivate a question from the App Intake form

5. Edit any other settings on the form that you want to change.
6. On the **Review and submit** tab, review the form preview and make any needed corrections by selecting **Preview**.
7. Save your changes.
   - To save your changes without submitting the form, select **Save**.
   - To submit the form, select **Submit**.

Change the order of the questions in the App Intake form

Customize the App Intake form by changing the order in which questions appear.

Before you begin

Role required: admin

Procedure

1. Navigate to **All** > **Service Catalog** > **Catalog Builder**.
2. On the **Catalog Items** tab, select **Apply for Citizen Development** from the list of catalog items.
3. Select **Edit**.
4. On the **Questions** tab, point to the question you want to move.
5. On the left side of the question, select the Row drag and drop gripper icon (⠿) and drag the question to a new position.

Define dynamic behavior for an existing App Intake form question

Define dynamic behavior for a question on the App Intake form based on answers to other questions.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin
Role required: admin

About this task
Sometimes, if a user gives a specific answer to one question on the App Intake form, you might need a follow-up question to gather more information. For example, if you ask if they have taken any ServiceNow training, and the answer is "no", the user goes to the next visible question. If the answer is "yes", another question appears asking which training courses they have completed. Use the following steps to define this type of dynamic behavior.

Procedure
1. Navigate to **All** > **Service Catalog** > **Catalog Builder**.
2. On the **Catalog Items** tab, select **Apply for Citizen Development** from the list of catalog items.
3. Select **Edit**.
4. On the **Questions** tab, point to the question you want to update.
5. Select the Add dynamic behavior icon ( ) next to the question that you want to update.
6. Select **Define new behavior**.
7. On the **Actions** tab, specify the behavior of the question when certain conditions are met.

Actions tab

Field	Description
Specify the behavior of this question when the trigger conditions are met	
Make the question mandatory	Specifies how the dynamic behavior setting affects the mandatory state of the field, with the following options: ○ No action ○ Yes ○ No
Make the question visible	Specifies how the dynamic behavior setting affects the visible state of the field, with the following options: ○ No action ○ Yes ○ No
Make the question read-only	Specifies how the dynamic behavior setting affects the read-only state of the field, with the following options: ○ No action ○ Yes ○ No
Clear value	Option to clear the question value.

8. On the **Conditions** tab, specify the conditions that trigger the action on the question. For more information about building condition statements, see Condition builder.
9. On the **Settings** tab, specify the scenarios where the behavior setting applies.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Settings tab

Field	Description
Applicability	
Scenarios where this dynamic behavior applies	
Applies when the item is being requested	Option to specify if the dynamic behavior applies when the item is being requested.
Applies while viewing the catalog tasks after the request is submitted	Option to specify if the dynamic behavior applies while viewing the catalog tasks after the request is submitted.
Applies while viewing the requested item record after the request is submitted	Option to specify if the dynamic behavior applies while viewing the submitted record.

10. Select **Add behavior**.

Deactivate a question from the App Intake form

Deactivate a question from the App Intake form if it's no longer needed.

Before you begin

Role required: admin

Procedure

1. Navigate to **All** > **Service Catalog** > **Catalog Builder**.
2. On the **Catalog Items** tab, select **Apply for Citizen Development** from the list of catalog items.
3. Select **Edit**.
4. On the **Questions** tab, point to the question you want to remove.
5. Remove the question from the form by selecting the Deactivate question icon (⊗).
6. Select **Deactivate**.

Manage user groups for Application Intake

Control which user groups are available for admins to give Creator Studio and App Engine Studio app development permissions to during the Application Intake process. These groups are managed on the User Groups Permission Types [sn_app_intake_permission_type] table.

Before you begin

Role required: admin

Procedure

1. Navigate to **All** > **sn_app_intake_permission_type.list**.
2. Select a record in the list to update it.
3. On the group record, change the status of the **Active** option to control whether the group is available in the Application Intake guided setup.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

What to do next

Confirm that you see the correct active groups in an application request in App Engine Management Center.

Configure ReleaseOps in AEMC

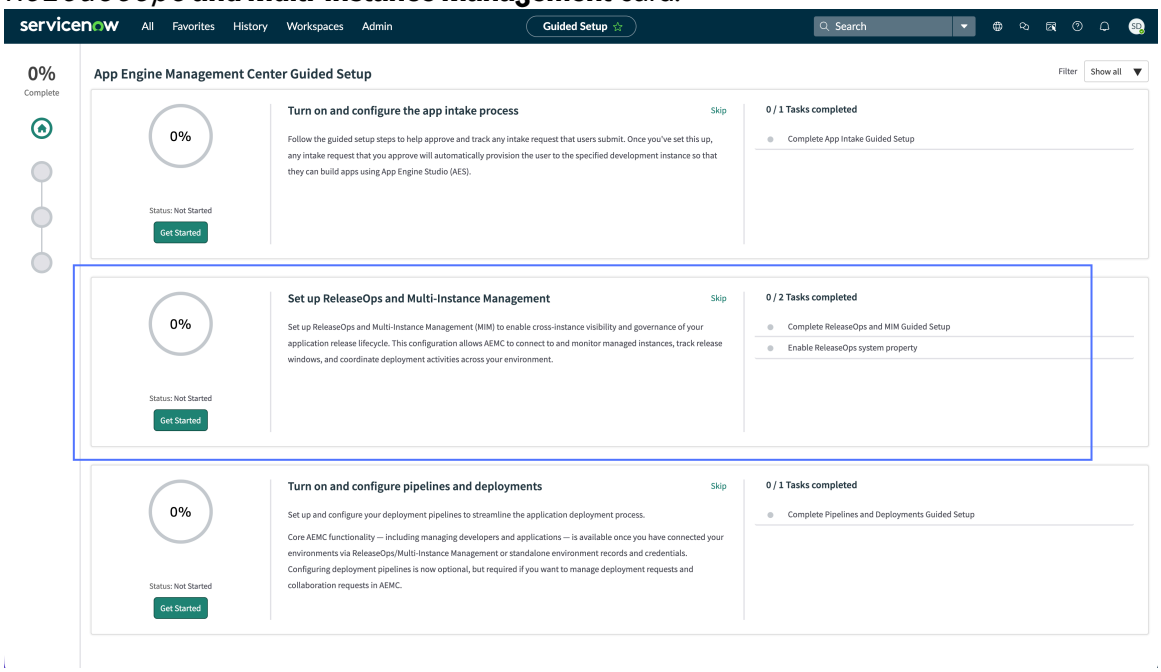
Complete ReleaseOps guided setup in AEMC to start using ReleaseOps for your deployments.

Before you begin

Role required: admin

Procedure

1. On your production instance, navigate to **All** > **App Engine** > **Administration** > **Guided Setup**.
2. Select **Get Started**.
3. On the App Engine Management Center guided setup page, select **Get Started** in the **Set up *ReleaseOps* and Multi-Instance Management** card.



4. Complete ReleaseOps guided set up in AEMC.
5. Enable ReleaseOps system property.
6. Run trust profile setup in AEMC.

Complete ReleaseOps guided set up in AEMC

Complete ReleaseOps guided setup in AEMC to configure your ReleaseOps ecosystem to handle your deployments.

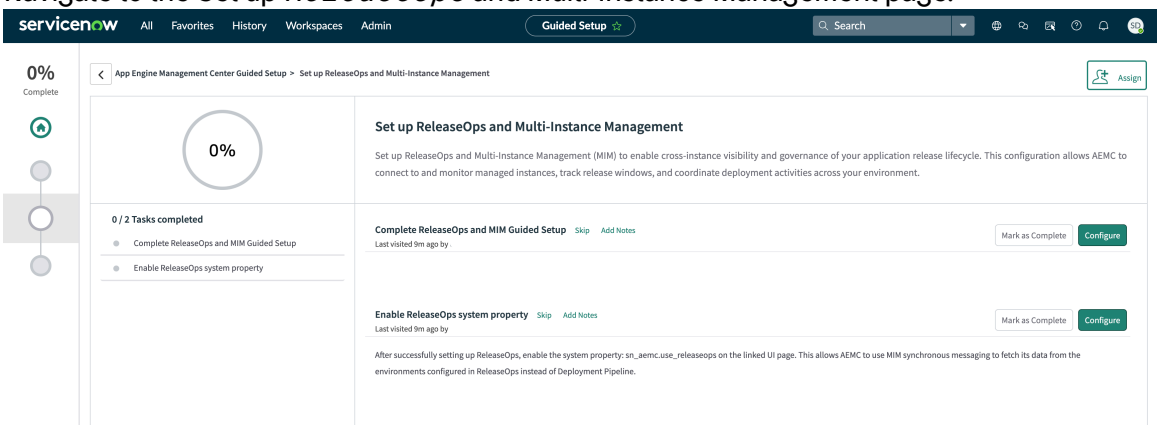
Before you begin

Role required: admin

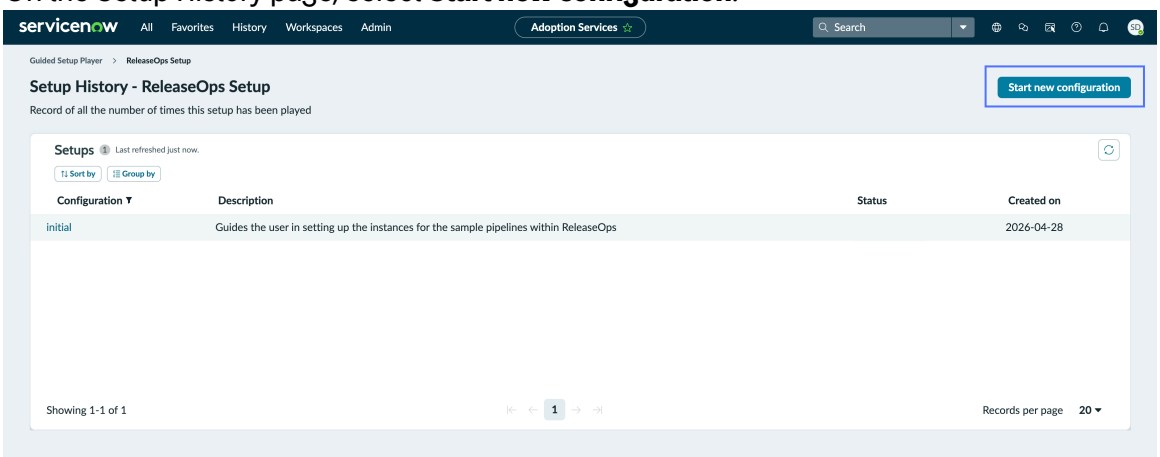
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

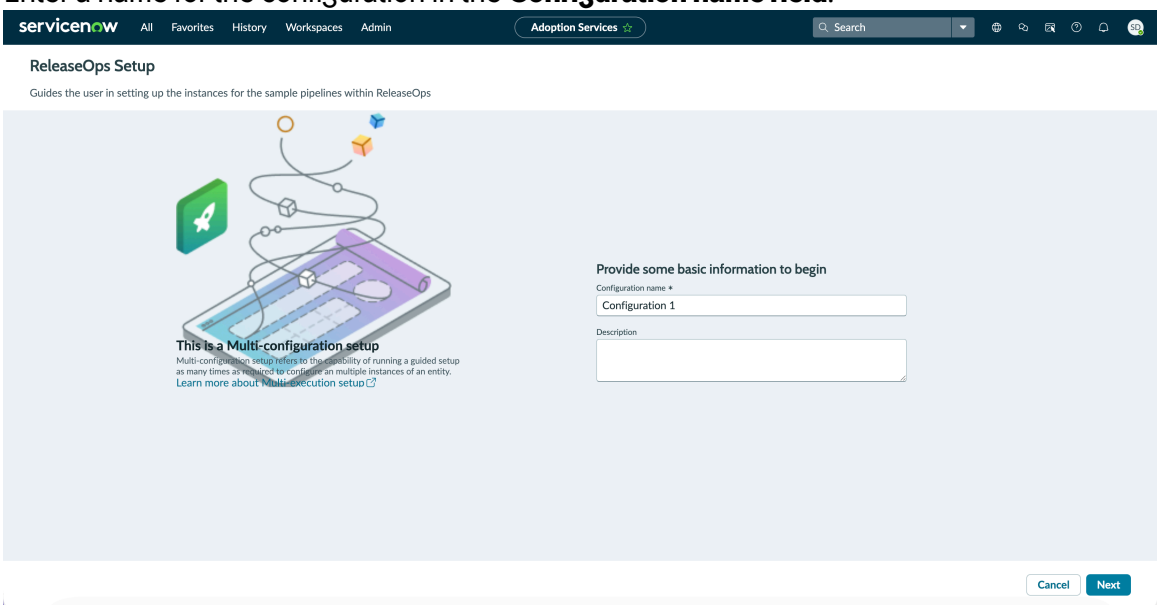
1. Navigate to the Set up *ReleaseOps* and Multi-Instance Management page.



2. In the **Complete** *ReleaseOps* **and MIM Guided Setup** step, select **Configure**.
3. On the Setup History page, select **Start new configuration**.



4. Enter a name for the configuration in the **Configuration name field**.



5. **Optional:** Enter a description for the configuration in the **Description** field.
6. Select **Next**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

When you select **Next**, you're automatically redirected to the ReleaseOps guided setup page.

7. Complete ReleaseOps guided setup.

What to do next

Complete *ReleaseOps* configuration in AEMC by enabling the *ReleaseOps* system property.

Enable ReleaseOps system property

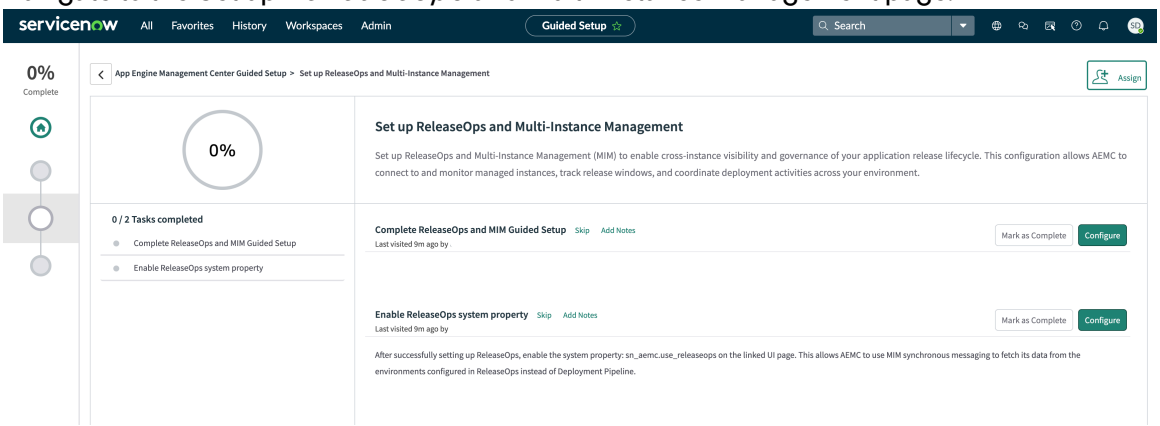
Enable the ReleaseOps system property to have AEMC fetch data from the environments configured in ReleaseOps.

Before you begin

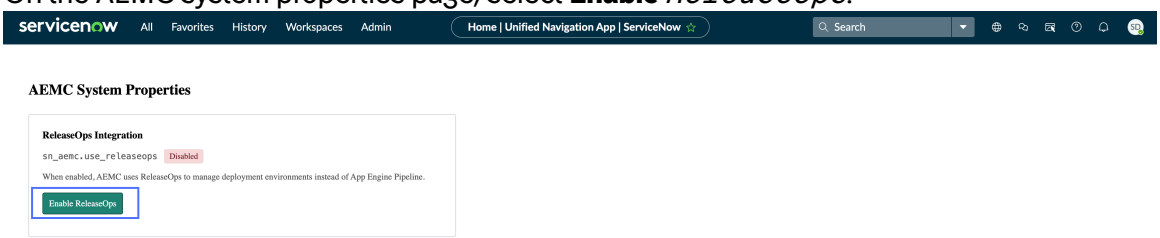
Role required: admin

Procedure

1. Navigate to the Set up *ReleaseOps* and Multi-Instance Management page.



2. In the **Enable** *ReleaseOps* **system property** step, select **Configure**.
3. On the AEMC system properties page, select **Enable** *ReleaseOps*.



Run trust profile setup in AEMC

After completing ReleaseOps guided setup in AEMC, run trust profile setup to complete the configuration process.

Before you begin

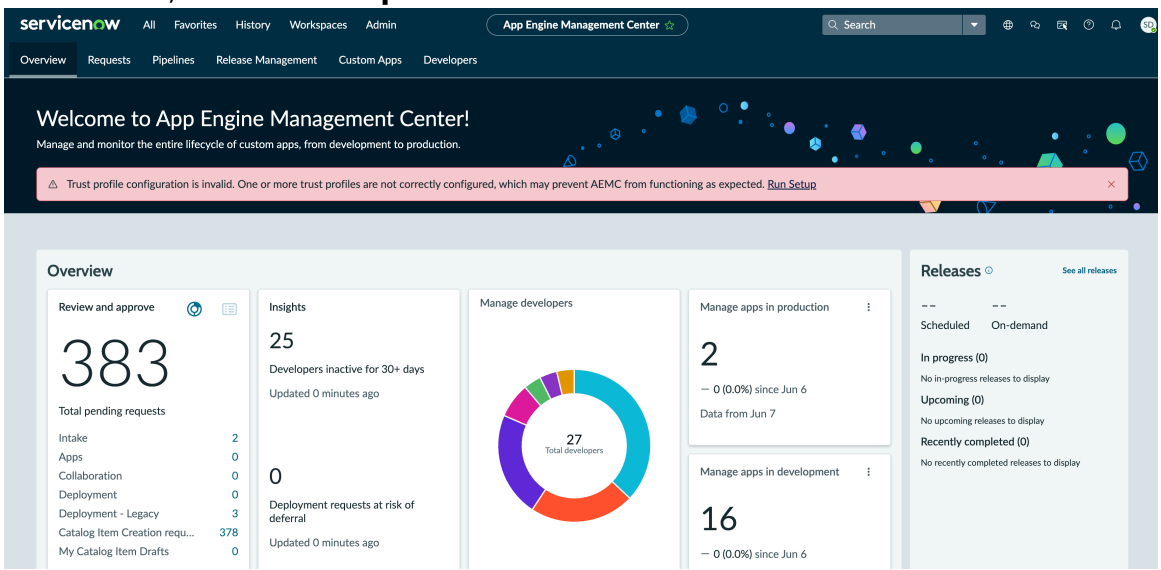
Role required: admin

After

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **App Engine** > **Administration** > **App Engine Management Center**.
2. In the banner, select **Run Setup**.

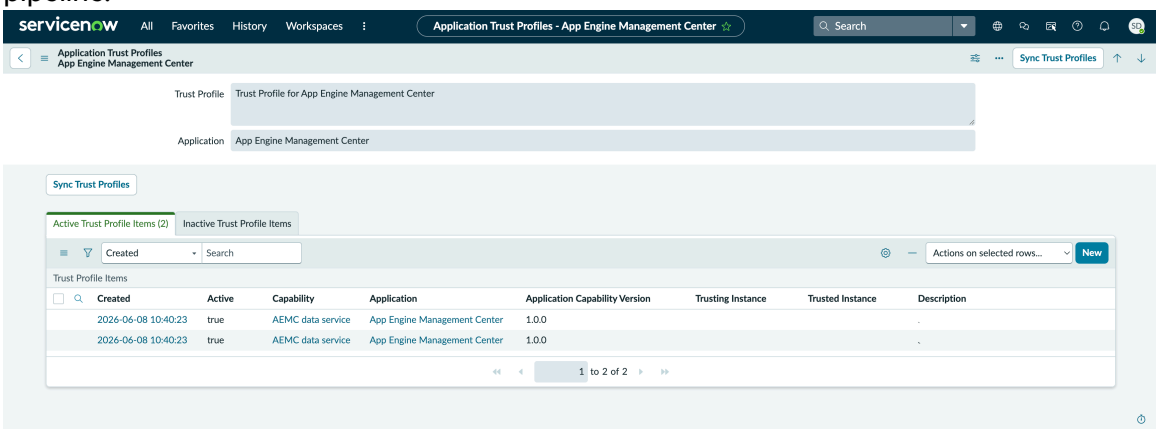


3. **Optional:** Verify that the trust profiles are set up.

   a. Navigate to **All** > **Multi-Instance Management** > **Application Trust Profiles**.

   b. From the list, select **Trust Profile for** *`App Engine Management Center`*.

   c. Verify that the Active Trust Profile Items list contains each instance in your ReleaseOps pipeline.



Configure Pipelines and Deployments

Use the Pipelines and Deployments guided setup to complete the initial configuration of Pipelines and Deployments. Detailed instructions for each step are provided in subsequent sections of the product documentation.

Before you begin

Role required: admin

About this task

The Pipelines and Deployments guided setup provides a sequence of tasks that help you configure Pipelines and Deployments on the ServiceNow AI Platform. For more information about each task, see Pipelines and Deployments configuration tasks.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Starting with version 28.2.1 of Pipelines and Deployments, you can migrate your existing pipelines to ReleaseOps. To learn more about the migration process and the advantages of migrating to ReleaseOps, see Migrating App Engine pipelines to ReleaseOps. To get started migrating your existing pipelines, see Migration tasks.

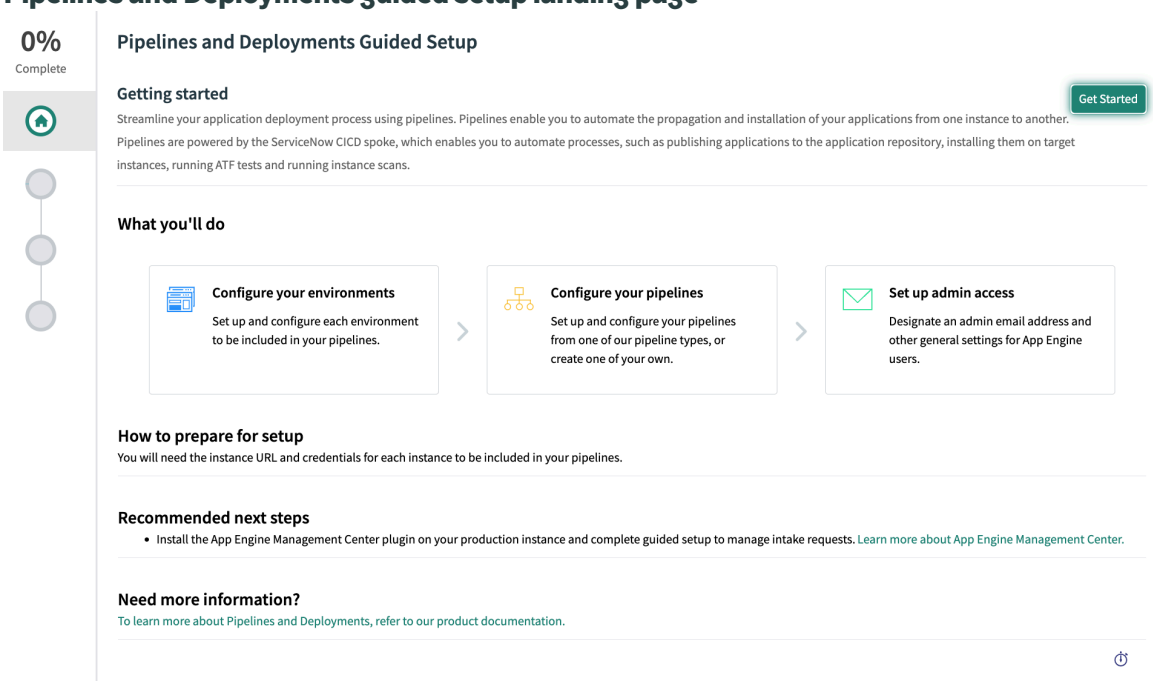
For general information about guided setup, see Using guided setup.

Procedure

1. Navigate to **All** > **App Engine** > **Pipelines and Deployments** > **Guided Setup**.

   The landing page provides information on the different categories, tools, and user access.

Pipelines and Deployments guided setup landing page



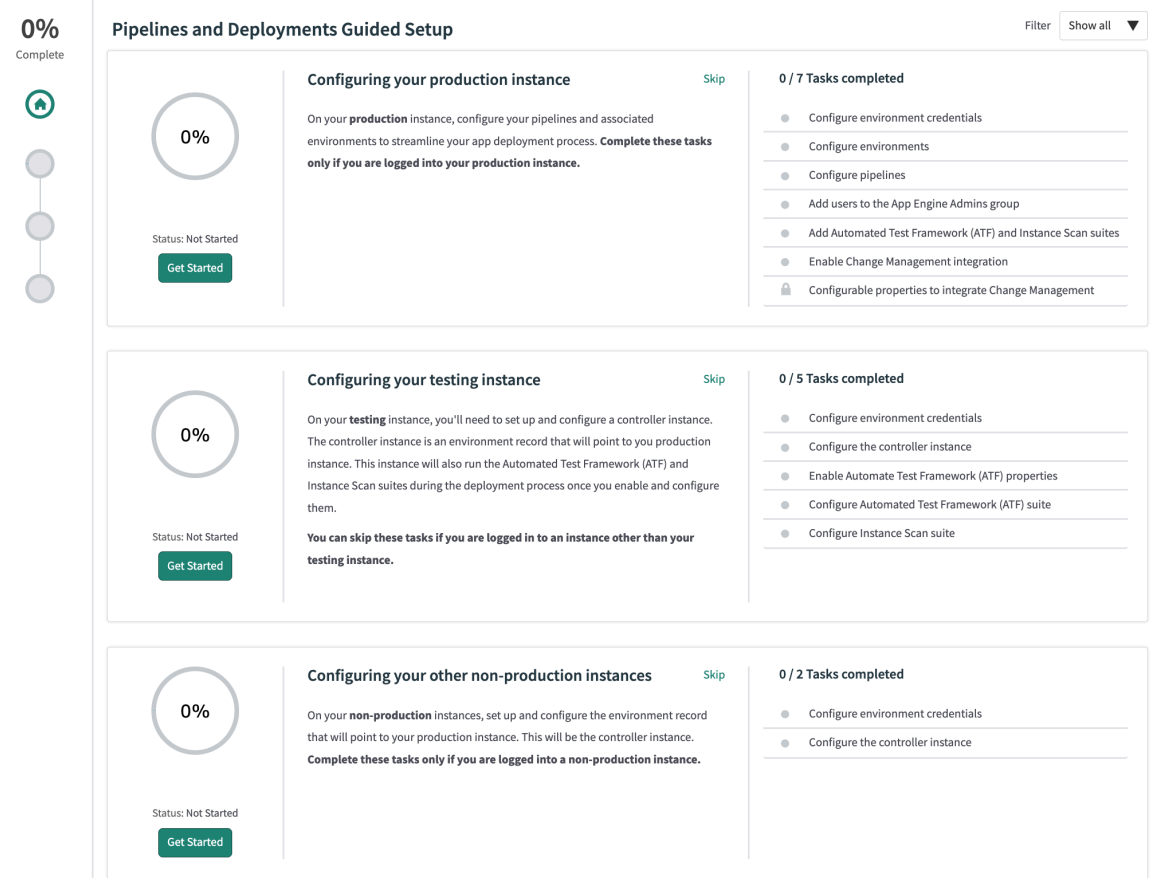
2. To initiate guided setup, select **Get Started**.

   The Pipelines and Deployments Guided Setup category page provides a list of different categories of tasks you must complete before you can use Pipelines and Deployments.

   Note: If you have previously started any of the guided setup tasks, and then exited without completing them, the **Get Started** button will be labeled **Continue**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Pipelines and Deployments guided setup category page



Important: Each of the tasks in the following list must be performed on different environments in your pipeline.

3. Select the first **Get Started** button to begin performing configuration tasks on your production environment.
   a. Configure environment credentials.
   b. Configure your pipeline environments.
   c. Configure your pipeline.
   d. Add users to the App Engine Admin group.
   e. Add ATF and instance scan suites for testing.

      Note: You can add ATF and instance scan suites for testing only if you've already created them and set up your testing instance. If you have not done this yet, you must come back to this step.

   f. Enable Change Management integration.
   g. Configure properties to integrate Change Management.

   When you have completed all of the tasks in this category, the Guided Setup screen reappears.

4. Select the next **Get Started** button to begin performing configuration tasks on your testing environment.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. Configure environment credentials.

b. Configure your controller instance.

c. Enable Automated Test Framework (ATF) properties.

d. Configure Automated Test Framework (ATF) suite.

e. Configure Instance Scan suite.

When you have completed all of the tasks in this category, the Guided Setup screen reappears.

5. Select the last **Get Started** button to begin performing tasks for all of the other non-production (that is, test, development, staging, and so forth) instances in your pipeline.

   a. Configure environment credentials.

   b. Configure your controller instance.
   Congratulations! You have completed guided setup for the Pipelines and Deployments application.

6. **Optional:** Migrate your existing pipelines to ReleaseOps.
   See Migration tasks for more information.

Pipelines and Deployments configuration tasks

As you work through the Pipelines and Deployments guided setup, you must perform different configuration tasks on each of your instances.

Setting up environments, pipeline records, and user access

The tasks for configuring Pipelines and Deployments are listed below, along with links to detailed instructions for completing them. These tasks build on each other, so it's important to complete them in the sequence listed in the Guided Setup.

Pipelines and Deployments configuration tasks

Instance	Task	Description
Production	Configure environment credentials.	Configure credentials for each instance in your pipeline so that the flows used by AEMC can access the instances in your pipelines. For more information, see Configure environment credentials.
	Configure environments.	Set up your production and non-production (that is, development, test, and/or staging) environments by adding the URLs and credentials used to access each instance. For more information, see Configure your pipeline environments.
	Configure pipelines.	Set up and configure each pipeline by naming the pipeline, identifying each of the environments associated with the pipeline, and specifying the order in which apps are deployed between the environments in that pipeline.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Pipelines and Deployments configuration tasks (continued)

Instance	Task	Description
		For more information, see Configure your pipeline.
	Add users to the App Engine admin group.	Add users involved with this flow of tasks to the App Engine admin group. **Note:** If already added users on the production instance using AES guided setup or Configure Application Intake guided setup, you can skip this step. For more information, see Add users to the App Engine Admin group.
	Add Automated Test Framework (ATF) and Instance Scan suites.	Add your customized test suites as part of an application deployment in addition to the instance scan and ATF testing suites that are standard in AEMC. For more information, see Add ATF and instance scan suites for testing.
	Enable Change Management integration	Connect an existing Change Management program with AEMC so that deployments through the pipeline to the production environment are automatically scheduled based on the Change request state and planned change window. For more information, see Enable Change Management integration.
	Configure properties to integrate Change Management	Change requests and configuration items (CIs) are created using specifications set up in your existing Change Management program. If you want to change the change model, change template, or CI creation subflows used during app deployments, you must configure these properties during Guided Setup. For more information, see Configure properties to integrate Change Management.
Testing	Configure environment credentials	For more information, see Configure environment credentials.
	Configure the controller instance	Identify the controller instance that contains the pipeline deployment flows, as well as all of the data associated with the pipeline. As each instance completes its assigned tasks, it communicates with the controller instance to determine where the next stage of deployment takes place.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Pipelines and Deployments configuration tasks (continued)

Instance	Task	Description
		Note: Your production instance should be identified as the controller. For more information, see Configure your controller instance.
	Enable ATF properties on instances of type Testing.	Enable system properties that allow the ATF suite to run during the deployment process. If you do not enable these properties, a warning displays during the deployment process, but you can continue with the deployment. **Note:** If you plan on cloning your production instance to one or more non-production instances, you should either create a data preserver for these settings or enable these settings on your production instance. For more information, see Create a clone preserver. For more information, see Enable Automated Test Framework (ATF) properties.
	Configure Automated Test Framework (ATF) suite	For more information, see Exploring Automated Test Framework.
	Configure Instance Scan suite	For more information, see Configuring Instance Scan.
Non-production instances that aren't of type Testing	Configure environment credentials	For more information, see Configure environment credentials.
	Configure the controller instance in your non-production instances	For more information, see Configure properties to integrate Change Management.

Configure environment credentials

Configure credentials in your production instance so that the flows used by the App Engine Management Center (AEMC) can access different instances.

Before you begin

You must use a functional user account with admin permissions to create the credentials for use in your pipelines. This user account should have the following characteristics:

- It should not be associated with a person in your organization.
- It should not need its password reset regularly.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- It should be controlled by your administrator.
- It should have admin privileges on all instances in the pipeline.

You can create a new functional user account specifically for pipeline credentials, or use one that you have previously created with these characteristics. Either way, this user account must be created before you can create the pipeline credentials. For more information about creating this user, see Create a user.

Role required: admin

About this task

Create the pipeline credentials with a functional user account that you created on the Users [sys_users] table. For example, you create a user account called Pipeline Runner and give it admin permissions. This account user name and password are used to set up the credential record.

Procedure

1. Navigate to **All** > **Connections & Credentials** > **Connection & Credential Aliases**.
2. Select **New**.
3. On the form, change the **Type** to **Credential**.
   Changing the type removes several fields from the form.
4. On the form, fill in the remaining fields.

Field	Description
Name	Unique name for the credential record. For example, Pipeline Credentials.
Application	Scope the record is being created in, which should be **Global**.
ID	Unique ID for the credential record. This field populates automatically based on the name. IDs are written in the form [scope_name.credential_name], unless the application is in the Global scope, then the ID is just the credential name.

5. Select **Submit**.
   The Connections & Credentials Aliases table reopens.
6. Open the credential that you created to add credential records to it.
7. On the **Credentials** tab, select **New**.
8. In the **What kind of Credentials would you like to create?** list, select **Basic Auth Credentials**.

   Note: If you want to use OAuth credentials in your pipelines instead, see Configure OAuth credentials for use in Pipelines and Deployments.

9. On the form, fill in the fields.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Name	Unique name for the basic auth credential record. For example, Pipeline Credentials.
Order	Order in which the flows you're using read the credential records. For example, if you have multiple credentials for backups, order them like 100, 200, 300, and so on. Flows read from the top of the list.
User name	User name for the functional account. For example, pipeline.runner.
Password	Password for the functional user account.
Active	Option to make the record active.
Credential alias	Alias for the record, repeated from the credential record ID. The alias exists so that only certain users can see what the user name and password are. When flows execute, they use the credential alias so they can install the apps.

10. Select **Submit**.
11. Repeat this process on each non-production instance in your pipeline.
    For the credential record, use the user name and password that the pipeline should use to access the production instance.

Configure OAuth credentials for use in Pipelines and Deployments

Use OAuth credentials in your pipelines to add another level of security to your pipeline. This task helps you create and configure OAuth credentials for use in your pipelines.

Before you begin

In the top right corner of your instance, make sure you set the application scope to **Global**. Open all your instances (development, test, production, and the like) in separate browser tabs. The following tasks lay out the steps for a three-instance environment. However, if you use any other non-production instances (stage, and the like), there are steps that detail where you may need to repeat a task on that instance.

Role required: admin

About this task

Creating and connecting OAuth credentials consists of several tasks that must be completed in the order laid out. Pay close attention to make sure you're in the correct instance for each step. For more information, see OAuth 2.0 .

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Complete all of the steps under Create OAuth API endpoints for external clients on the specified instances.
2. Complete all of the steps under Create third-party OAuth provider records on the specified instances.
3. Complete all of the steps under Use OAuth to create pipeline credentials on the specified instances.

What to do next

When you Configure your pipeline environments, make sure you select the correct instance credential records for the instance you're configuring.

Create OAuth API endpoints for external clients

Create OAuth API endpoints to enable your controller instance to have two-way communication with your non-production instances. Follow and complete each step carefully on the specified instances before moving on to create your third-party OAuth provider records.

Before you begin

In the top right corner of your instance, make sure you set the application scope to **Global**.

Open all of your instances (development, test, production, and the like) in separate browser tabs.

Role required: admin

About this task

To create OAuth API endpoints for external clients and use OAuth in your pipelines, you must create several records, each on different instances in your pipeline. Begin on your production instance, which should be your controller instance.

Procedure

1. On your production instance, navigate to **All** > **System OAuth** > **Application Registry**.
2. Select **New**.
3. Select **Create an OAuth API endpoint for external clients**.
4. On the form, fill in the fields.

Application Registries form

Field	Action
Name	Enter `Pipeline Controller Client`.
Redirect URLs	**a.** Unlock the field. **b.** Enter the URL for your production, development, and test instances, each with `oauth_redirect.do` after the backslash. **c.** Lock the field. Separate each of the three URLs with a comma and a space. For example: `https://<production instance name>.service-now.com/`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Action
	`oauth_redirect.do, https://<development instance name>.service-now.com/oauth_redirect.do, https://<test instance name>.service-now.com/oauth_redirect.do.`

5. Select **Submit**.

Important: Complete the next steps on your development instance.

6. On your development instance, navigate to **All** > **System OAuth** > **Application Registry**.
7. Select **New**.
8. Select **Create an OAuth API endpoint for external clients**.
9. On the form, fill in the fields.

Application Registries form

Field	Action
Name	Enter `Pipeline Development Client`.
Redirect URLs	**a.** Unlock the field. **b.** Enter the URL for your production and development instances, each with `oauth_redirect.do` after the backslash. **c.** Lock the field. Separate the two URLs with a comma and a space. For example: `https://<production instance name>.service-now.com/oauth_redirect.do, https://<development instance name>.service-now.com/oauth_redirect.do.`

10. Select **Submit**.

Important: Complete the next steps on your test instance.

11. On your test instance, navigate to **All** > **System OAuth** > **Application Registry**.
12. Select **New**.
13. Select **Create an OAuth API endpoint for external clients**.
14. On the form, fill in the fields.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Application Registries form

Field	Action
Name	Enter `Pipeline Test Client`.
Redirect URLs	**a.** Unlock the field. **b.** Enter the URL for your production and test instances, each with `oauth_redirect.do` after the backslash. **c.** Lock the field. Separate the two URLs with a comma and a space. For example: `https://<production instance name>.service-now.com/oauth_redirect.do, https://<test instance name>.service-now.com/oauth_redirect.do`.

15. Select **Submit**.
16. **Optional:** Repeat this process from steps 11–15 for any other non-production instances (staging, and the like) you have.

What to do next

Follow the steps in Create third-party OAuth provider records on the specified instances.

Create third-party OAuth provider records

Create third-party OAuth provider records to enable each of your instances to access the API endpoints you've created.

Before you begin

Complete the tasks in Create OAuth API endpoints for external clients.

In the top right corner of your instance, make sure you set the application scope to **Global**.

Open all of your instances (development, test, production, and the like) in separate browser tabs. Begin the steps below on the production instance.

Role required: admin

Procedure

1. On your production instance, navigate to **All** > **System OAuth** > **Application Registry**.
   You must create three records here, one for each of the three instances (development, test, and production). If you have any additional non-production instances (staging, and the like), create a record for each of those following the method shown here.
2. Select **New**.
3. Select **Connect to a third party OAuth Provider** to create a record for your development instance.
4. On the form, fill in the fields.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Application Registries form

Field	Action
Name	Enter `Dev Instance Connection`.
Client ID	**a.** On your development instance, open the Application Registry list (**All** > **System OAuth** > **Application Registry**). **b.** Open the **Pipeline Development Client** record. **c.** Copy the **Client ID**. **d.** On your production instance, paste the Client ID from your development instance into the **Client ID** field.
Client Secret	**a.** On your development instance, open the Application Registry list (**All** > **System OAuth** > **Application Registry**). **b.** Open the **Pipeline Development Client** record. **c.** Unlock the **Client Secret** field and copy the text. **d.** On your production instance, paste the Client Secret from your development instance in the **Client Secret** field.
Default Grant type	Change to **Authorization code**.
Authorization URL	**a.** Unlock the field. **b.** Enter the URL for your development instance followed by `oauth_auth.do`. **c.** Lock the field. For example: `https://<development instance name>.service-now.com/oauth_auth.do`.
Token URL	**a.** Unlock the field. **b.** Enter the URL for your development instance followed by `oauth_token.do`. **c.** Lock the field. For example: `https://<development instance name>.service-now.com/oauth_token.do`.

5. Select **Submit**.
6. Select **New**.
7. Select **Connect to a third party OAuth Provider** to create a record for your test instance.
8. On the form, fill in the fields.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Application Registries form

Field	Action
Name	Enter `Test Instance Connection`.
Client ID	**a.** On your test instance, open the Application Registry list (**All** > **System OAuth** > **Application Registry**). **b.** Open the **Pipeline Test Client** record. **c.** Copy the **Client ID**. **d.** On your production instance, paste the Client ID from your test instance into the **Client ID** field.
Client Secret	**a.** On your test instance, open the Application Registry list (**All** > **System OAuth** > **Application Registry**). **b.** Open the **Pipeline Test Client** record. **c.** Unlock the **Client Secret** field and copy the text. **d.** On your production instance, paste the Client Secret from your test instance in the **Client Secret** field.
Default Grant type	Change to **Authorization code**.
Authorization URL	**a.** Unlock the field. **b.** Enter the URL for your test instance followed by `oauth_auth.do`. **c.** Lock the field. For example: `https://<test instance name>.service-now.com/oauth_auth.do`.
Token URL	**a.** Unlock the field. **b.** Enter the URL for your test instance followed by `oauth_token.do`. **c.** Lock the field. For example: `https://<test instance name>.service-now.com/oauth_token.do`.

9. Select **Submit**.
10. On the Application Registry list, select the **Pipeline Controller Client** record.
11. Copy the **Client ID** and paste it somewhere like a notes application.
12. Unlock the **Client Secret** field, copy the text, and paste it in a note with the Client ID.
13. Go back to the Application Registry list and select **New**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

14. Select **Connect to a third party OAuth Provider** to create a record for your production instance.
15. On the form, fill in the fields.

Application Registries form

Field	Action
Name	Enter `Prod Instance Connection`.
Client ID	**a.** Copy the **Client ID** that you noted down. **b.** Paste the Client ID into the **Client ID** field.
Client Secret	**a.** Copy the **Client Secret** that you noted down. **b.** Unlock the **Client Secret** field, and paste the text in the field.
Default Grant type	Change to **Authorization code**.
Authorization URL	**a.** Unlock the field. **b.** Enter the URL for your production instance followed by `oauth_auth.do`. **c.** Lock the field. For example: `https://<production instance name>.service-now.com/oauth_auth.do`.
Token URL	**a.** Unlock the field. **b.** Enter the URL for your production instance followed by `oauth_token.do`. **c.** Lock the field. For example: `https://<production instance name>.service-now.com/oauth_token.do`.

16. Select **Submit**.

Important: Complete the next steps on your development instance.

17. On your development instance, navigate to **All** > **System OAuth** > **Application Registry**.
18. Select **New**.
19. Select **Connect to a third party OAuth Provider** to create a record for your production instance.
20. On the form, fill in the fields.

Application Registries form

Field	Action
Name	Enter `Prod Instance Connection`.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Action
Client ID	**a.** Copy the **Client ID** that you noted down. **b.** Paste the Client ID into the **Client ID** field.
Client Secret	**a.** Copy the **Client Secret** that you noted down. **b.** Unlock the **Client Secret** field, and paste the text in the field.
Default Grant type	Change to **Authorization code**.
Authorization URL	**a.** Unlock the field. **b.** Enter the URL for your production instance followed by `oauth_auth.do`. **c.** Lock the field. For example: `https://<production instance name>.service-now.com/oauth_auth.do`.
Token URL	**a.** Unlock the field. **b.** Enter the URL for your production instance followed by `oauth_token.do`. **c.** Lock the field. For example: `https://<production instance name>.service-now.com/oauth_token.do`.

21. Select **Submit**.

> **Important:** Complete the next steps on your test instance.

22. On your test instance, navigate to **All** > **System OAuth** > **Application Registry**.
23. Select **New**.
24. Select **Connect to a third party OAuth Provider** to create a record for your production instance.
25. On the form, fill in the fields.

Application Registries form

Field	Action
Name	Enter `Prod Instance Connection`.
Client ID	**a.** Copy the **Client ID** that you noted down. **b.** Paste the Client ID into the **Client ID** field.
Client Secret	**a.** Copy the **Client Secret** that you noted down. **b.** Unlock the **Client Secret** field, and paste the text in the field.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Action
Default Grant type	Change to **Authorization code**.
Authorization URL	**a.** Unlock the field. **b.** Enter the URL for your production instance followed by `oauth_auth.do` **c.** Lock the field. For example: `https://<production instance name>.service-now.com/oauth_auth.do`.
Token URL	**a.** Unlock the field. **b.** Enter the URL for your production instance followed by `oauth_token.do` **c.** Lock the field. For example: `https://<production instance name>.service-now.com/oauth_token.do`.

26. Select **Submit**.
27. **Optional:** Repeat steps 22–26 for any other non-production instances that you have (staging, and the like).

What to do next

Now that you've completed the pre-work for using OAuth, complete all the steps in Use OAuth to create pipeline credentials on the specified instances.

Use OAuth to create pipeline credentials

Create credential records on each of your instances to enable OAuth use in your pipeline.

Before you begin

Complete the tasks in Create OAuth API endpoints for external clients and Create third-party OAuth provider records.

In the top right corner of your instance, make sure you set the application scope to **Global**.

Open all of your instances (development, test, production, and the like) in separate browser tabs.

Role required: admin

About this task

To configure credentials correctly, you must create records for each of your production and non-production instances on your production instance. Then, you must create a record for your production instance on each of your non-production instances connecting them all together. Use the videos in each section to follow along with the steps. Begin on your production instance.

Procedure

1. On your production instance, navigate to **All** > **Connections & Credentials** > **Connection & Credential Aliases**.
2. Select **New**.
3. Change the **Type** to **Credential**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

4. In the **Name** field, enter `Pipeline Dev OAuth`.
5. Select **Submit**.
6. Reopen the record you just created (Pipeline Dev OAuth).
7. On the Credentials related list, select **New**.
8. Select **OAuth 2.0 Credentials**.
9. On the form, fill in the fields.

OAuth 2.0 Credentials form

Field	Action
Name	Enter `Dev OAuth`.
OAuth Entity Profile	Search for and select the **Dev Instance Connection default profile**.

10. Select **Submit**.
11. Reopen the **Dev OAuth** credential record.
    An expected error appears, directing you to verify the OAuth configuration and select **Get OAuth Token** to request a new token.
12. Under Related Links, select **Get OAuth Token**.
    A development instance opens requesting to connect to your ServiceNow account.
13. Select **Allow**.
14. Go back to the Connection & Credential Aliases list (**All** > **Connections & Credentials** > **Connection & Credential Aliases**).
15. Select **New**.
16. Change the **Type** to **Credential**.
17. In the **Name** field, enter `Pipeline Test OAuth`.
18. Select **Submit**.
19. Reopen the record you just created (Pipeline Test OAuth).
20. On the Credentials related list, select **New**.
21. Select **OAuth 2.0 Credentials**.
22. On the form, fill in the fields.

OAuth 2.0 Credentials form

Field	Action
Name	Enter `Test OAuth`.
OAuth Entity Profile	Search for and select the **Test Instance Connection default profile**.

23. Select **Submit**.
24. Reopen the Test OAuth credential record.
    An expected error appears, directing you to verify the OAuth configuration and select **Get OAuth Token** to request a new token.
25. Under Related Links, select **Get OAuth Token**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

A test instance opens requesting to connect to your ServiceNow account.

26. Select **Allow**.
27. Go back to the Connection & Credential Aliases list (**All** > **Connections & Credentials** > **Connection & Credential Aliases**).
28. Select **New**.
29. Change the **Type** to **Credential**.
30. In the **Name** field, enter `Pipeline Prod OAuth`.
31. Select **Submit**.
32. Reopen the record you just created (Pipeline Prod OAuth).
33. On the Credentials related list, select **New**.
34. Select **OAuth 2.0 Credentials**.
35. On the form, fill in the fields.

OAuth 2.0 Credentials form

Field	Action
Name	Enter `Prod OAuth`.
OAuth Entity Profile	Search for and select the **Prod Instance Connection default profile**.

36. Select **Submit**.
37. Reopen the Prod OAuth credential record.
    An expected error appears, directing you to verify the OAuth configuration and select **Get OAuth Token** to request a new token.
38. Under Related Links, select **Get OAuth Token**.
    A production instance opens requesting to connect to your ServiceNow account.
39. Select **Allow**.
40. **Optional:** If you have any other non-production instances (staging, etc.), create a credential record for each of them on your production instance using the methods above.

> **Important:** After you've completed the steps above on your production instance, follow the steps below to create one credential record on each of your non-production instances connecting them to your production instance. Complete the next steps on your development instance.

41. On your development instance, navigate to **All** > **Connections & Credentials** > **Connection & Credential Aliases**.
42. Select **New** to create a record to connect your development instance to production.
43. Change the **Type** to **Credential**.
44. In the **Name** field, enter `Pipeline Prod OAuth`.
45. Select **Submit**.
46. Reopen the record you just created (Pipeline Prod OAuth).
47. On the Credentials related list, select **New**.
48. Select **OAuth 2.0 Credentials**.
49. On the form, fill in the fields.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

OAuth 2.0 Credentials form

Field	Action
Name	Enter `Prod OAuth`.
OAuth Entity Profile	Search for and select the **Prod Instance Connection default profile**.

50. Select **Submit**.
51. Reopen the Prod OAuth credential record.
    An expected error appears, directing you to verify the OAuth configuration and select **Get OAuth Token** to request a new token.
52. Under Related Links, select **Get OAuth Token**.
    A production instance opens requesting to connect to your ServiceNow account.
53. Select **Allow**.

> **Important:** Complete the next steps on your test instance.

54. On your test instance, navigate to **All** > **Connections & Credentials** > **Connection & Credential Aliases**.
55. Select **New** to create a record to connect your test instance to production.
56. Change the **Type** to **Credential**.
57. In the **Name** field, enter `Pipeline Prod OAuth`.
58. Select **Submit**.
59. Reopen the record you just created (Pipeline Prod OAuth).
60. On the Credentials related list, select **New**.
61. Select **OAuth 2.0 Credentials**.
62. On the form, fill in the fields.

OAuth 2.0 Credentials form

Field	Action
Name	Enter `Prod OAuth`.
OAuth Entity Profile	Search for and select the **Prod Instance Connection default profile**.

63. Select **Submit**.
64. Reopen the Prod OAuth credential record.
    An expected error appears, directing you to verify the OAuth configuration and select **Get OAuth Token** to request a new token.
65. Under Related Links, select **Get OAuth Token**.
    A production instance opens requesting to connect to your ServiceNow account.
66. Select **Allow**.
67. **Optional:** If you have any other non-production instances (staging, etc.), create a credential record on that instance for production using steps 54-66.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

What to do next

Now that you've created all of the credential records connecting your instances, you can use those records to configure your pipeline environments. For more information, see Configure your pipeline environments.

Configure your pipeline environments

Set up your app development production and non-production environments (for example, development, test, and/or staging) by adding the URLs and credentials used to access each instance.

Before you begin

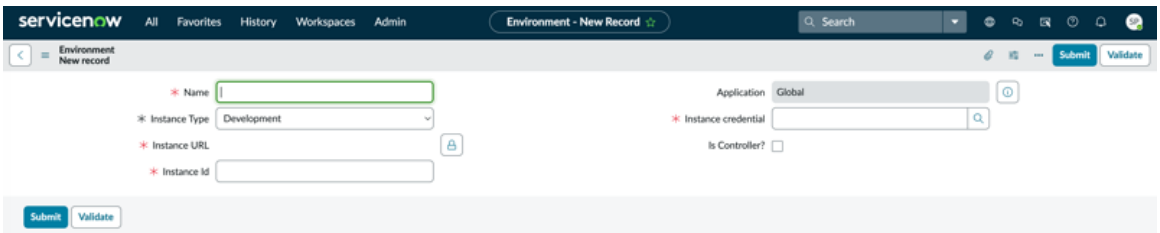
Role required: admin

Note: Only users with the System Administrator (admin) role can define instance credentials for environments. Users with the App Engine Administrator [sn_app_eng_notify.app_engine_admin] role can view environment records; however, the **Instance credential** field is not visible.

Procedure

1. In your production instance, navigate to **All** > **App Engine** > **Pipelines and Deployments** > **Environments**.
2. Select **New**.

Environment - new record



3. On the form, fill in the fields.

Environment form

Field	Description
Name	Name of the environment. Enter a name that distinguishes the instance as a development, test, or production environment. For example, if you're defining a development environment, you could include `Dev` in the name.
Instance Type	The type of environment you are defining: ○ Sandbox ○ Development ○ Testing ○ Staging ○ Production
Instance URL	Web address of your ServiceNow instance.
Instance ID	The instance_id sys_property for the instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
	Note: The Instance ID is automatically generated when you complete all of the other fields and select **Validate**.
Application	Application scope of the environment.
Instance credential	Authentication data related to the instance that you're configuring as an environment. **Note:** If you are using OAuth 2.0 credentials, select the appropriate credential for the instance record. For example, on your development instance record, select the OAuth Dev credential, and for your test instance record, select the OAuth Test credential.
Is Controller?	Identifies if this instance is a controller. This should be selected for the production environment record where you plan to manage deployment requests. For more information, see Configure your controller instance.

4. Select **Submit**.
5. Repeat the previous steps as you define each remaining environment in your pipeline. For example, if you defined a production environment, repeat the procedure again to define a development environment. Then repeat the procedure once more to define a test environment.

Configure your pipeline

Configure your app development pipeline so that your administrator can quickly move an application from one environment to another.

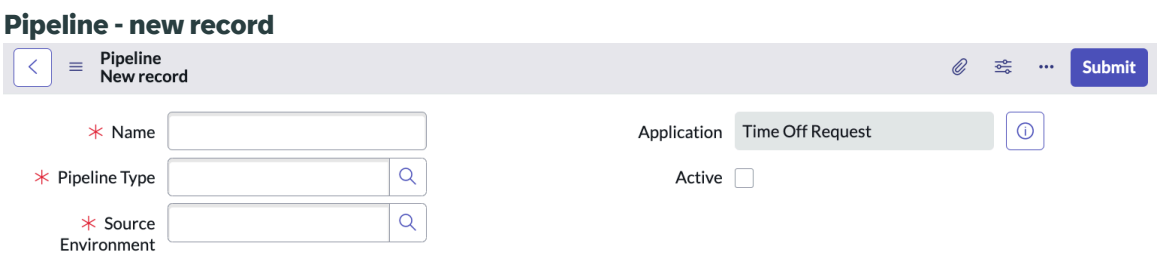
Before you begin

You must create all of your pipeline environment records before completing these steps. For more information, see Configure your pipeline environments.

Role required: admin or app_engine_admin

Procedure

1. In your production instance, navigate to **All** > **App Engine** > **Pipelines and Deployments** > **Pipelines**.
2. Select **New**.



3. On the form, fill in the fields.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Pipeline form

Field	Description
Name	Name of the pipeline. Enter a name that distinguishes this pipeline from other pipelines.
Pipeline Type	The type of pipeline you are defining. The most common use case for pipelines is to select Application Deployment; however, you can define other types as needed.
Source Environment	The environment for this pipeline, usually the development environment.
Application	Application scope of the pipeline.
Active	Select the check box to activate this pipeline.

4. Select **Submit**.
5. Reopen the pipeline record you just created.
6. In the Pipeline Environments Order section, select the environments in the pipeline, excluding the **Source Environment**, and specify the order in which apps should be deployed through the pipeline.
7. Select **Update**.

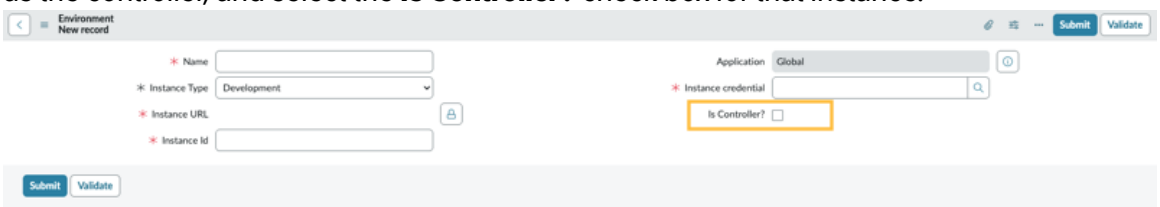
Configure your controller instance

You must identify one of the instances in your app development pipeline as the controller instance. All communication between the instances in your pipeline, including the deployment order for applications in the pipeline, takes place in the controller instance.

Before you begin
Role required: admin

About this task
Typically, your production instance should be identified as the controller instance. All request and approval records are stored on your controller instance.

Procedure
1. Navigate to **All** > **App Engine** > **Pipelines and Deployments** > **Environments**.
2. When you are configuring your pipeline environments, decide which instance you want to use as the controller, and select the **Is Controller?** check box for that instance.



3. Select **Save**.
   When the workflow for the pipeline runs, the non-production instances communicate with the controller instance to determine to which instance the app should next be deployed.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add ATF and instance scan suites for testing

Add your customized test suites as part of an application deployment in addition to the instance scan and Automated Test Framework (ATF) testing suites that are standard in AEMC.

Before you begin

Because the ATF and instance scan suites exist only on testing instances, you must have your pipeline set up to complete this task. For more information, see Configure your pipeline environments.

Role required: admin

Procedure

1. On your production instance, navigate to **All** > **App Engine** > **Pipelines and Deployments** > **Scan Suites**.
2. If this is the first time you are looking at this table, select **Populate default suites** to load the instance and ATF scan suites that ServiceNow provides.
3. Add a new scan suite by selecting **New**.
4. On the form, fill in the fields.

Scan Suite form

Field	Description
Suite	Table with all the ATF and Instance scan suites that exists on your testing instance.
Error severity	Error options that indicate the severity of a scan suite not passing testing.
Instance Type	Type of instance the scan suite will run on. This field should be set to **Testing**.

5. Select **Submit**.
   The suites you add will run on apps installed on the instance you selected during testing.

Enable Automated Test Framework (ATF) properties

Pipelines and Deployments includes an Application Test Framework (ATF) suite called the Application Deployment Test Suite. Two system properties control whether the test runs automatically whenever an app is deployed to a Test environment.

Before you begin

Role required: admin

About this task

The tests in the Application Deployment Test Suite can be run on production instances; however, the flows included in the base system run only on non-production (Test) instances. Additionally, the tests should run only on an instance defined with an **Instance Type** of **Testing**. For more information, see Configure your pipeline environments.

Note: If you plan on cloning your production instance to one or more non-production instances, you should either create a data preserver for these settings or enable these settings on your production instance. For more information, see Create a clone preserver.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to the System Properties table by typing `sys_properties.list` in the navigation filter and pressing **Enter**.
2. Locate the following two properties:
   - sn_atf.runner.enabled
   - sn_atf.schedule.enabled
3. If you want these tests to run automatically on non-production instances whenever an app is deployed to a Test instance, open each property record, change the **Value** field to true, and select **Update**.

Enable Change Management integration

Enable Change Management integration with AEMC so that deployments through the pipeline to the production environment are automatically scheduled based on the Change request state and planned change window.

Before you begin
Role required: admin

About this task
This task can be completed within the Guided Setup system. This task opens within Guided Setup when you select **Configure**. If you prefer, you can follow the instructions below to access the same configuration task outside of the Guided Setup.

Procedure

1. Navigate to the System Properties table by typing `sys_properties.list` in the navigation filter and pressing **Enter**.
2. Locate the following property: sn_deploy_pipeline.change_management.enabled.
3. Connect your existing Change Management processes to AEMC by changing the **Value** field to `true`.
4. Select **Update**.
5. In the Pipelines and Deployments Guided Setup, select **Mark Complete**.
   Marking the enablement step complete enables you to configure properties related to Change Management integration including the change model, change template, and CI creation subflow. For more information, see Configure properties to integrate Change Management.

Configure properties to integrate Change Management

Configure predefined properties through the Pipelines and Deployments Guided Setup to tailor your Change Management and AEMC integration to your company's needs. Configuring these properties is optional.

Configure the change model

Configure the change model you would like to use during application deployments. If not configured, the **Normal** change model is used by default.

Before you begin
Role required: admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

About this task

There are several change models you can use to tailor change activities and flows for specific use cases. For more information about each type of model, see . Use the steps below to select which change model you would like to use during app deployments.

This task can be completed within the Guided Setup system. This task opens within Guided Setup when you select **Configure**. If you prefer, you can follow the instructions below to access the same configuration task outside of the Guided Setup.

Procedure

1. Navigate to the System Properties table by typing `sys_properties.list` in the navigation filter and pressing **Enter**.
2. Locate the following property: sn_deploy_pipeline.change_management.default.model.
   To change the value of this property, and therefore the model that's being used, you must identify the **sys_id** of the change model you want to switch to.
   a. Navigate to **All** > **Change** > **Administration** > **Change Models**.
   b. Select the Change model you want to make the default for app deployments (for example, **Standard**).
   c. Right-click in the header bar for the record, and select **Copy sys_id**.
3. In the **sn_deploy_pipeline.change_management.default.model** property, paste the sys_id you have copied into the **Value** field, replacing its current contents.
4. Select **Update**.
5. In the Pipelines and Deployments Guided Setup, in the **Configure Change Model** section, select **Mark Complete**.

Configure the change template

Configure the default change template used to create change requests during application deployments.

Before you begin

Role required: admin

About this task

There are several templates that you can choose from to use as the default template for creating change requests. For more information, see Create a change request template. Use the following steps to select which change template you would like to use during app deployments.

This task can be completed within the Guided Setup system. This task opens within Guided Setup when you select **Configure**. If you prefer, you can follow the instructions below to access the same configuration task outside of the Guided Setup.

Procedure

1. Navigate to the System Properties table by typing `sys_properties.list` in the navigation filter and pressing **Enter**.
2. Locate the following property: sn_deploy_pipeline.change_management.default.template.
   To change the value of this property, and therefore the default template used for change requests, you must identify the **sys_id** of the referenced template in the change template that you want to switch to.
   a. Navigate to **All** > **Change** > **Standard Change** > **All Templates**.
   b. Select the update personalized list icon ( ) from the list header.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

c. In the Available column, select **Template** and use the move icon () to add it to the Selected column.

d. Select **OK**.

e. Select the value in the **Template** field of the change template that you want to make the default for app deployments.

f. Right-click in the header bar for the template record and select **Copy sys_id**.

3. In the **sn_deploy_pipeline.change_management.default.template** property, paste the sys_id that you have copied into the **Value** field, replacing its current contents.
4. Select **Update**.
5. In the Pipelines and Deployments Guided Setup, in the **Configure Change Template** section, select **Mark Complete**.

Configure CI creation subflow

Customize how Configuration Items (CIs) are created during application deployment using AEMC.

Before you begin

Role required: admin

About this task

By default, the **Create CMDB CI if not present** subflow will run automatically during deployment to create a CI if for your application if it doesn't exist. However, you can change how Configuration Items (CIs) are created during deployment by using the following steps.

Procedure

1. Navigate to the System Properties table by typing `sys_properties.list` in the navigation filter and pressing **Enter**.
2. Locate the following property:
   sn_deploy_pipeline.change_management.config_ci_creation_subflow.
   To change the value of this property, and therefore the template that's being used, you must identify the **sys_id** of the subflow you want to switch to.

   a. Navigate to **All** > **Flow Designer**.

   b. Select the update personalized list icon  from the list header.

   c. In the Available column, select **Template** and use the move icon  to add it to the Selected column.

   d. Select the More Actions menu icon , and select **Copy subflow**.

   e. Enter a name for the subflow, and make sure the application is in the **Deployment Pipeline** scope.

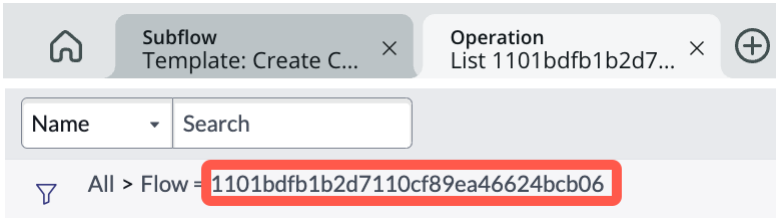
   f. Select **Copy**.

   g. Modify the subflow, and select **Save**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

h. To find the sys_id of the subflow, select the More Actions menu icon ⋮, and select **Executions**.

i. Copy the sys_id.
The sys_id is listed at the top of the page, after **All> Flow=**.



3. In the **sn_deploy_pipeline.change_management.config_ci_creation_subflow** property, paste the sys_id you have copied into the **Value** field, replacing its current contents.
4. Select **Update**.
5. In the Pipelines and Deployments Guided Setup, in the **Configure CI Creation Subflow** section, select **Mark Complete**.

Migration tasks

You must complete several tasks to migrate your App Engine pipelines to ReleaseOps.

Starting with version 28.2.1 of AEMC, you can migrate your existing pipelines to ReleaseOps. To learn more about the migration process and the advantages of migrating to ReleaseOps, see Migrating App Engine pipelines to ReleaseOps. To start migrating your existing pipelines, complete each of the following procedures.

1. Install ReleaseOps on each environment (instance) in your App Engine pipeline.
2. Complete ReleaseOps setup by configuring a new ReleaseOps ecosystem and configuring multi-instance management.
3. Configure the system property to enable migration to ReleaseOps.
4. Map deployment request states to match ReleaseOps deployment request states to the corresponding App Engine deployment request states.
5. Add custom conditions to enable ReleaseOps deployments in Workflow Studio.

Configure the system property to enable migration to ReleaseOps

Update the *sn_deploy_pipeline.migrate_releaseops* system property to enable deployments using ReleaseOps.

Before you begin

Role required: admin or app_engine_admin

You must have ReleaseOps installed and set up on your production and other pipeline environments to complete the following procedure. See Migrating App Engine pipelines to ReleaseOps for more information.

Procedure

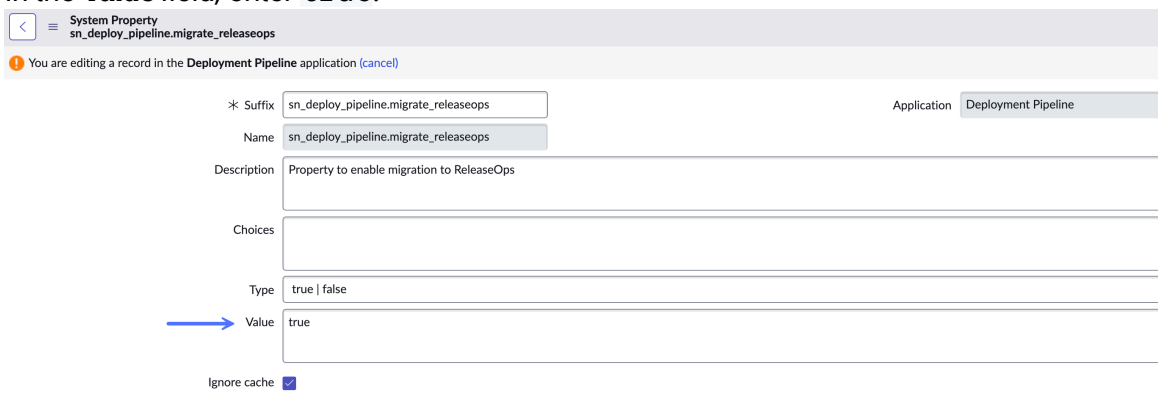
1. On your production instance, navigate to **All** > **System Properties** > **All Properties**.
2. In the search bar, enter `sn_deploy_pipeline.migrate_releaseops` and select the Enter key.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

3. Select the *sn_deploy_pipeline.migrate_releaseops* property from the list.

   Note: By default, the *sn_deploy_pipeline.migrate_releaseops* property value is set to false.

4. **Optional:** If you aren't in the correct application scope, select the link to open the record in the correct scope.
5. In the **Value** field, enter `true`.



6. Select **Update** or select and hold (or right-click) in the header and select **Save** to update the record.

Result

You have set the *sn_deploy_pipeline.migrate_releaseops* system property to enable migration to ReleaseOps.

What to do next

If you customized your ReleaseOps deployment request states, you must map the deployment request states onto the corresponding App Engine deployment request states. See Map deployment request states for more information.

Map deployment request states

If your ReleaseOps deployment request has custom states, you must configure the Deployment Request State Map table so that your ReleaseOps deployment request states map to the correct App Engine states.

Before you begin

Role required: admin or app_engine_admin

Procedure

1. On your production instance, navigate to **All** > **App Engine** > **Pipelines and Deployments** > **ReleaseOps State Mapping**.
2. Select the ReleaseOps deployment request state in the **ReleaseOps DR state** column that you want to update.

   Important: It's not recommended to change the mapping of any ReleaseOps deployment request end states, including:

   - Canceled
   - Completed
   - Failed

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

3. On the state mapping record page, select the value in the **App Engine DR state** field that corresponds to the ReleaseOps deployment request state.
4. **Optional:** In the **Notes** field, enter additional details about the state mapping.
5. Select **Update** or select and hold (or right-click) in the header and select **Save** to update the record.
6. **Optional:** Repeat the procedure in steps 2 through 5 for each deployment request state that you want to update.
7. To create a deployment request state mapping record, select **New** on the Deployment Request State Maps table.
   You can create a deployment request state mapping record when there are more or new ReleaseOps deployment request states that must be mapped onto corresponding App Engine deployment request states.
   a. In the **App Engine DR state** field, select the App Engine deployment request state from the list.
   b. In the **ReleaseOps DR state** field, select the corresponding ReleaseOps deployment request state.
   c. **Optional:** In the **Notes** field, enter additional details about the state mapping.
   d. Select **Update** or select and hold (or right-click) in the header and select **Save** to update the record.

Result

You have mapped all ReleaseOps deployment request states onto the corresponding App Engine deployment request states.

What to do next

If you want to configure additional conditions to enable migration to ReleaseOps, see Add custom conditions to enable ReleaseOps deployments.

Add custom conditions to enable ReleaseOps deployments

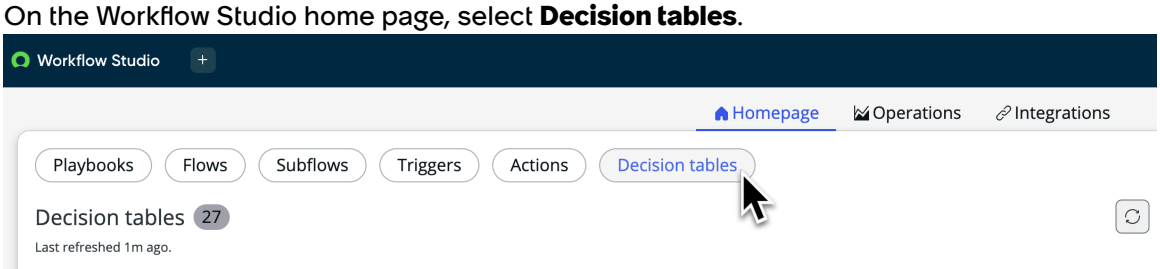
If you want to add additional conditions to enable *ReleaseOps* deployments, modify the Deployment Migration to *ReleaseOps* decision table.

Before you begin

Role required: admin or app_engine_admin

Procedure

1. On your production instance, navigate to **All** > **Process Automation** > **Workflow Studio**.
2. On the Workflow Studio home page, select **Decision tables**.



3. Select the decision table with the name **Deployment Migration to** *ReleaseOps*.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

4. Add conditions and results to the decision table to set up additional conditions for *ReleaseOps* deployments.

   Tip: For more information about how to use decision tables, see the following resources:

   - Using decision tables
   - Modify decision table rules in Workflow Studio

Configure a standalone environment

With App Engine Management Center (AEMC), you no longer need an active deployment pipeline to get started. Complete the standalone environment setup to start deploying changes quickly. You can add a pipeline later when you're ready to automate deployments.

Before you begin

Role required: admin

Related tasks:

- Configure environment credentials
- Configure your pipeline environments

About this task

The standalone configuration option enables you to set up AEMC without having to configure an active deployment pipeline. Configure your development environment first, then production, and optionally your testing environment. You can add a pipeline later when you're ready to automate deployments.

For general information about guided setup, see Using guided setup.

Procedure

1. Navigate to **All** > **App Engine** > **Pipelines and Deployments** > **Guided Setup**.

   Note: The landing page provides information about the different categories, tools, and user access.

2. Select **Get Started** (or **Continue** if you have previously started guided setup).

   Note: The Pipelines and Deployments Guided Setup category page appears, displaying a list of setup tasks grouped by environment. For standalone installation, you will complete environment setup tasks only and skip pipeline creation entirely. You can add a pipeline later if needed.

3. In the development environments category, select **Get Started** and complete setting up the controller instance.

   For more information, see Configure your controller instance.

   Note: When you have completed this environment setup task, the Guided Setup screen reappears.

4. If you have a testing environment, select **Get Started** in the testing environment category and configure your controller instance.

   Note: When you have completed this task, the Guided Setup screen reappears.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

5. In the production environment category, select **Get Started** and add users to the App Engine admin group.

   For more information, see Add users to the App Engine admin group.

   Note: When you have completed this environment setup task, the Guided Setup screen reappears. Standalone AEMC environment setup is complete.

Result
Your AEMC standalone environment is set up and ready to use. When you're ready to automate your deployments, see Configure your pipeline.

Using AEMC

AEMC enables you to manage app development and deployments, all in one centralized location. Learn about how you can use AEMC to govern your organization's app development.

Aims of AEMC

The App Engine Management Center (AEMC) is designed to empower App Engine admins with all of the tools needed to make governing app development easier. AEMC provides a unified experience for managing all stages of application development, from idea submission to deployment. Using AEMC, admins can manage requests for new apps and delegate development and collaboration permissions. Admins can also oversee the entire deployment process with insights into individual deployment requests, test suites, and scheduled deployments. The following list outlines the tasks that the AEMC can help you to accomplish:

- Mange intake requests submitted by developers for new apps.
- Manage app requests submitted by no-code developers from within Creator Studio.
- Manage collaboration requests submitted by developers working on new apps.
- Manage deployment requests for apps that are ready to be moved through the pipeline.
- Schedule app deployments.
- View insights about developers and custom apps within their ServiceNow environment.

Managing app development using the App Engine Management Center

Track and manage requests for new applications, collaboration requests, and insights into developer contributions and custom app production across your organization. You can also assign Developer Sandboxes licenses to managed instances.

Use AEMC from your production instance to manage app development across your organization, including:

- Intake requests
- App requests
- Collaboration requests
- Developer Sandboxes licenses
- Custom app production and usage
- Developer contributions and insights

Managing requests using AEMC

You can track and approve or reject app intake and collaboration using the App Engine Management Center (AEMC).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

To work more efficiently, you can also filter and search for requests.

Manage intake requests

View details of intake requests submitted by developers in App Engine Studio in the App Engine Management Center (AEMC) dashboard, and approve or reject them.

Before you begin

Before intake requests can be displayed in AEMC, you must confirm that the Application Intake application has been properly configured. For more information, see Configure Application Intake. For more information about how to submit intake requests, see Submit your idea for app development.

App requests submitted from Creator Studio restricted users are managed on the App tab on the Requests page. For more information, see Manage app requests from Creator Studio.

Role required: sn_app_eng_notify.app_engine_admin

Procedure

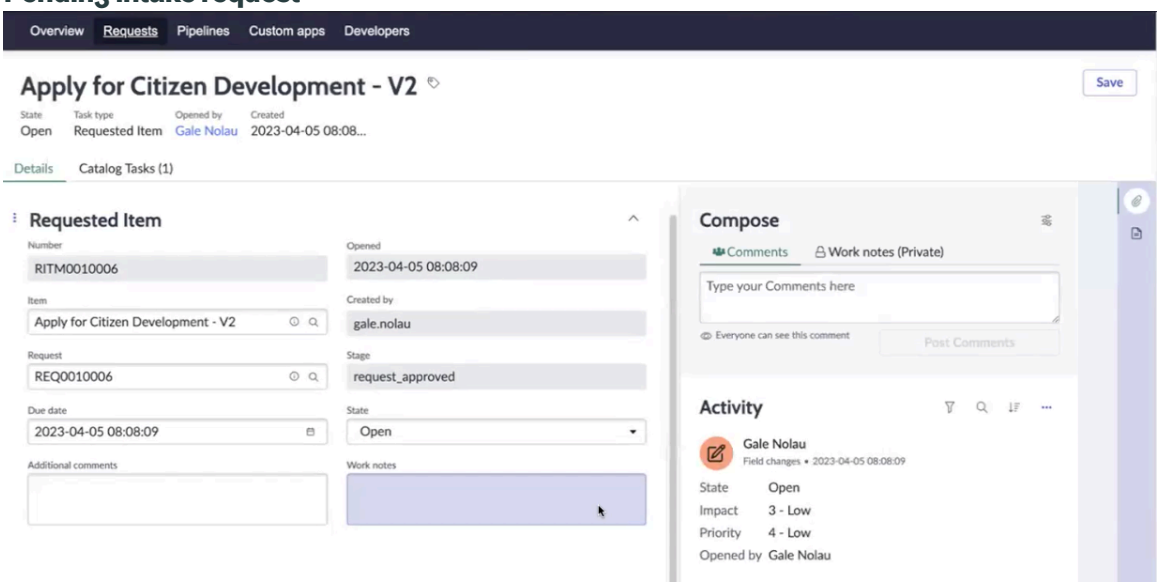
1. Navigate to **All** > **App Engine** > **Administration** > **App Engine Management Center**.

   The AEMC dashboard shows the number of pending intake requests.

   Note: To view a list of all intake requests, either select the number of requests or select the Requests page.

2. Select an intake request to review its details.

Pending intake request



3. Return to the **Details** tab and review the request.

   Note: The **Administration** section includes a **Development instance** field. Make sure that the development instance the developer intends to use has been selected. Also, if the user isn't defined on that instance, a message displays and you must manually add the user to that instance. For more information, see Create a user.

4. Use the **Permission type** list in the Administration section to select one of the following options for provisioning groups to this application.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Permission type	Description
Select group(s)	Option to give one or more of the following groups permission to develop this application. ○ App Engine Studio User Limited ○ App Engine Studio Users ○ Creator Studio Restricted Users ○ Creator Studio Users For more information about how to control which groups show up in this list, see Manage user groups for Application Intake.
I will manually provision user	Option to add groups to the application manually.
Do not give requestor permissions	Option not to add any groups to the application at the time of the request.

5. Approve or reject the request.
   - If the request is approved, the user is added to the App Engine Studio User group and granted the App Engine Studio User role. The AES User role enables developers to begin creating their application in AES in the approved instance. The user also receives an email notification with a link to the provisioned instance. For more information about the AES User role, see Components installed with App Engine Studio.
   - If the request is rejected, a rejection email is sent to the user. The user can submit new requests. However, the rejected request can't be edited.

Manage app requests from Creator Studio

View and manage new app requests in App Engine Management Center (AEMC) that users with the Creator Studio Restricted role (sn_creator_studio_restricted_user) submit for apps they want to create.

Before you begin

Role required: sn_app_eng_notify.app_engine_admin

Procedure

1. Navigate to **All** > **App Engine** > **Administration** > **App Engine Management Center**.
2. View all app requests on the **Requests** page by selecting the **App** tab.
3. Select an app request to view its details.
4. Either approve or reject the request.
   - If the app is useful to your organization, approve the request.
   - If you see issues with the app request, reject the request. If an app request is rejected, the original requestor must put in a new request for the app with changes for reconsideration.

Manage collaboration requests

As an admin, view and approve or reject collaboration requests in the App Engine Management Center (AEMC) from developers who want assistance building their applications.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

To view collaboration requests in AEMC, you must first set up Pipelines and Deployments in your production instance. Otherwise, requests are created on the instance where they originated (that is, the instance with App Engine Studio (AES) installed).

Role required: sn_app_eng_notify.app_engine_admin

About this task

Developers with collaboration permissions can access and develop certain parts of an application. Requests for collaboration may come from App Engine Studio or Creator Studio.

Requests from collaborators may be approved automatically depending on permissions or roles they already have.

- For AES collaboration requests, users who have the AES User role are approved automatically.
- Creator Studio users who have already created an app (or have had one created for them) have delegated development permissions, and are automatically approved to collaborate on other apps.
- Current delegated developers without the Creator Studio User or Creator Studio Restricted User roles are automatically approved to collaborate in Creator Studio.

For more information about managing collaboration and permissions, see Collaborate with other developers.

Procedure

1. Navigate to **All** > **App Engine** > **Administration** > **App Engine Management Center**.

   The AEMC dashboard shows the number of pending collaboration requests.

   Note: To view a list of all collaboration requests, either select the number of requests or select the Requests page.

2. Select a collaboration request to review its details.
3. Return to the **Details** tab and review the request.
4. Approve or reject the request.
   - If the request is approved, the user invited to be a collaborator is added as a developer to the specified app. The collaborator can now work on the parts of the app defined in the Collaboration Descriptor, but cannot concurrently develop other apps in AES. For more information about the AES User role, see Components installed with App Engine Studio.
   - If the request is rejected, the request is closed, and the primary developer is notified by email.

Filter and search for requests in AEMC

The App Engine Management Center (AEMC) application provides tools for locating requests. You can perform a global search for request records or filter the list of current requests to locate the ones you want to work on.

Filtering requests

On the Requests, Pipelines, Release Management, Custom apps, and Developers pages, you can filter the list of requests by several different criteria.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

To create a filter, select the filter icon ( ), select **Advanced view**, and add criteria. Build a filter by adding conditions that contain a field, operator, and value(s). You can use an existing filter, save what you create, and limit who can use the filter. Build more complex filters by using the **or** and **and** connectors. Add as many filters as you need, and select **Update**.

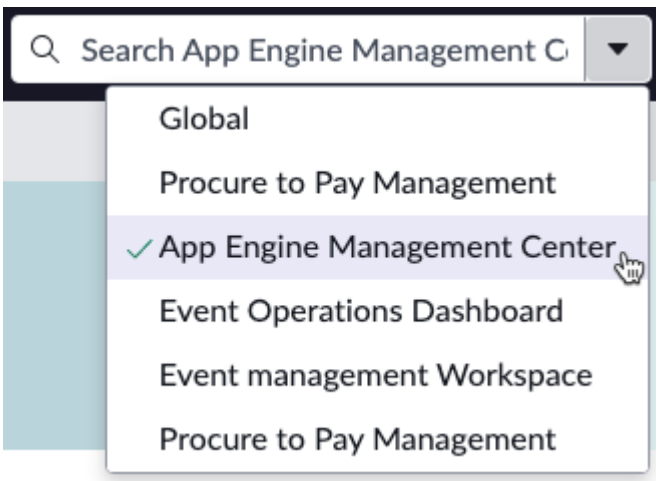
Note: On the **Requests** and **Release Management** tabs, you can use the **Saved filters** option (Filter) which provides default filtering capabilities along with advanced features and real time filtering options. Saved filters can be stored and shared with individual users or user groups..

When you filter on the Pipelines tab, only instances with requests that match the filtering criteria display information. For example, if your pipeline shows request data in three instances and you filter by an app name used in only one of those instances, only that instance returns results.

Searching for requests

You can use the **Search** field at the top of any AEMC screen to search for submitted requests. Simply type all or a portion of a request number using an asterisk wild card character, and select **View results**.

Search



For example, you can type DEV0001* and select **View results** (or simply press **Enter**), and a list of all collaboration requests that begin with DEV0001 are returned.

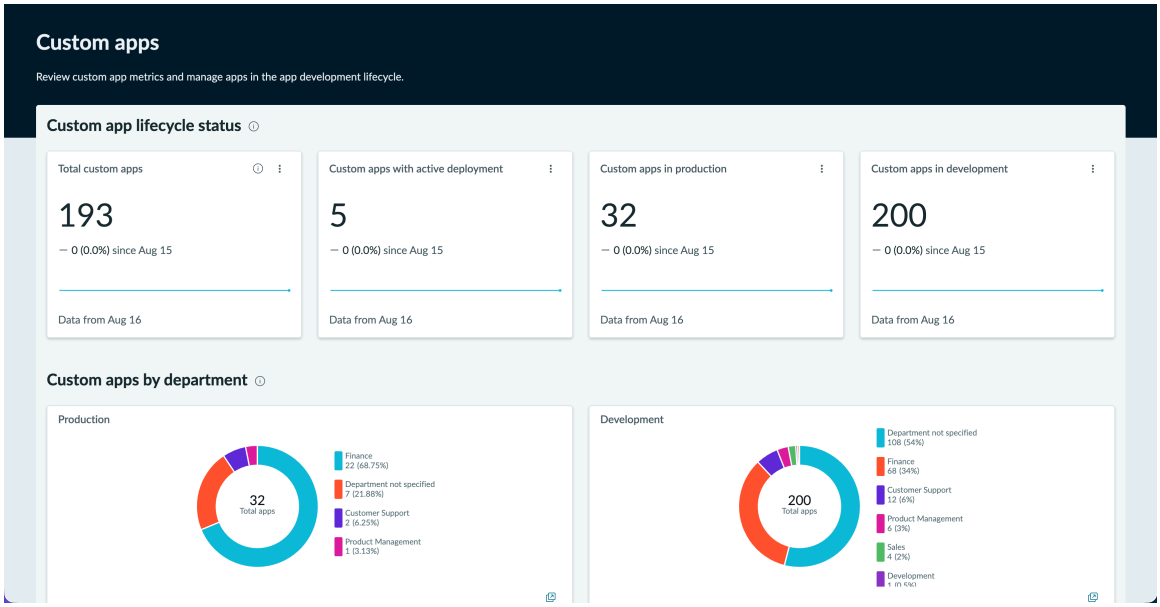
You can then select the record you want to review.

Manage custom apps using AEMC

Review custom app metrics and manage apps through the development life cycle using the App Engine Management Center (AEMC).

The Custom apps page in AEMC shows detailed metrics and a full list of your applications. In each tile in the Custom app lifecycle status section, trending data is available for the last 90 days. You can show all apps in the Custom apps list or limit the list to only certain types of applications using the filter. Select a trend chart to filter the list to those criteria.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



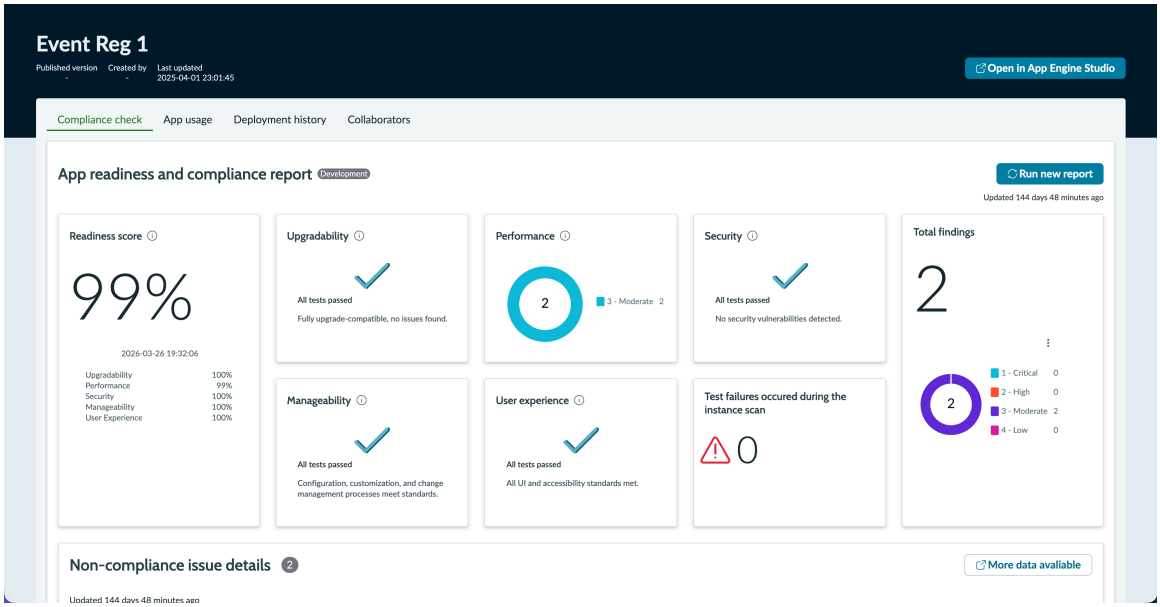
The Custom app page also lists all the applications individually, with a single record for each application. If the app has been published to production, the **Published Version** column shows the version number. If the **Published Version** column is blank, the app exists in development but has never been published to production.



Name	Published Version	Creator user name	Created on	Last updated on
Event Reg 1			2025-04-01 16:01:45	2025-04-01 16:01:45
Time Tracking App 1			2025-04-01 15:59:14	2025-04-01 15:59:14
Donut Requests			2025-03-04 11:56:29	2025-03-04 11:56:29
Requests app			2025-02-24 13:14:39	2025-02-24 13:14:40
New Exp2			2025-02-06 12:08:59	2025-02-06 12:08:59
Z expense app			2025-01-31 07:44:34	2025-01-31 07:44:34
Legal Service Delivery			2024-12-20 10:06:05	2024-12-20 10:06:05
test demo222			2024-12-19 23:11:50	2024-12-19 23:11:50
fdfdf			2024-12-19 13:35:38	2024-12-19 13:35:39
New app flow			2024-12-19 11:20:21	2024-12-19 11:20:21
TEst dfdfdsfsdf			2024-12-19 11:17:28	2024-12-19 11:17:28
Test for today			2024-12-18 16:03:21	2024-12-18 16:03:21
CS Deployment App			2024-12-18 07:32:39	2025-03-04 11:57:24
Test Instance App 2			2024-12-18 06:55:21	2024-12-18 06:55:43
Test Instance App			2024-12-18 06:53:37	2024-12-18 06:53:51
Amanda's Awesome STEP presentation from last week			2024-12-16 12:41:38	2024-12-16 12:41:38
Zahid's App			2024-12-06 09:58:03	2024-12-06 09:58:03

View the app's readiness and compliance report, usage data, subscription monitoring details, deployment history, and collaborators, if any, by selecting an application name from the list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Select **Open in App Engine Studio** to see the contents of the app and access more information. The App Engine Studio application opens in a new browser window.

App readiness and compliance report

See how ready the app is for deployment to production instances in the **Compliance check** tab. To learn more about the app readiness and compliance report, see App Readiness and Compliance Report.

App usage

See a high-level view of the usage data for apps on production instances by selecting the **App usage** tab. The data includes the user count, insert count, and update count aggregated for the current month.

- User count: The number of users who have accessed the application.
- Insert count: The number of new records on tables in the application.
- Update count: The number of records in the application that have been updated.

Usage data for your applications provides several benefits:

- Discover whether your app is being used. Are people using the app? Are they creating records? If not, consider updating or deleting the app.
- Investigate and fix problems with the app. Knowing when or how data was changed can help identify where something went wrong.
- Determine whether you must load balance the processing power of your instance by seeing when there might be consistently higher traffic in your app.

In the Usage Insights dashboards section, see detailed data such as active users, sessions, and page views by selecting an experience. For more information about how to understand and use this data, see Usage Insights.

Subscription monitoring

Understand the roles used in your custom application and their implications for licensing and subscriptions. The data in this section includes details about each role and the users assigned to each role.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Subscription and licensing information for each application provides several benefits:

- Discover what roles are active in your custom application.
- Discover which users are assigned each role and the permission levels they have.
- See the cost implications of each license type and where it's used in your apps.

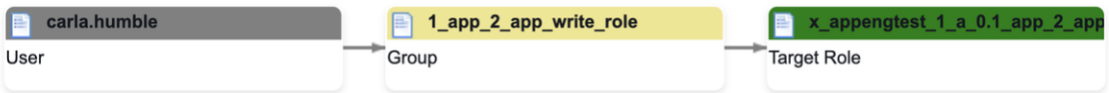
System administrators can see which subscription the app is mapped to and have an option to change the subscription. Several factors affect what you see in this section.

- If the app is mapped to an App Engine subscription, the **Subscription Monitoring** tab is visible.
- If the app is mapped to a different subscription such as ITSM, the **Subscription Monitoring** tab is not visible.
- If the app is mapped to App Engine, is switched to another subscription, and then is changed back to App Engine, the **Subscription Monitoring** tab is visible again.

App Engine admins do not have permissions to change subscriptions. However, if an app is in the production environment, and there is a **Subscription Monitoring** tab, the App Engine admin can infer that the app is mapped to an App Engine subscription. If the **Subscription Monitoring** tab is not there, that app isn't mapped to App Engine.

You can select a user's name in the **Assigned Users** section to see a role inheritance map. This map shows how this user inherited the role, whether from a parent role, a group assignment, or other inheritance. This information can help you take action if someone has a role you think they shouldn't, and you want to change it.

Role inheritance map



App deployments

View the list of deployments your app has gone through on the **Deployment history** tab.

Deployment history information provides several benefits:

- If an app is deployed to production and there's a defect, you can quickly see when the app was deployed and track down the problem.
- If an app is deployed multiple times, you can quickly track what's changed over multiple deployments.
- If an app is deployed to production, you can quickly see who approved the app deployment.

App collaborators

View the collaborators for the app by selecting the **Collaborators** tab. All collaborators are stored on the Development instance.

If you need to fix an issue or defect with an app, you can quickly see who has permissions to make changes and what those permissions are.

Managing developers using AEMC

View details about developers working on apps in App Engine Studio and Creator Studio using App Engine Management Center (AEMC).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

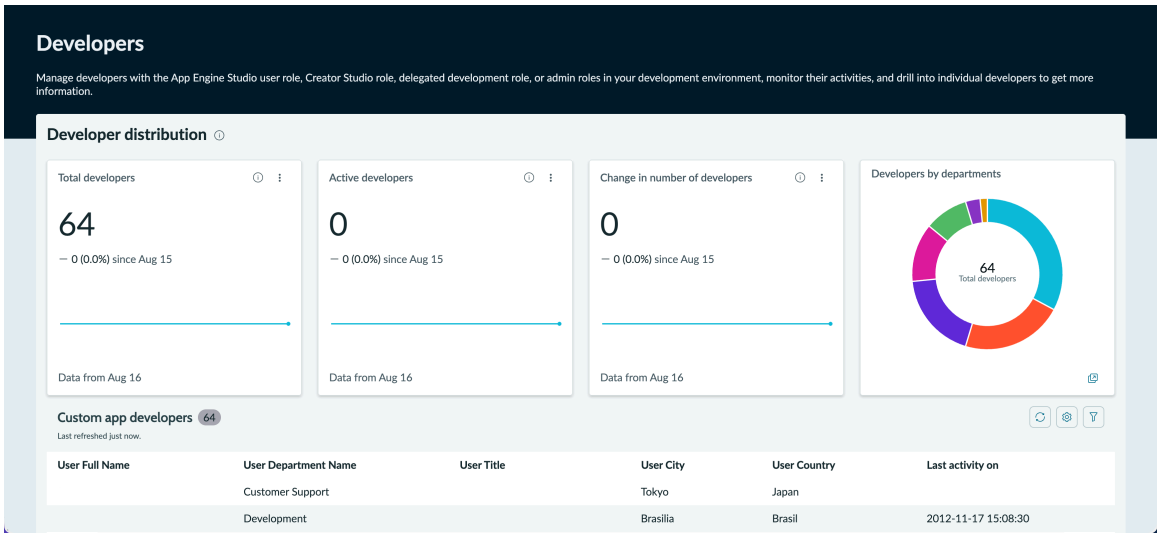
The App Engine adoption metrics section of the AEMC Overview page shows how many developers are working in App Engine Studio (AES) and their departments. Access more information by selecting the **Developers** page.

The Developers page in AEMC shows all developers with the following roles:

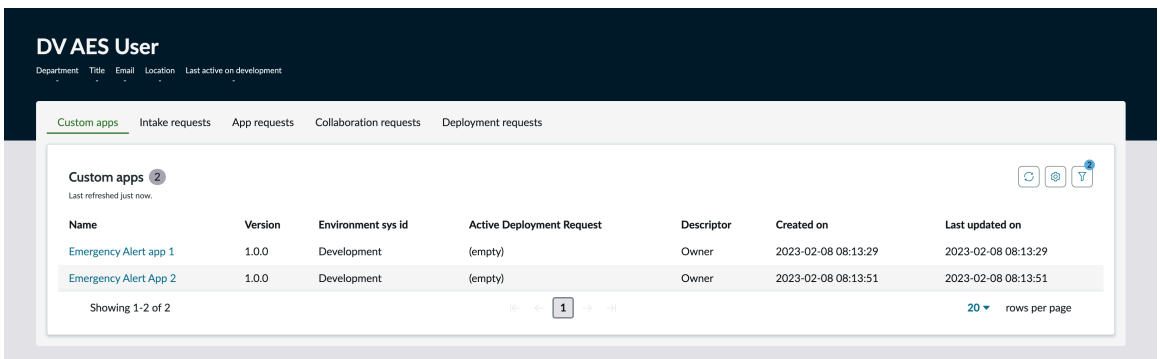
- AES Users role
- AES User Limited role
- Creator Studio Users role
- Creator Studio Restricted Users role
- Delegated development role in your development environment

On this page, you can monitor their activities and select individual developers to get more information.

Note: To remove a developer from the developer tab, you can either revoke all their permissions in Creator Studio, App Engine Studio, delegated development, and admin roles, or deactivate their account on the development instance.



Select a developer record to see more information about the custom apps they have created and their intake requests, app requests, collaboration requests, and deployment requests. You can filter the content on each tab to find specific results, or select an app from this page to see its usage, deployment history, and collaborators. Select any of the displayed records to view the history or to approve, reject, or delete the requests.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Assign Developer Sandboxes packs

Assign Developer Sandboxes packs to managed instances from the App Engine Management Center (AEMC).

About this task

Developer Sandboxes are packaged in increments of 10, called packs. For more information about Developer Sandboxes entitlements and limits, see Developer Sandboxes entitlements.

Before you begin

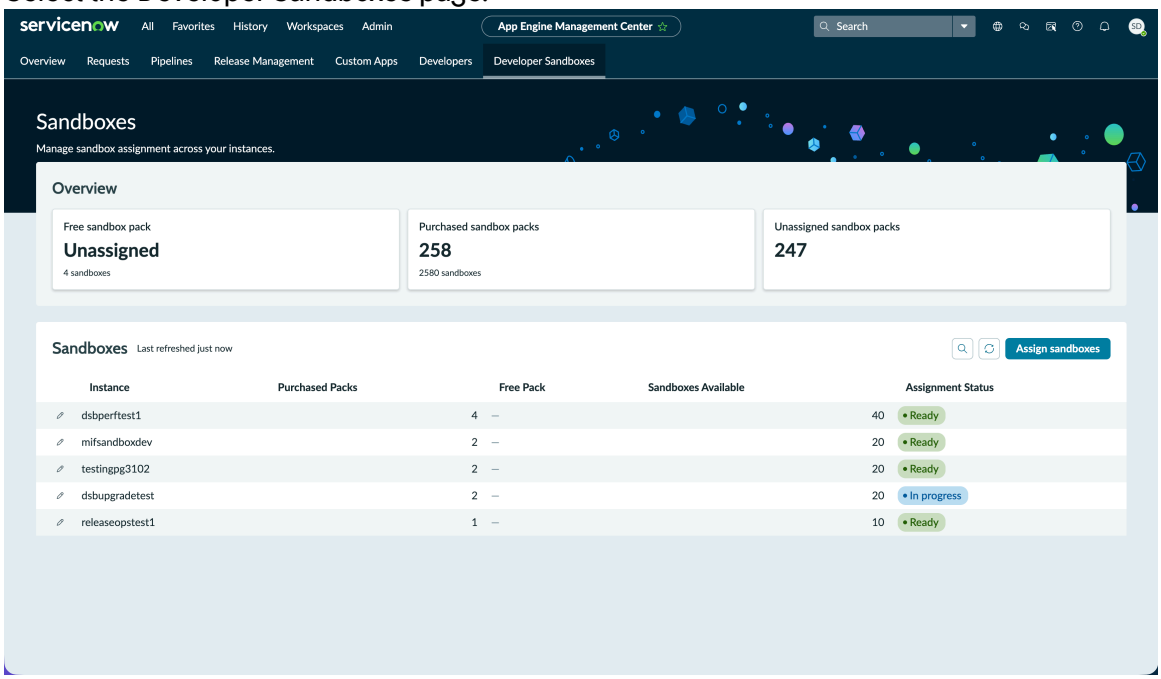
Warning: Developer Sandboxes pack assignments are permanent after they are submitted. Plan your sandbox pack assignment before you begin to avoid unintentional assignment.

- The Developer Sandboxes Licensing plugin (com.glide.dsb.licensing) must be installed on your controller instance.
- The Developer Sandboxes plugin (com.glide.dsb) must be installed on the instances you want to assign Developer Sandboxes packs to.
- Multi-Instance Management is required for assigning Developer Sandboxes licenses. It depends on the Multi-Instance Setup app (`sn-app-amf`), which must be installed separately from the ServiceNow® Store. This app is not a dependency of the `com.glide.dsb.licensing` plugin and is not installed automatically. For details on adding managed instances, see Connecting controller and managed instances.

Role required: sn_dsb_commons.sandbox_license_admin or admin

Procedure

1. Navigate to **All** > **App Engine** > **Administration** > **App Engine Management Center**.
2. Select the Developer Sandboxes page.

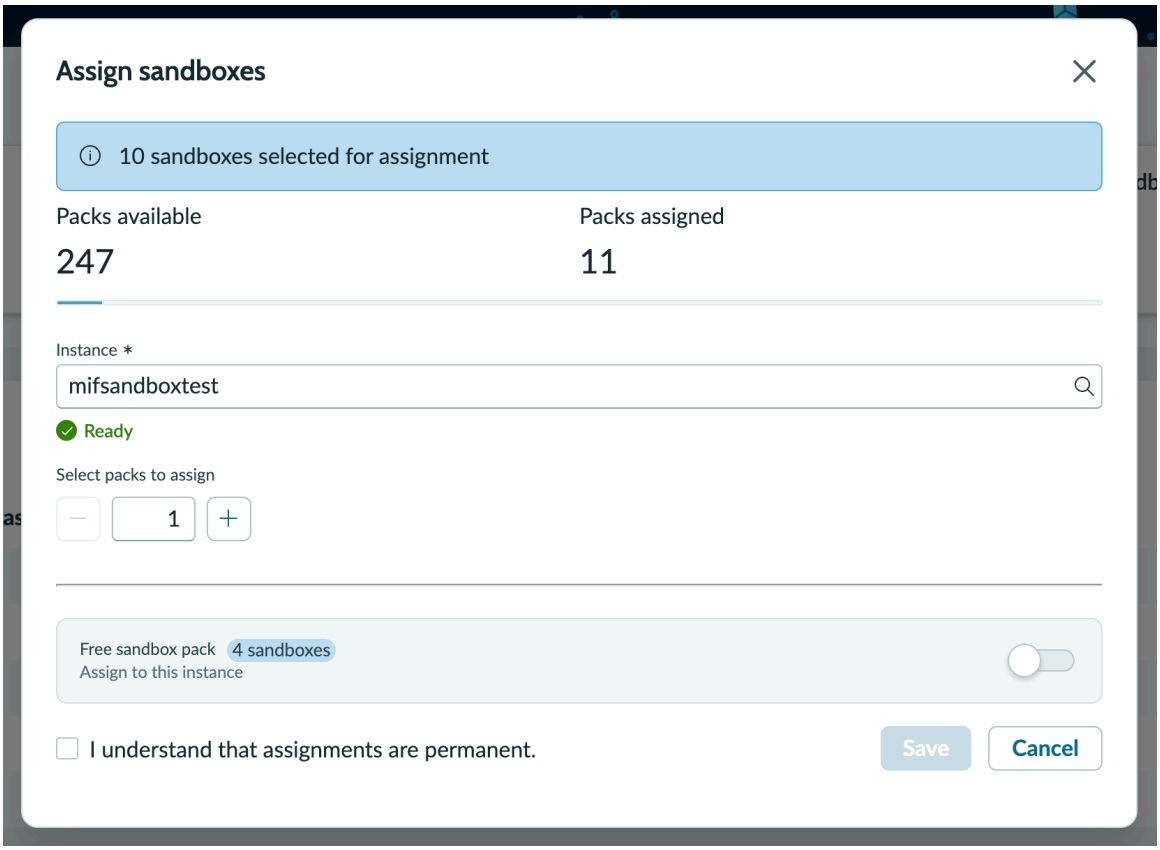


3. Select **Assign sandboxes**.
4. In the **Instance** field, select the instance to assign Developer Sandboxes packs to.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Multi-Instance Management is required for assigning Developer Sandboxes licenses. It depends on the Multi-Instance Setup app (`sn-app-amf`), which must be installed separately from the ServiceNow® Store. This app is not a dependency of the `com.glide.dsb.licensing` plugin and is not installed automatically. For details on adding managed instances, see Connecting controller and managed instances.

The **Assign sandboxes** modal displays **Ready** by the instance name to indicate that the selected instance can be assigned sandboxes (has the prerequisite Developer Sandboxes required plugins installed and Multi-Instance Management configured). When an instance does not meet the requirements for sandbox assignment, an error message appears.



5. Select the plus icon to add packs or the minus icon to remove packs.
   The maximum number of packs you can assign to an instance is 3 (30 sandboxes).
6. **Optional:** If a free pack is available, toggle on the switch to allocate the 4 free sandboxes to the selected instance.
7. Select the check box **I understand that assignments are permanent**.
8. Select **Save**.

Result

You have assigned Developer Sandboxes packs.

What to do next

For more information about Developer Sandboxes, see Developer Sandboxes.

Managing deployments using Pipelines and Deployments

Manage deployments using Pipelines and Deployments in the App Engine Management Center (AEMC).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Manage deployment requests

Review and approve or reject deployment requests in the App Engine Management Center (AEMC) as applications move from development to production.

Before you begin

Role required: sn_app_eng_notify.app_engine_admin

About this task

To view deployment requests in AEMC, pipelines must be configured correctly. For details, see Configure Pipelines and Deployments.

Procedure

1. Navigate to **All** > **App Engine** > **Administration** > **App Engine Management Center**.

   The AEMC dashboard shows the number of deployment requests.

   Note: To view a list of all deployment requests, either select the number of requests or select the Requests page.

2. Select a deployment request to review its details.
3. Review the details of the deployment request, including details about the application, the developer who requested deployment, and the deployment process.

   At the top of each deployment request, you can see the current position of the app in the pipeline.

   Note: If the request is in a testing environment, you can review the Automated Test Framework (ATF) test and instance scan results in the Activity stream or on the **Deployment Environment Results** tab.

4. Review the notes and application activity for the app manifest and any important details the developer may have added.
5. If a different App Engine Studio (AES) admin must review the deployment request, update the request assignee and select **Save**.
6. Approve or reject the request.
   - To move the application to the next deployment stage, select **Approve & Deploy app**.
   - If you have integrated your existing Change Management processes with AEMC, select **Approve & Create Change Request**. This action creates a change request, which can be seen and monitored on the **Change Requests** tab.
   - If the application is not ready or could potentially cause issues if it was moved on to the next deployment stage, select **Reject**. Add **Work notes** to identify the reasons for the rejection and any changes required to make the application ready for deployment.

     Note: Unless you are assigned a Change Management role, most of the information on the **Change Request** tab is read-only. You can monitor the change request and see as it changes states. However, if you have Change Management permissions, you may be able to update certain fields on the change request such as the planned start and end date or the state.

Schedule app deployments in AEMC

Schedule apps to deploy to production at a future time to load balance your systems using the App Engine Management Center (AEMC).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin
Role required: admin or app_engine_admin

About this task
Deploying apps through your pipelines requires a certain amount of processing power. Depending on the needs of your organization, you may want to schedule deployments to happen when less processing power is needed elsewhere or when it's generally more convenient.

Procedure
1. Navigate to **All** > **App Engine** > **Administration** > **App Engine Management Center**.
2. On the **Requests** page, select the deployment request for an application that is ready to go to your production environment.
3. Select **Deploy app**.
4. Either deploy your app now or set up a future deployment time.
   - To deploy now, select **Deploy**.
   - To deploy in the future, select **Schedule for later**.
5. **Optional:** If you scheduled the deployment for later, use the date and time picker to select when you want your app to deploy, select **OK**, and select **Deploy**.
   The Activity stream shows the deployment details and the version of the app that will be deployed at the selected date and time. The app is deployed within fifteen minutes of the selected deployment time. If you want to change this time frame, update how often the deployment workflow checks for a scheduled deployment record by navigating to **All** > **System Definition** > **Scheduled Jobs** > **Scheduled Deployments**.
6. Select **Save**.

Update a deployment request in AEMC

Change the details of a deployment request in App Engine Management Center (AEMC).

Before you begin
Role required: admin

Procedure
1. Navigate to **All** > **App Engine** > **Administration** > **App Engine Management Center**.
2. On the **Requests** page, select the deployment request that you want to update.
3. On the **Scheduled Deployments** tab, select the deployment record you want to update.
4. In the **Scheduled Deployment Date** field, select an updated date and time for the app to deploy to production.
5. Select **Save**.

Cancel a scheduled deployment in AEMC

Cancel a scheduled app deployment in App Engine Management Center (AEMC) if you no longer need to deploy the app to production. Canceling the scheduled deployment request cancels the scheduled deployment and also the entire request itself.

Before you begin
Role required: admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **App Engine** > **Administration** > **App Engine Management Center**.
2. On the **Requests** page, select the deployment request that you want to update.
3. On the deployment request, select **Cancel deployment**.

Deployment Request form in the Pipelines and Deployments app

The Deployment Request form is accessed from within the Pipelines and Deployments section of the App Engine Management Center.

To view a Deployment Request form, navigate to **All** > **App Engine** > **Administration** > **App Engine Management Center**. Then select the **Requests** page and open a record.

The fields that are included on a Deployment Request form are listed in the following table.

Deployment Request form

Field	Description
Number	Number to track the deployment request.
Assignment group	Group to which the deployment request is assigned.
State	State of the deployment request. All states are automatically updated by the pipeline flows: **New** The submitted request is new. **In Review** The submitted request is ready for review. **Closed - Published** The submitted request has been deployed to the production instance. **Closed - Rejected** The submitted request is not ready to publish. This state is enabled when you select **Reject** on the associated approval record at any time during the deployment process. The system then prompts you to include the reason why the application is rejected. These comments are then sent via email to the address included in the **Requestor Email** field on the **Requestor Details** related list. **Closed - Failed** This state is enabled when any of the pipeline flows or sub-flows fail for any reason. **Canceled** The submitted request has been canceled.
Assigned to	The user in the assignment group to whom this request is assigned.
Short description	A brief description for the request.
Release notes	Text that appears on the ServiceNow Store for this application after it is published.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Deployment Request form (continued)

Field	Description
App Details	
App name	Name of the submitted application.
App version	Version of the submitted application.
App Sys ID	System ID of the submitted application.
App Development URL	Link to app in App Engine Studio, Creator Studio, or ServiceNow Studio.
Requestor Details	
Requestor First/Last Name	First and last name of the user who requested to the app deployment.
Requestor User ID	Identifier of the user who requested to the app deployment.
Requestor Email	Email address of the user who requested to the app deployment.
Deployment Details	
Originating Environment	The instance of the originating environment.
Pipeline	Pipeline being used for deployment.
Current Environment	The target environment for the app.
Notes	
Additional comments (Customer visible)	Any additional comments to be shared with the customer, including the app manifest, which lists the number of each type of file in the app, and any ATF or instance scan results are shown.
Work notes	Internal notes.

When you have saved the form, additional tabs for approval-related information display. The App Engine admin can use these tabs to approve the request.

ReleaseOps

ReleaseOps automates deployment of changes across your pipeline, increases predictability and reliability of deployments, and reduces the risk of releasing changes to production.

https://player.vimeo.com/video/1210565804?h=baf385f190&badge=0&autopause=0&player_id=0&app_id=58479

Learn about ReleaseOps deployments.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Get started

Explore	Configure	Use
 Learn about ReleaseOps benefits and features.	 Configure ReleaseOps settings.	 Manage deployments with ReleaseOps.
Reference  Get details about ReleaseOps deployment and release states.		

Exploring ReleaseOps

ServiceNow® ReleaseOps is a solution to problem of deploying changes, customizations, and custom apps on the ServiceNow AI Platform. By automating the deployment process, ReleaseOps helps to increase the predictability and reliability of deployments, while also reducing the risk of releasing unwanted changes to production.

ReleaseOps overview

ReleaseOps automates and enhances the process of deploying changes, customizations, and custom applications on the ServiceNow AI Platform. ReleaseOps improves the existing pipelines deployment process by internally managing cross-instance trust and credential sharing, simplifying the setup and configuration of custom pipelines. ReleaseOps enables you to deploy changes using update sets and trigger deployment from directly within the ServiceNow Studio development environment. In addition, ReleaseOps leverages the automation capabilities of the ServiceNow Playbooks, resulting in deployments that are less error-prone and manual.

ReleaseOps handles deployments through releases. Releases define which changes, customizations, or custom apps will be moved to the production (or target) instance and when the changes will be deployed. The changes within a release are contained in deployment requests. Each deployment request contains one or more update sets or application installs. For more information about releases and deployment requests, see Releases in ReleaseOps and Deployment requests in ReleaseOps.

Releases are managed from your production instance, where the playbook is executed. Actions are orchestrated across instance with a cross-instance communication layer, alleviating credential setup across instances.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ReleaseOps availability

ReleaseOps is not supported in regulated environments or on-premise. Check your entitlements to determine whether you have access to ReleaseOps.

AI features in ReleaseOps

Note: Depending on your license, you will have access to certain application features, generative AI skills, agentic workflows, and AI agents. For more information, see ServiceNow product tiers.

ReleaseOps contains AI features that enable you to generate release notes and update set descriptions. For more information, see Exploring release lifecycle documentation AI agent.

ReleaseOps users

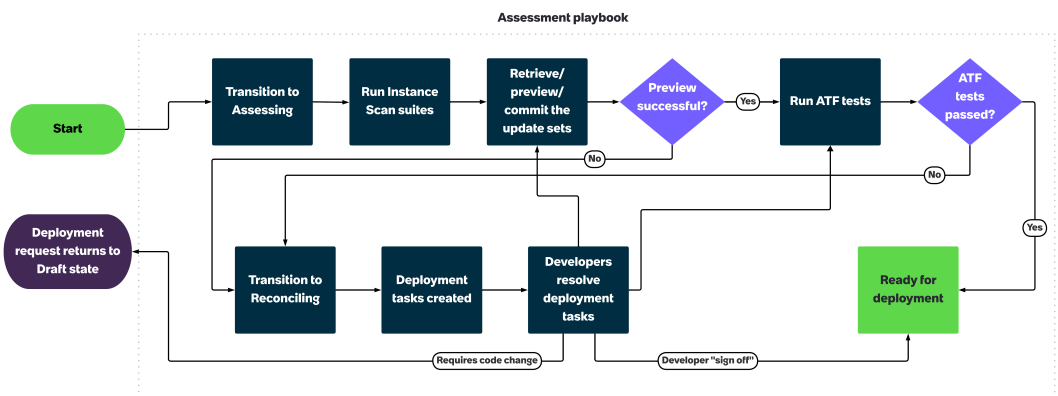
Users

Role	Description
Pipeline manager	A pipeline manager is responsible for creating and managing pipelines, including customizing playbooks. Pipelines define the flow of your deployments.
Release manager	A release manager is responsible for creating and scheduling releases, as well as verifying that the contents of a release are both relevant and correct. The release manager can remove deployment request items that don't meet the criteria for the release.
Developer	A developer is responsible for creating deployment requests to push application installations or update sets of work completed through the pipeline.
Tester	A tester is responsible for signing off on ATF test failures that occur to enable a deployment to continue. A tester can also send it back to development.

ReleaseOps workflow

The following workflow illustrates the sample pipeline workflow installed with ReleaseOps. Your pipeline manager can customize it as needed.

ReleaseOps sample pipeline workflow

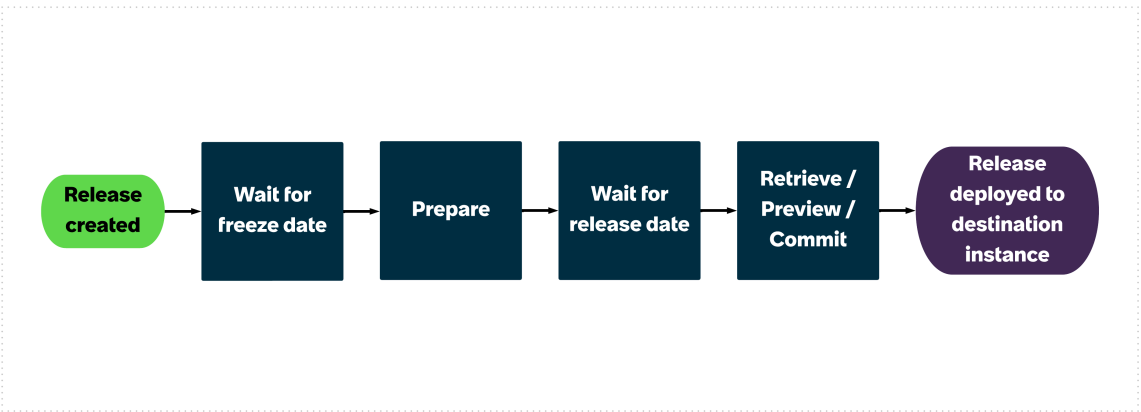


© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

In the traditional development to test to production workflow:

1. The release manager creates a release, which is associated with a pipeline.
2. Developers create and promote update sets, adding them to an existing deployment request or create a new deployment request, which is targeted to a release.
3. Developers add runbook tasks to deployment requests to enable custom and manual activities to occur throughout the assessment and release playbook stages.
4. When the update sets within a deployment request are functional and ready to be deployed, a developer sets the deployment request state to the **Ready to assess**.
5. The assessment playbook runs, during which Automated Test Framework (ATF) test suites and instance scans are run on the changes in the deployment request.
6. If there are failures during assessment, deployment tasks are created that the tester can sign off on or redirect to the developer to address.
7. If there are any runbook tasks set to occur during the assessment playbook stage, progression pauses until the runbook tasks have been addressed.
8. Once the deployment and runbook tasks have been addressed, the deployment request state is set to **Ready for deployment**.
9. On the scheduled date for the release, the release playbook runs.
10. If there are any runbook tasks set to occur during the release playbook stage, progression pauses until the runbook tasks have been addressed.
11. Once the runbook tasks have been addressed, all deployment requests in the **Ready for deployment** state move through the pipeline to production.

ReleaseOps sample release workflow



In the traditional release workflow:

1. A release manager creates a release and sets the state to **Active**.
2. On the scheduled freeze date, the release begins preparing for deployment.
3. During the preparation process, deployment requests that aren't ready are set to **Deferred** and can be attached to a future release.
4. Once preparation is complete, the release state is set to **Ready for deployment**.
5. When the release date arrives, the update sets within the deployment requests move from test to production in the order that they were deployed in the last instance.
6. The release and deployment requests are set to **Complete**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ReleaseOps benefits

ReleaseOps benefits

Benefit	Feature	Role
Define requirements for an application or update to get installed on a target instance.	Pipelines in ReleaseOps	Pipeline Manager
Deploy changes to a production or another target environment.	Releases in ReleaseOps	Release Manager
Push application installations or update sets of work completed through the pipeline with deployment requests.	Deployment requests in ReleaseOps	Developer
Scan deployment requests for changes to the current state of the production instance or target instance with the deployment analyzer. Use those findings to determine your actions in the pipeline.	Deployment analyzer in ReleaseOps	Developer
Leverage Automated Test Framework (ATF) code coverage to improve the efficacy of ATF test suites.	Deployment analyzer in ReleaseOps	Developer
Add custom, flexible, and manual activities to your assessment and release playbooks, without having to adjust the structure of your playbooks each time.	Runbook tasks in ReleaseOps	Developer

What to explore next

To learn more about configuring and using ReleaseOps, see:

- Configuring ReleaseOps
- Using ReleaseOps to manage deployments
- Promote an update set for deployment
- Create a deployment request for a scheduled release
- Create a deployment request for an on-demand release
- Attach an update set to an existing deployment request
- Create a release

Pipelines in ReleaseOps

A pipeline is the flow of a deployment in ReleaseOps. A pipeline's flow is defined within playbooks, which enables you to customize as needed.

A pipeline consists of two stages, each represented by its own playbook: an assessment stage and a release stage.

- The assessment stage moves changes in individual deployment requests through subprod instances and is designed to do deployment analysis and run scans and tests.
- The release stage moves all changes within a release across its member deployment requests that have passed assessment by the scheduled date to production.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ReleaseOps includes sample playbooks to that include basic pipeline structure and capabilities. You can use the ReleaseOps sample playbooks or duplicate and customize them for your company's processes. The ReleaseOps sample playbooks are the release deployment playbook, deployment request assessment playbook, and on-demand deployment request assessment playbook.

Sample playbooks included with ReleaseOps

Playbook	Stage	Description
Deployment request assessment playbook	Assessment stage for a standard/scheduled release	Moves update sets in deployment request to the test instance and executes defined Automated Test Framework (ATF) tests (as specified in the deployment request and the pipeline).
On-demand deployment request assessment playbook	Assessment stage for an on-demand release	Runs the deployment analyzer and Instance Scan with sample rules to either enable or deny an on-demand deployment. Doesn't run any ATF tests.
Release deployment playbook	Release stage	Prepares a release, including moving deployment requests that aren't ready out of the release and calculating update set ordering. Used for both on-demand and scheduled releases.

A pipeline maps intermediate instances to the playbook. For example, the sample playbooks reference a test instance, which must be mapped to a physical deployment instance. The actual pipeline runs from the source instance specified in the deployment request, to the destination instance specified in the release. Intermediate instances (such as test) are defined in the pipeline. ATF test suites can also be specified at the pipeline level to enforce a given list of tests to be run when a given pipeline is used.

Pipelines can use the sample ReleaseOps playbooks, custom playbooks, or a mix of both. Multiple pipelines can leverage the same playbooks by mapping a different instance definition, which is then referenced in the playbooks.

Because the activities required during deployment might vary from release to release, ReleaseOps also provides a more flexible solution to having to create and adjust custom playbooks each time: runbook tasks. Runbook tasks can be added at the deployment request level and enable you to define the kind of activity (manual or automated) needed at the given playbook stage. For more information about runbook tasks, see Runbook tasks in ReleaseOps.

Releases in ReleaseOps

Scheduled and on-demand releases are how changes are deployed to target instances with ReleaseOps. Releases can contain one or more deployment requests.

Creating a release

Generally, a release manager is responsible for creating releases and mapping a team or organization's release workflows to a release cadence.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

There can be a single release stream or multiple release streams available at any given time. However, only one release can be in the process of deploying to a target instance at any time.

For example, an organization might have two release tracks that they define:

- A regular weekly cadence for releases, at 10:00 each Friday.
- A daily cadence for high-priority releases, at 12:00 each day.

The weekly and daily release tracks can coexist without issue.

For more information about creating a release, see Create a release.

Release types

Releases that are planned and scheduled in advance are considered scheduled releases. These releases can consist of one or more deployment requests. Each deployment request in the release can include of one or more update sets.

On-demand releases are created dynamically when the need arises, such as deploying changes to patch a defect in an application. When a developer creates a deployment request and selects the option for an on-demand release, ReleaseOps dynamically creates a release to drive this request to its target instance. As a result, only one deployment request can be attached to an on-demand release.

Important: Changes in an on-demand release are intended to be deployed to their target as soon as possible. However, on-demand releases must still pass the checks in the organization's pipeline. Doing so helps to verify that the changes meet the requirements for the organization's compliance and governance policies, as well as quality assurance checks. Once the on-demand request has passed these checks, the deployment proceeds immediately.

ReleaseOps includes deployment analyzer, a tool provided to introspect a deployment. The deployment analyzer enables release administrators to verify that on-demand releases meet certain criteria, such as only making specific types of code changes. For more information, see Deployment analyzer in ReleaseOps.

Deployment requests in ReleaseOps

Deployment requests contain one or more update sets. They are attached to a release, which runs through a pipeline to a destination environment.

Deployment requests are the building blocks of moving code into production. They capture the content that a developer intends to move, including update sets. Deployment requests also describe how and when that code should go into production.

When creating a deployment request, you can attach it to either a release or a pipeline, if the release hasn't been created yet, which enables the request to move to the assessment stage. However, for the deployment request to move to production, a release must be defined.

Deployment analyzer in ReleaseOps

The deployment analyzer reviews the update set in your release against rules to verify compliance with customizable rules.

The deployment analyzer scans a given deployment request (and the update sets it contains) to check for changes as compared to the current state of the production instance (or the target release environment). To do, the deployment analyzer relies on rules and Automated Test Framework (ATF) code coverage. The deployment analyzer operates as a playbook activity, the results of which can be consumed to decide different actions in the deployment process.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Deployment analyzer rules

By default, the deployment analyzer has five rules. You can create rule types, definitions, and rules as needed. New rules appear in the playbook conditional setup automatically.

For more information, see Deployment analyzer rules.

Automated Test Framework (ATF) code coverage

The deployment analyzer also leverages Automated Test Framework (ATF) code coverage to evaluate code changes prior to deploying the changes to the production (or target) instance. ATF code coverage is a tool that enables you to determine what percentage of code in a deployment request is covered by ATF test suites.

By default, if ATF test suites cover less than 70 percent of the code in a deployment request, ReleaseOps transitions the deployment request to the **Reconciling** state and a test failure task is automatically created. You can adjust the ATF code coverage threshold in the deployment request assessment playbook. For more information, see Set Automated Test Framework (ATF) code coverage.

For more information about ATF code coverage, see ATF Code Coverage API.

Runbook tasks in ReleaseOps

Learn about how runbook tasks enable flexibility in ReleaseOps deployments, without needing to customize playbooks each time.

Many organizations use spreadsheets to track all of the manual and automated activities that must be completed to finish a software deployment or release. ReleaseOps runbook tasks enable you to add those activities directly into your ReleaseOps playbooks, without having to restructure your playbooks as activities vary from one release to the next.

Runbook tasks are activities that pause the progression of a playbook until the specified task is resolved. Runbook tasks are added at the deployment request level, where you specify the playbook, playbook stage, and the point at which the runbook task should occur (at either the beginning or end of the playbook stage). Common types of runbook tasks might include:

- Application install/update
- Data import - XML
- Manual action
- Script - scheduled job
- Update set commit

When the playbook executes and encounters a runbook task, the playbook pauses progression until the runbook task is resolved.

To add runbook tasks to your deployment requests, see Create a runbook task.

Exploring release lifecycle documentation AI agent

Learn about how the release lifecycle documentation AI agent works to help you to manage application development at scale by generating useful, transparent documentation throughout the app lifecycle.

Note: Depending on your license, you will have access to certain application features, generative AI skills, agentic workflows, and AI agents. For more information, see ServiceNow product tiers.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Release lifecycle documentation AI agent overview

The release lifecycle documentation AI agent is an autonomous AI agent that enables you to generate readable, useful documentation across the application lifecycle, including release notes and update set descriptions. Using the release lifecycle documentation AI agent can help to improve traceability, governance, and release transparency through improved documentation, while also reducing the manual effort required to document changes.

Release lifecycle documentation AI agent users

Users

User	Description
Developer	Quickly generate update set descriptions and release notes to provide greater visibility into the changes made.
Release manager	Review update set descriptions and release notes to understand the changes in a given deployment request or release.

Release lifecycle documentation agent benefits

Release lifecycle documentation agent benefits

Benefit	Feature	Users
Support effective communication and collaboration across teams, instances, and changes by adding clear descriptions to update sets.	Generate an update set description	Developer
Improve app governance and traceability over time by documenting the changes in a release.	Generate release notes	Developer

What to explore next

To learn more about configuring and using the release lifecycle documentation AI agent, see:

- Configure release lifecycle documentation AI agent
- Generate an update set description
- Generate release notes

Configuring ReleaseOps

Configure ReleaseOps to move changes from development to testing to production.

Starting with version 1.2.1 of ReleaseOps, you can complete initial configuration with ReleaseOps guided setup. Use the following procedure to complete ReleaseOps configuration using guided setup:

1. Install ReleaseOps on each instance that participates in your ReleaseOps ecosystem.
2. Complete ReleaseOps guided setup.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

3. Create a custom pipeline to add additional instances and playbook activities tailored to the unique needs of your organization.
4. Set Automated Test Framework (ATF) code coverage to enhance the efficacy of your ATF test suites.

You can also complete ReleaseOps configuration manually for more complex setup scenarios, and if you need to make additional changes to your ReleaseOps ecosystem after initial configuration. Use the following procedure to complete ReleaseOps manual setup:

1. Configure multi-instance management for instances using ReleaseOps.
2. Install ReleaseOps on each instance that participates in your ReleaseOps ecosystem.
3. Complete ReleaseOps manual setup.
4. Create a custom pipeline to add additional instances and playbook activities tailored to the unique needs of your organization.
5. Set Automated Test Framework (ATF) code coverage to enhance the efficacy of your ATF test suites.

Install ReleaseOps

Install ReleaseOps from the ServiceNow Store for the first time.

Before you begin

Role required: admin

Because ReleaseOps involves a multi-instance ecosystem, you must complete this procedure on each instance in your ReleaseOps ecosystem (for example, development, test, and production).

Procedure

1. Open the ServiceNow Store website.
2. In the **Search** bar, enter `ReleaseOps` and select enter or return on your keyboard.
3. Select the ReleaseOps tile.
4. Select **Get**.
5. Sign in with your ServiceNow user ID.
6. Run the installation on your instance, and accept all of the plugins.
   For more information about installing a ServiceNow® Store application on your instance, see Install an application or plugin.

Result

ReleaseOps is installed on your instance.

Complete ReleaseOps guided setup

Starting with version 1.2.1 of ReleaseOps, guided setup helps to make the initial configuration of ReleaseOps easier. Follow the steps in guided setup to begin using ReleaseOps quickly.

About this task

For more complex setup scenarios, or when making changes to your existing ReleaseOps ecosystem, use the procedure outlined in ReleaseOps manual setup.

When completing guided setup, you define several kinds of instances for your ReleaseOps pipeline. To learn more about each instance, see Instances in ReleaseOps configuration.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

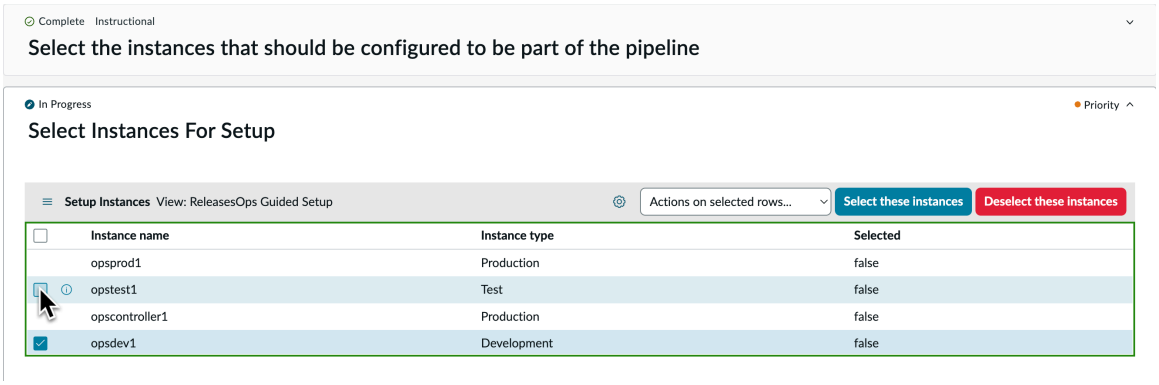
Before you begin

Role required: admin or sn_releaseops.releaseops_pipeline_admin

Procedure

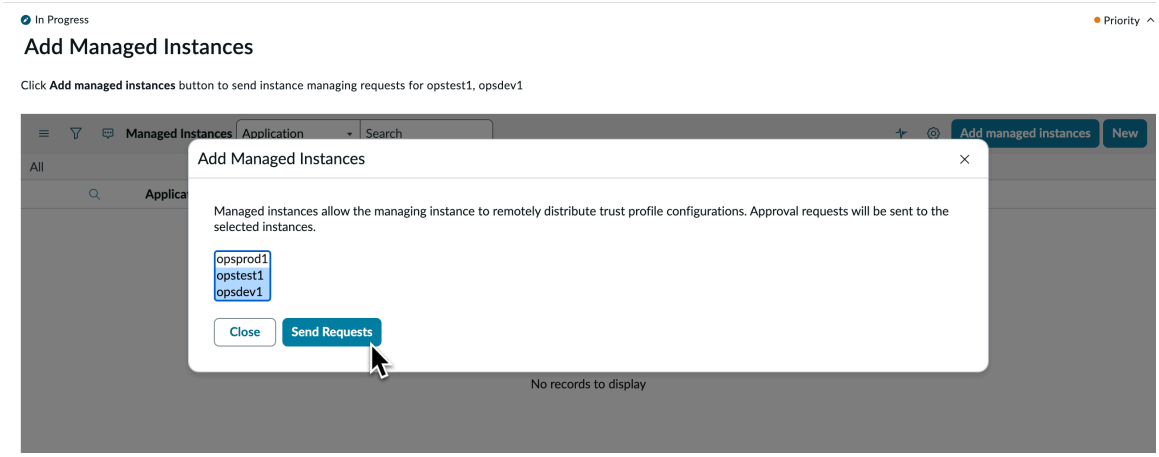
1. Navigate to **All** > **ReleaseOps** > **ReleaseOps Guided Setup**.
2. Select the instances that you want to participate in your ReleaseOps pipeline.
   a. In the **Select instances to configure** card, select **Start**.
   b. For each instance that you want to participate in your ReleaseOps pipeline, select the check box next to the instance name.

> **Important:** You must select your controller instance and at least once instance with an **Instance type** of "Test" when selecting instances for your pipeline. For more information about instances, see Instances in ReleaseOps configuration.



   c. Select **Select these instances**.
   d. To remove a selected instance from your pipeline, select the check box next to the instance name, then select **Deselect these instances**.
   e. Verify that the **Instance type** for each instance is correct.
   f. To change an instance type, double-click (or use the keyboard shortcut) in the **Instance type** field, select the correct instance type from the list, then select the save icon ( ).
   g. When finished, select **Mark as complete**.
3. Add the instances in your ReleaseOps pipeline as managed instances.
   a. Select **Add managed instances**.
   b. In the **Add managed instances** modal, select the instances that you selected in step 2. To select more than one instance from the list, use the keyboard shortcut to select multiple instances.
   c. Select **Send requests**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Approval requests are sent to each instance that you added as a managed instance.

d. Select **Mark as complete**.
Once you have added managed instances and sent approval requests to each instance, ReleaseOps validates each instance in the background. Validation might take several minutes. Once validation is complete, you can proceed to the next step of guided setup.

4. Approve the management requests for each managed instance in your pipeline.

Note: You must be an admin to approve instance management.

a. Select the link to the approval request for each managed instance in your pipeline.

b. If you aren't logged in to your managed instance, log in with your credentials.

c. On the approval request record, select **Approve Manager Instance**.

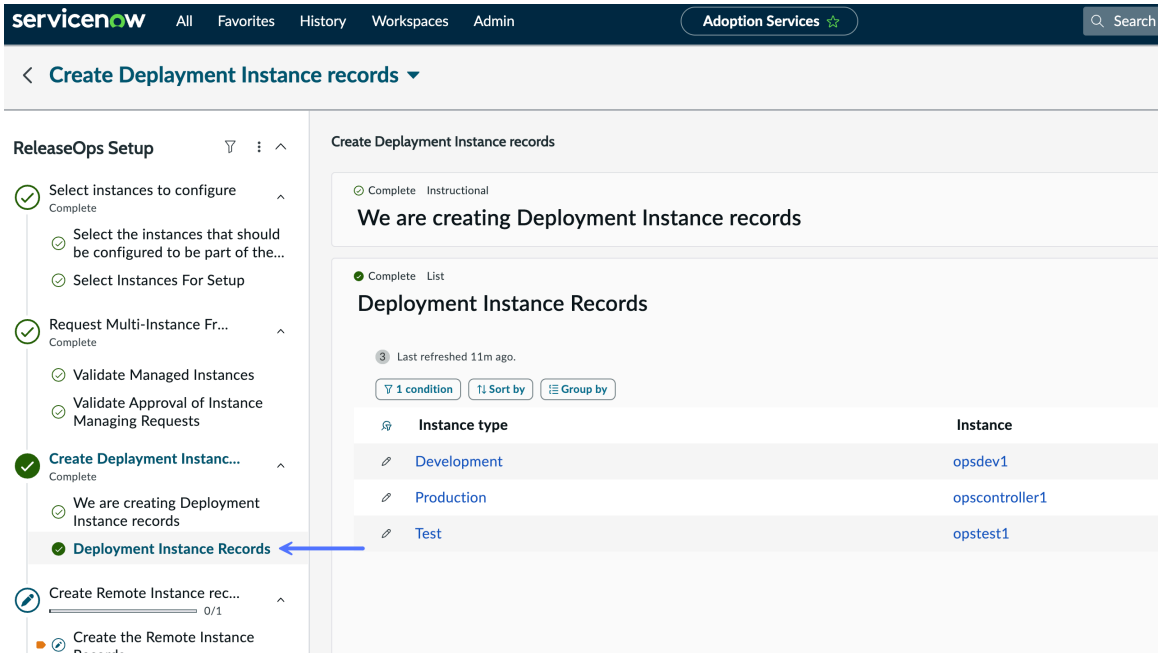
d. Return to your guided setup window.

e. Select **Mark as complete**.
Once you have approved the management requests for each managed instance in your pipeline, ReleaseOps validates the approvals in the background. Validation might take several minutes. Once validation is complete, you can proceed to the next step of guided setup.

5. Optional: Review the deployment instances for each instance that participates in your ReleaseOps pipeline by selecting **Deployment instance records** in the side panel.

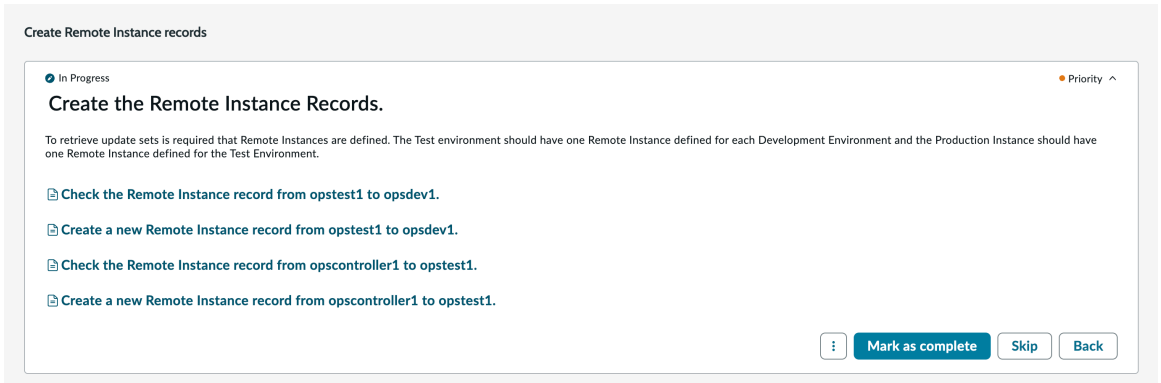
(Optional) During guided setup, ReleaseOps automatically creates deployment instances for each instance that you added in step 2.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



6. Create or verify the remote instances for each instance in your pipeline.

   Note: You must be an admin or an update set admin to view or create remote instance records.



   - If you already have remote instances set up, select **Check the remote instance record** for the existing remote instance. You might be prompted to log in to the instance with your credentials, if you aren't already logged in. Once logged in, the remote instance record displays.
   - If you don’t have remote instances set up, select **Create a new remote instance record** for the remote instance that you need to create. You might be prompted to log in to the instance with your credentials, if you aren't already logged in. Once logged in, a new remote instance record appears. You must complete the fields in the record and select **Test connection** to verify that the remote instance is accessible. Select **Submit** once the remote instance test is successful.

   a. Once you have created or verified the remote instances for each instance in your pipeline, return to your guided setup window and select **Mark as complete**.

7. **Optional:** Review the pipeline instances for each instance that participates in your ReleaseOps pipeline by selecting **Pipeline instance records** in the side panel.

8. Set the controller property on the development instance.

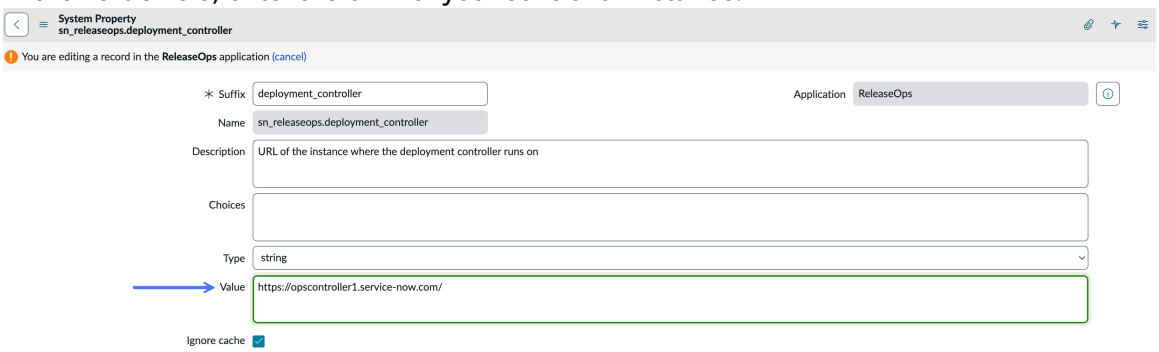
   Note: You must be an admin to edit the controller property.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. Select the link to open your development instance.

b. If you aren't already logged in, log in with your credentials.

c. In the **Value** field, enter the URL for your controller instance.



d. Save the record by selecting **Update**.

e. Return to your guided setup window.

f. Select **Mark as complete**.

9. Select **Close**.

Result

You have completed guided setup for ReleaseOps. You can start using ReleaseOps to manage your deployments. For more information, see Using ReleaseOps to manage deployments.

Complete ReleaseOps manual setup

Complete ReleaseOps manual setup to configure a new ReleaseOps ecosystem using the sample pipelines and playbooks.

About this task

Starting with version 1.2.1 of ReleaseOps, you can use guided setup for a quicker initial configuration experience. For more information, see Complete ReleaseOps guided setup.

The manual setup procedure outlined in this task can also be used to complete ReleaseOps initial configuration. Manual setup can be useful in more complex setup scenarios and when making changes to your existing ReleaseOps ecosystem.

Before you begin

Role required: admin or sn_releaseops.releaseops_pipeline_admin

Confirm that all instances (production, development, and test) have ReleaseOps installed. Your test instance must also have ATF Test Generator and Cloud Runner installed. For more information, see Install ReleaseOps.

Confirm that multi-instance management has been configured to enable all non-production instances (development and test) to be managed by the production instance. For more information, see Configure multi-instance management for instances using ReleaseOps.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. On your production instance, create a deployment instance record for each instance that you want to participate in your ReleaseOps ecosystem.

   Note: For the ReleaseOps sample release pipelines, the three instances are:
   - Development
   - Test
   - Production

   a. Navigate to **All** > *ReleaseOps* > **Deployment Instances**.

   b. Add a new record for each instance that participates in your ReleaseOps ecosystem by selecting **New**.

   c. Select the **Instance** field, then select the instance that you want to add to your *ReleaseOps* ecosystem.

   d. Select the **Instance type** field, then select the correct instance type from the list.

   e. Select **Submit**.

   f. Repeat steps b-e for each instance that participates in your ReleaseOps ecosystem.

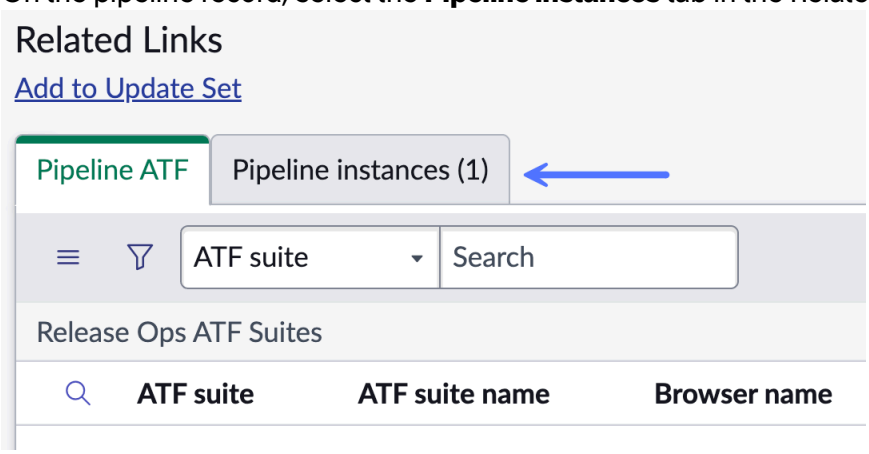
2. On your production instance, configure the test instance for both the sample On Demand Pipeline and sample Release Pipeline.
   For sample ReleaseOps pipelines, you must only configure a new pipeline instance for your test instance. You don't need to configure pipeline instances for your development and production instances, because these environments are defined within deployment requests and releases.

   For custom ReleaseOps pipelines, you can configure additional pipeline instances for the non-production instances that participate in your ReleaseOps ecosystem, excluding your development instance. For more information about custom pipelines, see Create a custom pipeline.

   a. Navigate to **All** > **ReleaseOps** > **Pipelines**.

   b. Select one of the sample pipeline records from the list.

   c. On the pipeline record, select the **Pipeline instances** tab in the Related Links section.



   d. On the **Pipeline instances** tab, select **New**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

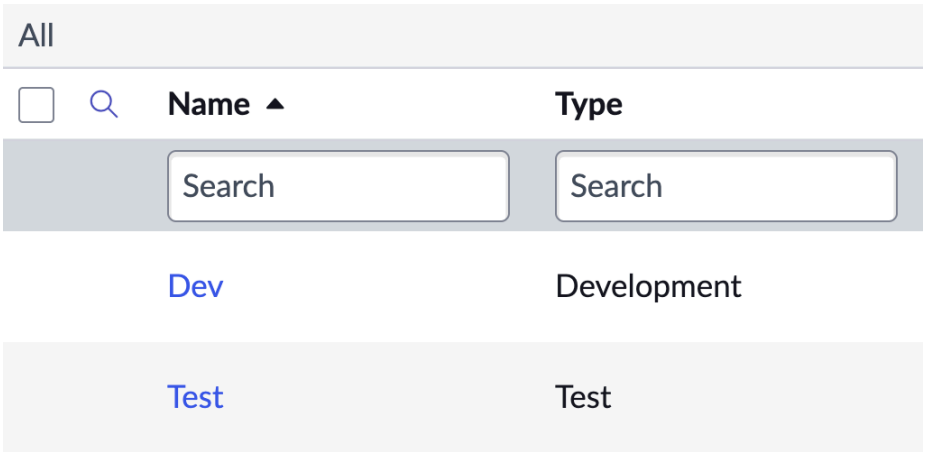
e. On the new pipeline instance form, select the Lookup using list icon ( ) for the **Deployment instance** field, then select the test instance from the list.

f. In the **Label** field, select **Test**.

g. Select **Submit**.

h. Repeat the process in steps b-g to configure test instances for each of your sample pipelines.

3. On your production instance, navigate to **All** > **System Update Sets** > **Update Sources** and verify that there are valid update sources defined for both your development and test instances.

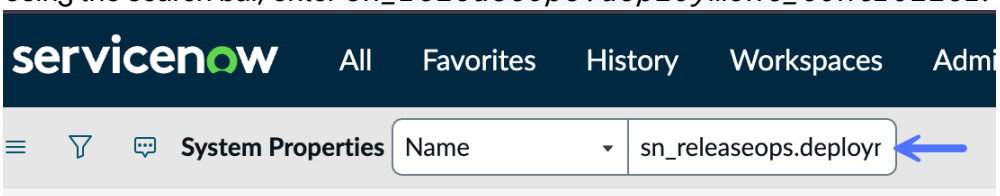


Note: If there aren't valid update sources defined for both your development and test instances, you can define them by navigating to **All** > **System Update Sets** > **Update Sources** on your development and test instances and creating a new update source record.

4. On your test instance, navigate to **All** > **System Update Sets** > **Update Sources** and confirm that there's a valid update source defined for your development instance.
5. On your development instance, set the property *sn_releaseops.deployment_controller* to the production (or deployment controller) URL.

a. Navigate to **All** > **System Properties** > **All**.

b. Using the search bar, enter *sn_releaseops.deployment_controller*.



c. Select the record for *sn_releaseops.deployment_controller*.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

d. In the **Value** field, enter the URL for your production (or deployment controller) instance.

e. Select **Update**.

Result

ReleaseOps is ready to deploy changes from your development to test to production instances.

Configure multi-instance management for instances using ReleaseOps

Manage the multi-instance configurations to enable access across instances in your ReleaseOps ecosystem.

Before you begin

Note: Starting with version 1.2.1 of ReleaseOps, you can configure multi-instance management during guided setup. For more information, see Complete ReleaseOps guided setup.

Role required: admin or sn_mif.mif_admin

You must have ReleaseOps installed before configuring mulit-instance management. For more information, see Install ReleaseOps.

All non-production (development and test) instances must be configured as instances managed by your production instance to participate in your ReleaseOps ecosystem. For more information, see Cross-instance application trust configuration.

Procedure

1. Within your production instance, navigate to **All** > **Multi-Instance Management** > **Managed Instances**.
2. Select **Add managed instances**.
3. In the Add managed instances dialog, select the check box for each instance that you want to add to your ReleaseOps ecosystem.

   Note: You can select and hold the Shift key to select more than one instance at a time.

4. Select **Send requests** to send the request to your selected instance or instances.
   It might take up to several minutes for the managed instance requests to appear on the instances that you selected.
5. Within your development and test instances, navigate to **All** > **Multi-Instance Management** > **Manager Instances**.
6. Open the record that was created from your production instance.
7. Select **Approve Manager Instance**.

Trouble?

If you receive Trust Profile errors, confirm that both Managed Instances and Manager Instances are correctly set up and approved. For more information, see Cross-instance application trust configuration.

Create a custom pipeline

Create a custom pipeline to move changes from your development environment to production (or your target instance) by duplicating one of the ReleaseOps sample playbooks.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

Role required: releaseops.pipeline_admin and playbook.admin or pd_author

For more information about playbooks, see Exploring Playbook.

About this task

Custom playbooks and pipelines enable you to create deployment structures tailored to the unique needs of your organization's development environment. Custom playbooks and pipelines are best suited for deployment processes that are consistent and expected to occur during each release. For deployment processes that are more variable, you might consider adding runbook tasks instead of custom playbooks or playbook activities. For more information, see Runbook tasks in ReleaseOps.

Procedure

1. Duplicate one of the ReleaseOps playbooks.

   Important: When creating a custom playbook, duplicate one of the ReleaseOps sample playbooks as your starting point. Don't customize the ReleaseOps sample playbooks directly, so that you can continue to refer back to them as new features are added during instance upgrades.

   a. Navigate to **All** > **Workflow Studio** and select the desired playbook.

   b. Enter a name for your new playbook and update any other required details.
   For more information, see Duplicate a playbook.

2. Add or update stages in the playbook as desired.

   Note: Certain stages make sense at certain parts of the flow. For example, the deployment analyzer runs before the deployment state **Move to test**. Other stages might need to be run after **Move to test** or after **Run ATF tests**. Typically, you shouldn't add anything after **Ready for deploy**.

   For more information, see Add and configure a stage in a playbook.

3. Add or update decisions in the playbook as needed.
   For more information, see Decision activities.

4. Select **Activate** to finalize your updates and make your new playbook available for use.

5. Create a pipeline that leverages your new playbook.

   a. Navigate to **All** > *ReleaseOps* > **Pipelines** and select **New**.

   b. Give your pipeline a name, and select playbooks for the **Assessment playbook** and **Release playbook** fields.
   You can use your own custom playbooks for both, or use a combination of custom and sample playbooks.

   c. Select and hold (or right-click) the header on the Pipeline form and select **Save**.

6. On your newly created pipeline, locate the Pipeline instances related list and select **New**.

   a. Select the label `Test.`

   b. Select your deployment instance.

   c. Select **Submit**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example: Create an assessment playbook

Create a custom pipeline to move changes through your non-production environment by duplicating the Deployment Request Assessment playbook. Add the deployment analyzer to detect if there are code changes in the deployment and skip Automated Test Framework (ATF) tests during assessment.

Before you begin

Role required: releaseops.pipeline_admin and playbook.admin or pd_author

Procedure

1. Duplicate the Deployment Request Assessment playbook.
   a. Navigate to **All** > **Workflow Studio** and select the **Deployment Request Assessment** playbook.
   b. Select **Duplicate** from the More actions menu.
   c. Enter a name for your new playbook and update any other required details.
2. Add a new stage to the playbook by selecting the + icon after Start, and enter a label, for example `Analyze Deployment`.

   Note: Certain stages make sense at certain parts of the flow. For example, the deployment analyzer runs before the deployment state (Move To Test). Other stages might need to be run after Move to Test or after Run ATF Tests. Typically, you shouldn't add anything after Ready for Deploy.

3. Add the **Run Deployment Analyzer** activity.
4. Select the newly added Run Deployment Analyzer activity to configure it.
   a. Select the **Automation** tab on the additional options modal.
   b. In the Inputs section of the modal, select the icon and then select **Parent Record - Deployment Request**.
   c. Select **Save and close**.
5. Locate the Run ATF Tests section of the playbook, and add a new decision after the Transition to Assessing activity by selecting the + icon, and then selecting **Add a decision (if/then)**.
   For more information, see Decision activities.
6. Select the newly added decision to configure it.
   a. Select the **Branches** tab, and enter a label, for example `Skip Tests`.
   b. Select **Add Condition**.
   c. From the Field dropdown menu, select **1.1 Deployment Analyzer** > **Outputs** > **Has Code Change**.
   d. Select **is** from the Operator dropdown menu.
   e. Select **false** from the True/False dropdown menu.
   f. Select **Save and Close**.
7. Hover over the Skip Tests branch, select **Drag new connector** icon, and drag the connector to the junction after the Create Test Failure Task.
   Confirm that the default branch is connected to Run ATF Tests for Deployment.
8. Select **Activate** to finalize your updates and make your new playbook available for use.
9. Create a pipeline that leverages your assessment playbook.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. Navigate to **All** > *ReleaseOps* > **Pipelines** and select **New**.

b. Give your pipeline a name, and select your newly created playbook in the **Assessment playbook** field.

c. Select and hold (or right-click) the header on the Pipeline form and select **Save**.

10. On your newly created pipeline, locate the Pipeline instances related list and select **New**.

a. Select the label `Test`.

b. Select your deployment instance.

c. Select **Submit**.

Result
Your custom pipeline, which leverages the custom playbook you created from the default Deployment Request Assessment playbook for its assessment stage, is now ready for use.

Note:

- The pipeline instance label (in the example above, the `Test` label entered for step 10a) is used to map the instance definition with the instance as referenced in the playbook. Mapping the instance definition with the instance label enables a playbook to be used by multiple pipeline definitions with different topologies (but the same process). For example, you can create another pipeline record and associate it to the same playbooks, but define a different test instance mapping.
- The source instance for the playbook is defined in the deployment request, so you can have multiple source or development environments for one release.
- The destination instance is defined in the release, so you can reuse the same pipeline to release to multiple destination or production environments.
- Only intermediate instances must be mapped.

Instances in ReleaseOps configuration

Learn about the different types of instances in ReleaseOps configuration.

Controller instance and managed instances

The terms controller instance and managed instances are specific to multi-instance management. ReleaseOps uses multi-instance management to handle trust configurations across instances and orchestrate deployments from a single instance. The instance that manages the other instances and orchestrates deployments is the controller instance. The other instances that the controller instance manages are the managed instances. During ReleaseOps configuration, you must add managed instances and authorize your controller instance to manage those instances.

For more information about multi-instance management, see Cross-instance application trust configuration. To see a complete list of managed instances, you can navigate to **All** > **Multi-instance management** > **Managed instances**.

Deployment instances and pipeline instances

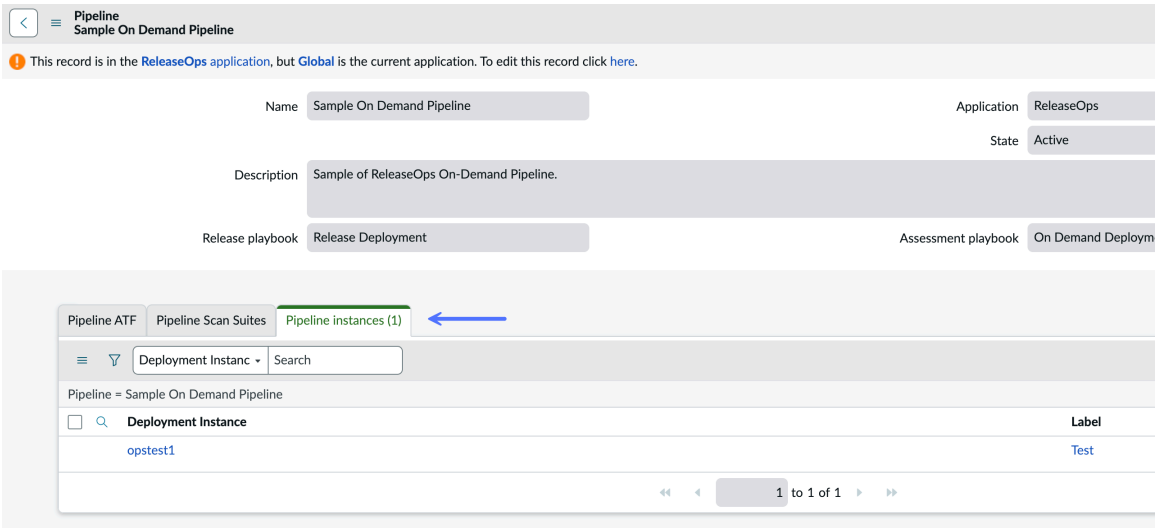
A deployment instance defines the environment of the instance, such as development, test, or production. During ReleaseOps configuration, deployment instances are created for each instance in your pipeline. You must define the environment type for each deployment instance.

You can view a complete list of deployment instances and their environment types by navigating to **All** > **ReleaseOps** > **Deployment instances**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

A pipeline instance maps a deployment instance to the pipeline that it participates in. Only intermediate instances are mapped using pipeline instances, because the source and destination instances are specified in the deployment request record and the release record respectively. For example, in a three instance (development, test, production) pipeline, the test instance is the only instance that must be defined with a pipeline instance.

To view a list of pipeline instances for a given pipeline, navigate to **All** > **ReleaseOps** > **Pipelines** and select the pipeline from the list. On the pipeline record page, select **Pipeline instances** to view the pipeline instances for the selected pipeline.



Remote instances

ReleaseOps uses update sources to move update sets between instances. A remote instance is the source instance, or update source, where an update set originates. For more information about update sources, see Set up the source instance for an update set.

If you previously worked with update sets and have remote instances already defined, ReleaseOps can use those instances when defining remote instances during guided setup. Otherwise, you can create remote instances during the guided setup process.

Set Automated Test Framework (ATF) code coverage

Define what percentage of code coverage results in Automated Test Framework (ATF) test failure tasks that require remediation.

Before you begin

Role required: sn_releaseops.releaseops_tester or sn_releaseops.releaseops_developer

About this task

The Automated Test Framework (ATF) code coverage threshold is the percentage of code that is covered by ATF test suites. By default, if ATF test suites cover less than 70 percent of the code in a deployment request, ReleaseOps transitions the deployment request to the **Reconciling** state and a test failure task is automatically created. You can adjust the ATF code coverage threshold in your deployment request assessment playbook.

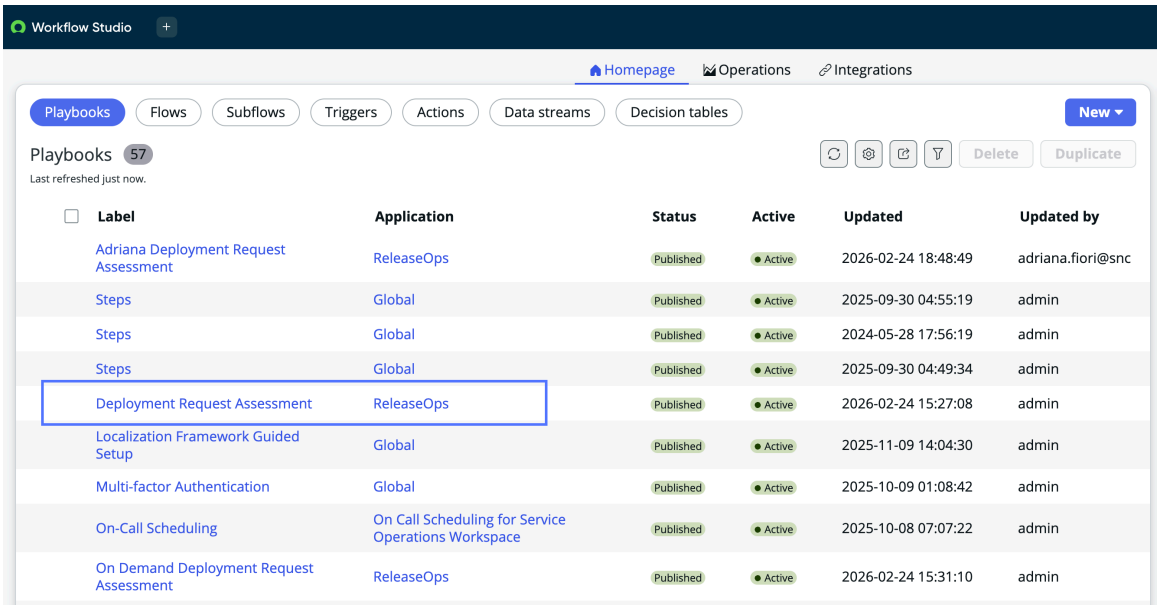
To learn more about ATF code coverage, see Deployment analyzer in ReleaseOps.

Procedure

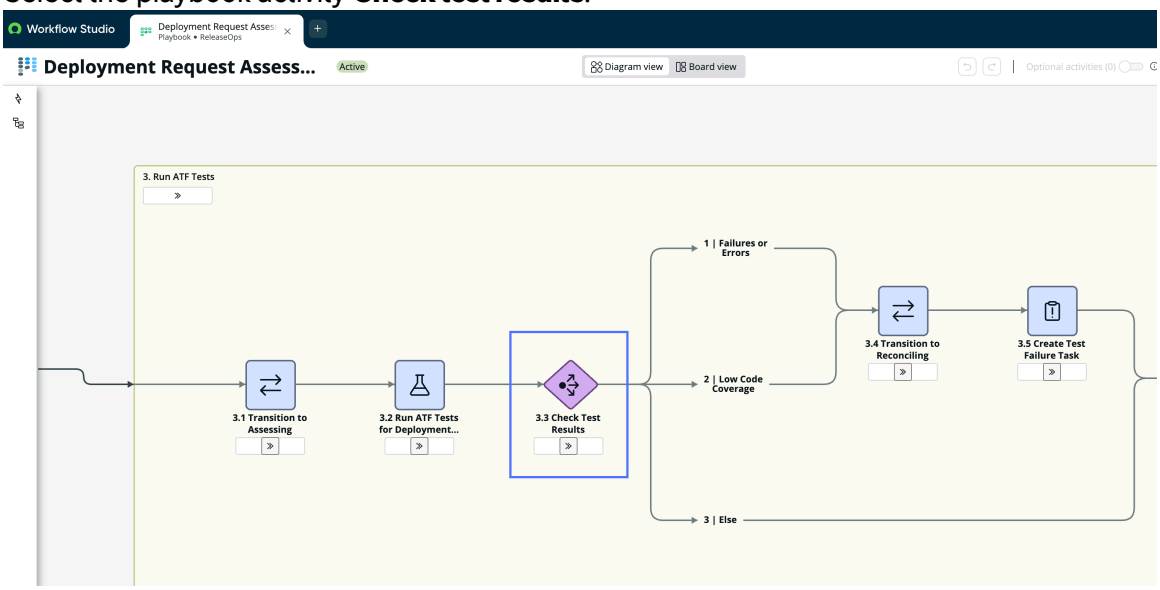
1. Navigate to **All** > **Process Automation** > *Workflow Studio*.
2. Select the deployment request assessment playbook.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Important: Remember to select the playbook that you use for the deployment request assessment stage of your pipeline, not the sample ReleaseOps deployment request assessment playbook. Preserving the ReleaseOps sample playbooks enables you to refer back to them as new features are added during instance upgrades.

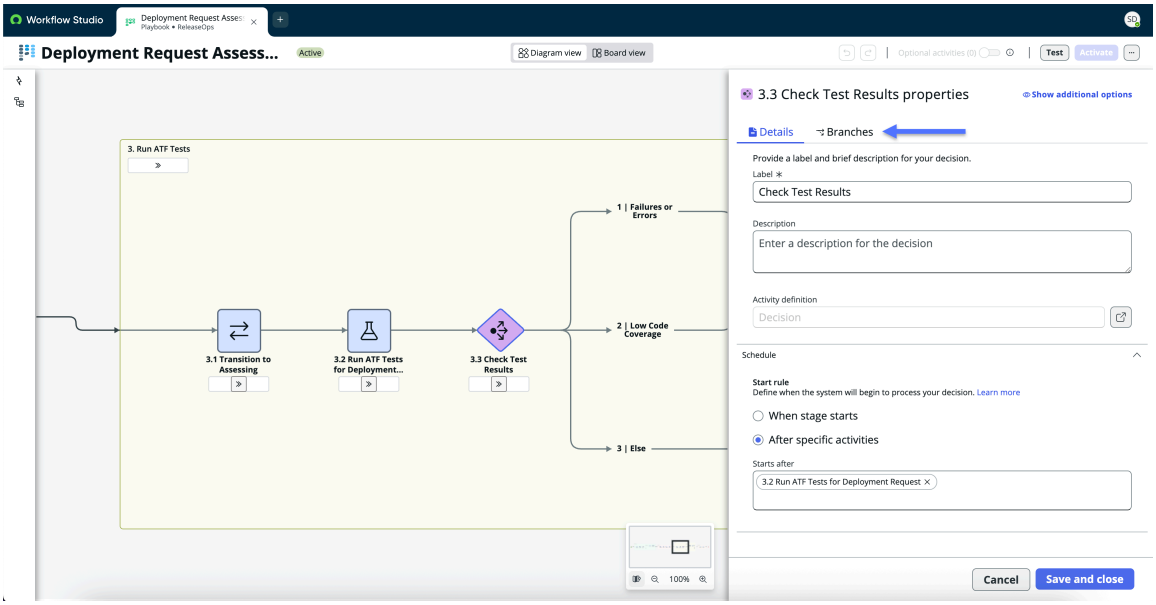


3. Select the playbook activity **Check test results**.

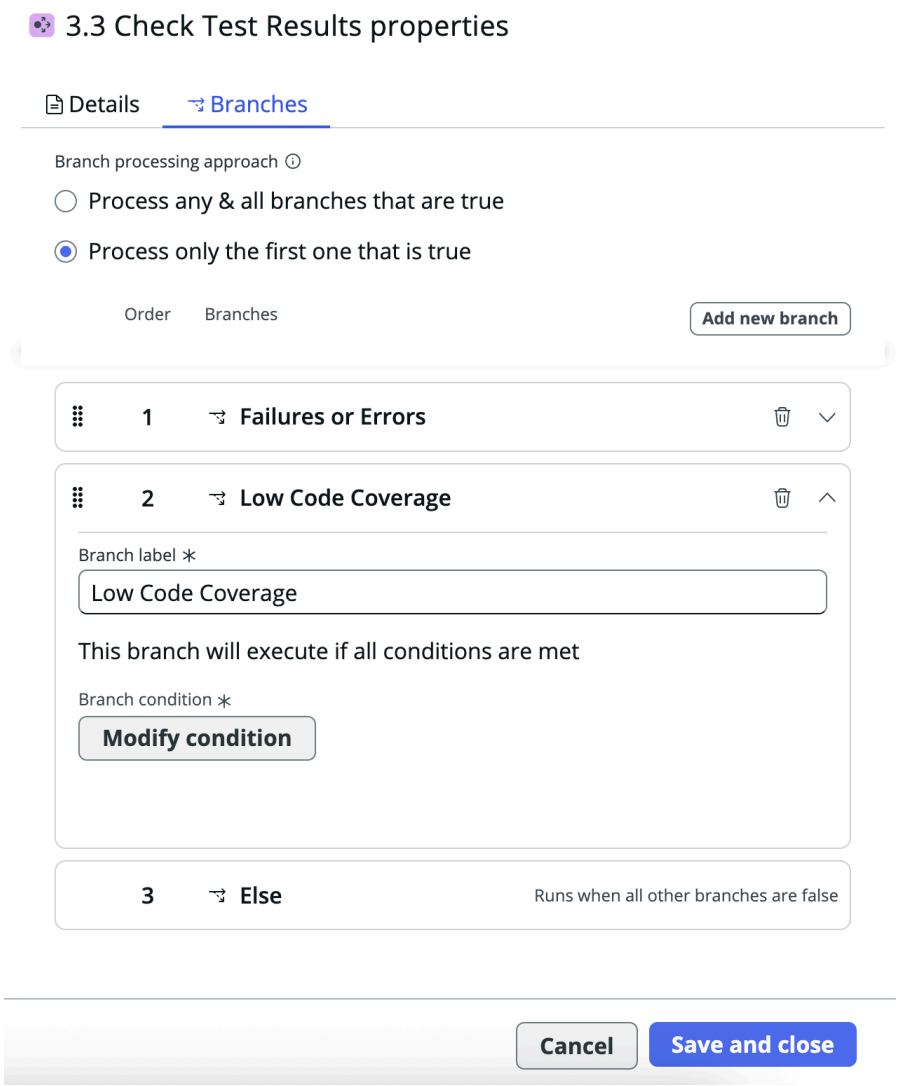


4. In the side panel, select the **Branches** tab.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



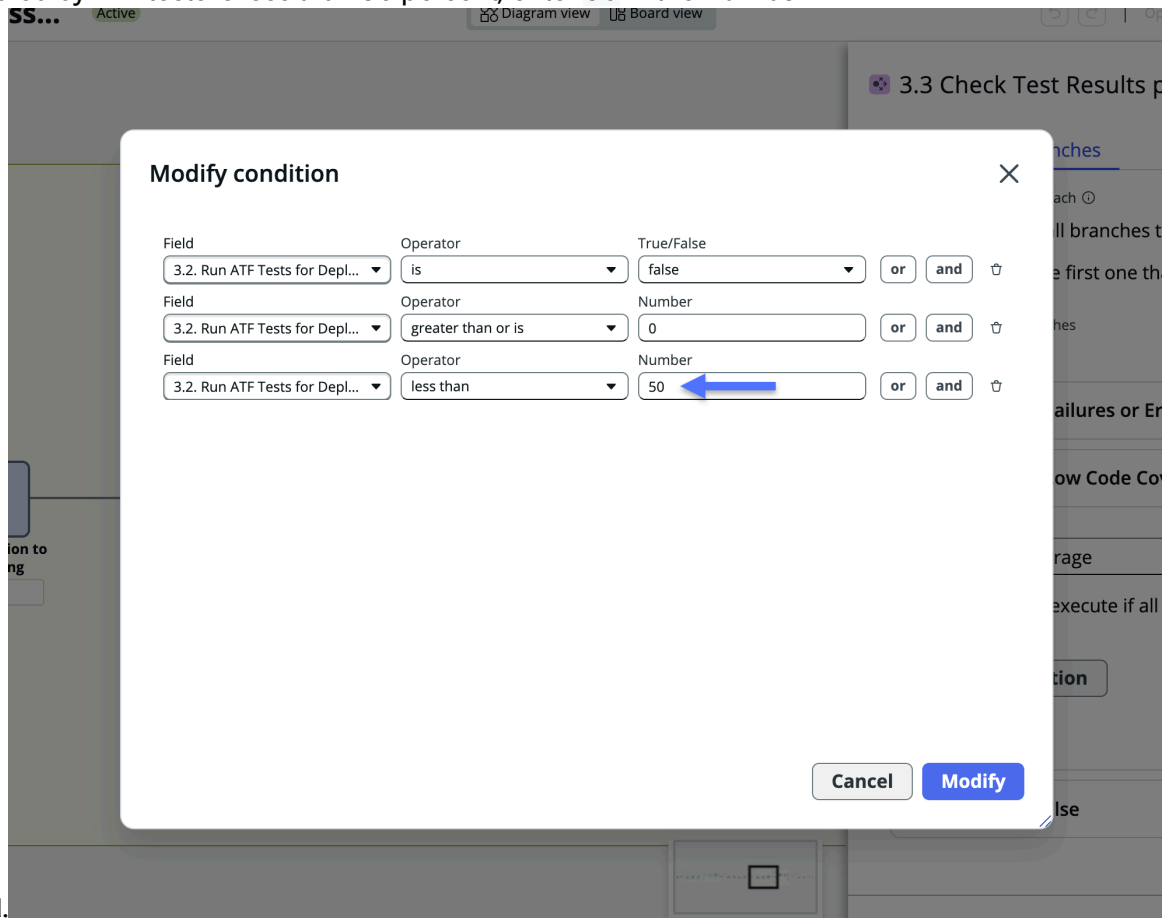
5. Select the low code coverage branch.



6. Select **Modify condition**.
7. Adjust the percentage of code coverage that triggers a test failure task by entering the percentage (as an integer) in the **Number** field.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For example, if you want a test failure to be triggered when the code covered by ATF tests is less than 50 percent, enter 50 in the **Number** field.



8. Select **Modify** to apply your changes to the branch.
9. Select **Save and close**.

Configure release lifecycle documentation AI agent

Configure the release lifecycle documentation AI agent to start automating your app governance tasks.

Before you begin

Note: Depending on your license, you will have access to certain application features, generative AI skills, agentic workflows, and AI agents. For more information, see ServiceNow product tiers.

To configure the release lifecycle documentation AI agent, you must have the ServiceNow Otto® panel turned on and configured. See Activate the ServiceNow Otto panel standard chat for more information.

Role required: admin

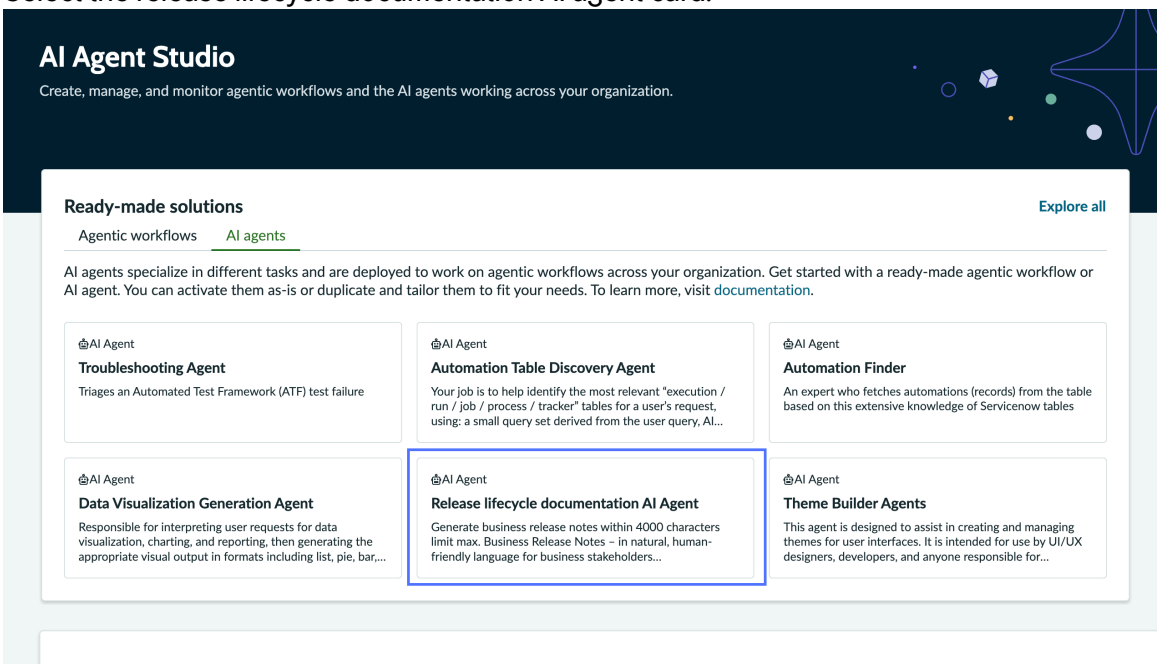
About this task

By default, the release lifecycle documentation AI agent is inactive. To use the release lifecycle documentation AI agent, you must configure the AI agent in AI Agent Studio. For more information about configuring AI agents, see Configure AI agents.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

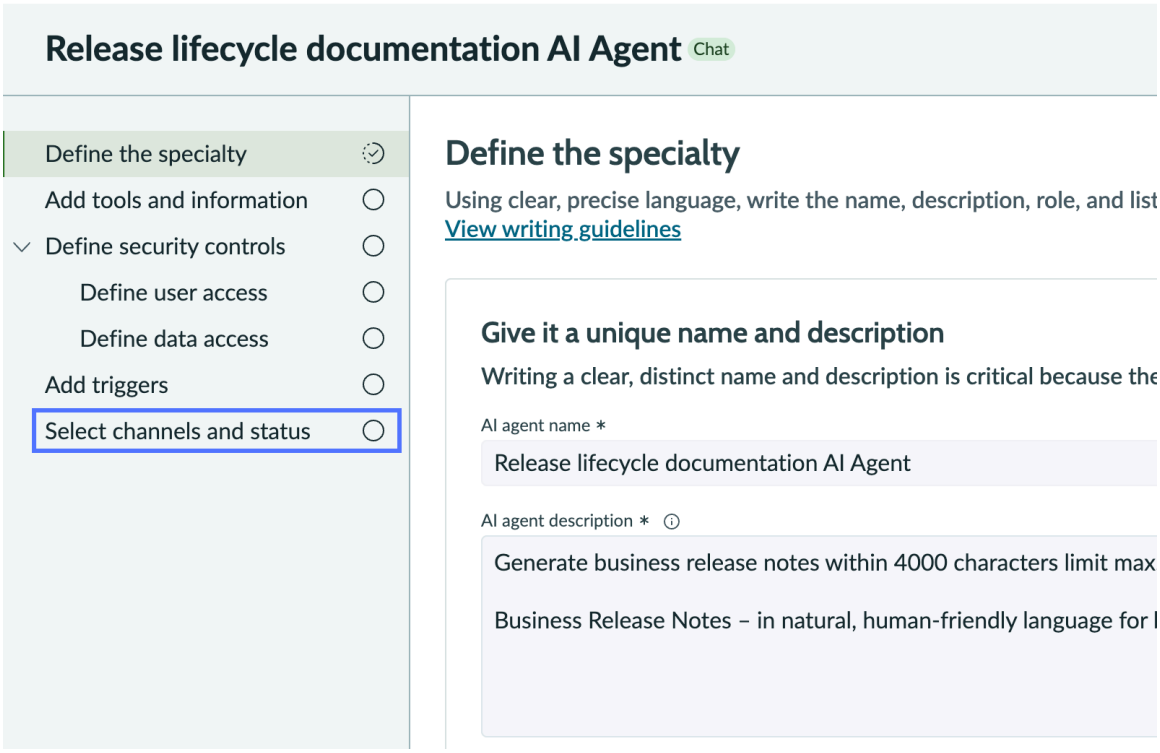
1. Install the ServiceNow Otto for Creator application and all plugins.
   For more information, see Install ServiceNow Otto for Creator.
2. Navigate to **All** > **AI Agent Studio** > **Overview**.
3. Select the **AI Agents** tab.
4. Select the release lifecycle documentation AI agent card.



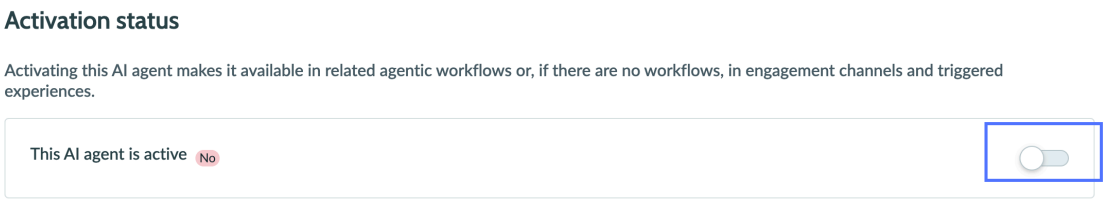
5. **Optional:** Define additional security roles for who can access the AI agent and what data the AI agent has access to.
   By default, using the release lifecycle documentation AI agent requires the update_set_admin and sn_aia.viewer roles. You can add additional role requirements if needed.

   (Optional) For more information about the roles required, see Roles required for using the release lifecycle documentation AI agent. For more information about defining AI agent access, see Define security controls for an AI agent.
6. In the side panel, select **Select channels and status**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



7. Toggle the switch to make the AI agent active.



8. Select **Test**.
   Testing an AI agent enables you to see that it functions the way that you defined it. To learn more, see Manually test the execution of an AI agent.

What to do next

Start using the release lifecycle documentation AI agent to generate update set descriptions and release notes.

- Generate release notes
- Generate an update set description

Roles required for using the release lifecycle documentation AI agent

Learn about which roles are required for using the release lifecycle documentation AI agent.

Role	Description
update_set_admin	Grants access to update set operations that the release lifecycle documentation AI agent uses.
sn_aia.viewer	Grants access to view and interact with AI agent results.
sn_releaseops.release_notes_user	Grants access to generate and view release notes within ReleaseOps.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Role	Description
	Note: The releaseops_admin role also includes the release_notes_user role. Assigning the releaseops_admin role is sufficient for using the release lifecycle documentation AI agent, if the user already needs full ReleaseOps admin access.

Using ReleaseOps to manage deployments

Use ReleaseOps to automate the process of managing your releases and deploying changes to a target instance.

ReleaseOps leverages the automation capabilities of the ServiceNow AI Platform to streamline deployments through a pipeline, empowering teams to manage releases and deploy changes with greater confidence. ReleaseOps enables you to promote changes for deployment easily. Once your changes are promoted for deployment, you can create a deployment request that's tailored to the needs of your release, either scheduled or on-demand. ReleaseOps automates the process of verifying and testing deployment requests, helping to ensure that changes are stable for deployment to target instances. Once the changes are verified and tested, the changes are released to a target instance, often a production instance, automatically.

To learn more about the different components of a ReleaseOps release and the ReleaseOps deployment process, see Exploring ReleaseOps.

The following list outlines the tasks that ReleaseOps can help you to accomplish:

- Release managers can create a release, enabling them to schedule the release date and select which developers and applications the release applies to.
- Developers can promote changes for deployment.
- Developers can create deployment requests for scheduled and on-demand releases, which they attach their update sets to.
- Developers can reconcile deployment tasks, enabling deployment requests to proceed through the pipeline.

Promote an update set for deployment

When you're ready to deploy your changes, promote your update set to begin the deployment process.

Before you begin

Role required: developer or release_admin

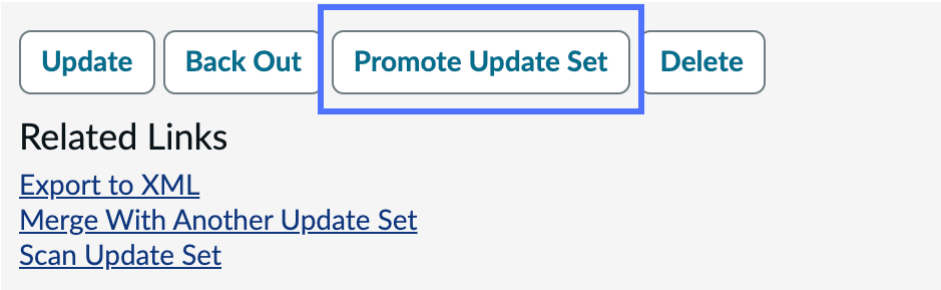
Note: You must have the sn_releaseops.deployment_controller property set on your development instance or the instance from which you intend to promote changes for deployment.

This procedure describes the process of promoting an update set for deployment once an update set has already been created. To learn more about the process of creating an update set, see Working with update sets.

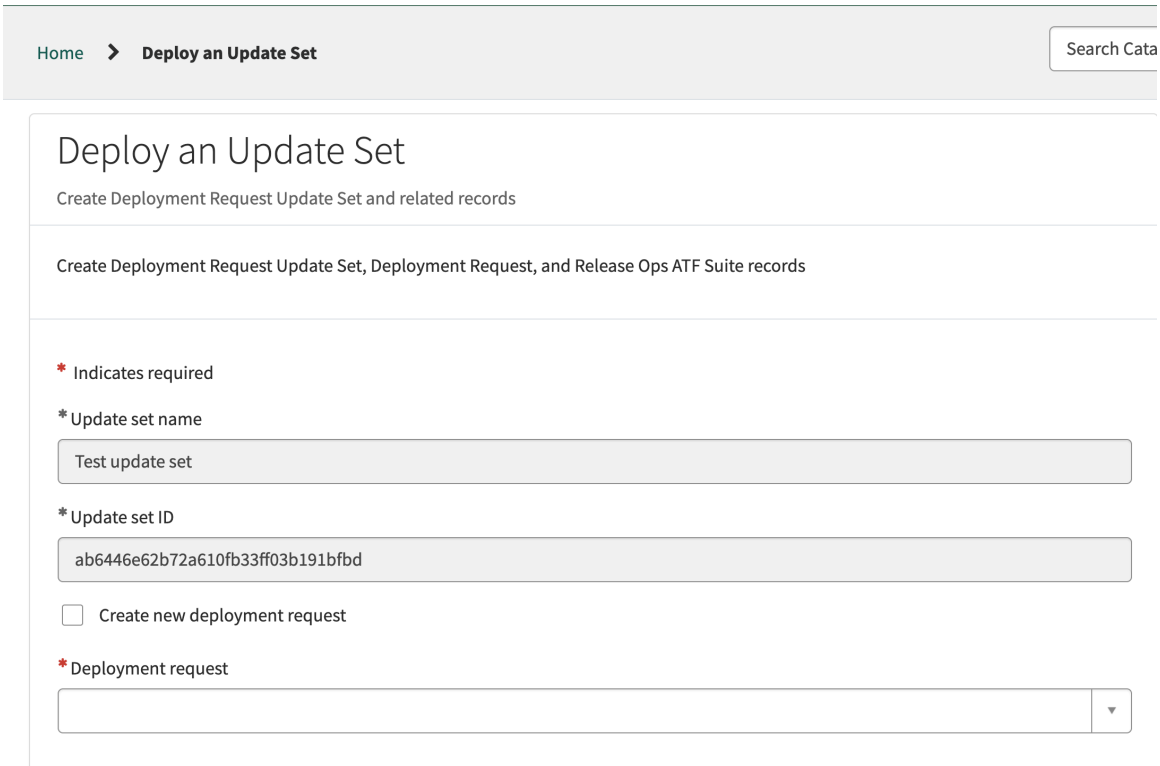
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Open the update set record with the changes that you want to promote for deployment.
   a. Navigate to **All** > **System Update Sets** > **Local Update Sets**.
   b. Select the update set record with the changes that you want to promote for deployment.
2. **Optional:** If the update set **State** field isn't already set to complete, set the state to complete. The **Promote update set** button, which enables you to promote the update set changes for deployment, displays only once the update set **state** field is set to Complete.
   a. Select the **State** field, then select Complete from the list.
   b. Select and hold (or right-click) in the update set record header, then select **Save**.
3. Select the **Promote update set** button.



When you select **Promote update set**, you're automatically redirected to the deployment request form on the production instance, where you can select the deployment request to which the update set changes should be attached.



What to do next

Choose from one of the following options to attach the update set to a deployment request and submit the changes for deployment through the pipeline.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- To attach the update set to an existing deployment request, see Attach an update set to an existing deployment request.
- To attach the update set to a new deployment request for a scheduled release, see Create a deployment request for a scheduled release.
- To attach the update set to a new deployment request for an on-demand release, see Create a deployment request for an on-demand release.

Create a deployment request for a scheduled release

Create a deployment request for a scheduled release to contain your update set and enable your changes to move through the pipeline for deployment.

About this task

You can associate deployment requests for scheduled releases with either a release or a pipeline the release will use. This enables deployment requests to begin assessment independent of the release being created, which can be useful for assessments that require more time. The deployment request must eventually be associated with a release to move to production.

Before you begin

This procedure applies to deployment requests for scheduled releases. To create a deployment request for an on-demand release, see the procedure in Create a deployment request for an on-demand release.

Role required: developer or release_admin

Procedure

1. On the Deploy an update set form, select the check box for **Create new deployment request**. Additional fields appear on the Deploy an update set form, which enable you to define the properties of the new deployment request.
2. In the **Short description** field, enter a description for the deployment request.
3. Assign the deployment request to either an assignment group or a user.
   You can assign a deployment request to both an assignment group and a user, depending on who in your organization might need to be notified about issues. You must select at least one of the assignment options to complete the deployment request form.
   - To assign the deployment request to an assignment group, select the **Assignment group** field, then select the group that you want to assign the deployment request to from the list.
   - To assign the deployment request to a user, select the **Assigned to** field, then select the user that you want to assign the deployment request to from the list.
4. Select the **Source environment** field, then select your development environment from the list.
5. Select either a release or a pipeline to associate the deployment request with.
   - To associate the deployment request with a release, select the **Release** field, then select the release from the list.

     Note: When creating the deployment request, you can select releases that are in **Draft** or **Active** states. However, to start the assessment process, the selected release must be **Active**. For more information, see Release states.

   - To associate the deployment request with a pipeline, select the **Pipeline** field and select the pipeline from the list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

6. **Optional:** Select the **ATF suites** field, then select the tests that you want to run for your deployment request.
   The list of ATF suites is populated by the tests that are available on your production instance. The ATF suites that you select within the deployment request form aren't the only tests that might run for your deployment request. Additional tests might be defined at the pipeline level, depending on the pipeline that your organization has configured.
7. Create your deployment request by selecting **Submit**.
   Your deployment request is created. You can make additional changes to the deployment request as needed, such as adding more update sets.
8. When you're ready to submit your deployment request through the pipeline, select the **Ready to assess** button.

   ⚠ **Warning:** When you select **Ready to assess**, assessment of the deployment request begins immediately, and making additional changes to the deployment request isn't recommended. Only select **Ready to assess** when you're ready to deploy your changes to the pipeline.

   After you select **Ready to assess**, ReleaseOps automatically begins assessing your deployment request. You can see how the assessment is running and any issues that the assessment encountered in the Activity section of the deployment request form. After your deployment request passes the assessment checks, your deployment request moves into the pipeline that your organization has configured.

What to do next

If your deployment request is associated with a pipeline (not a release), you must add a release to the deployment request record before it can move to production. For more information, see Add a release to a deployment request.

Create a deployment request for an on-demand release

Create a deployment request for changes that will deployed with an on-demand release.

Before you begin

Role required: developer or release_admin

Procedure

1. On the **Deploy an update set** form, select the check box for **Create new deployment request**.
   Additional fields appear on the **Deploy an update set** form, which enable you to define the properties of the new deployment request.
2. Enter a short description for the deployment request in the **Short description** field.
3. Assign the deployment request to either an assignment group or a user.
   You can assign a deployment request to both an assignment group and a user, depending on who in your organization might need to be notified about issues. You must select at least one of the assignment options to complete the deployment request form.
   - To assign the deployment request to an assignment group, select the **Assignment group** field, then select the group that you want to assign the deployment request to from the list.
   - To assign the deployment request to a user, select the **Assigned to** field, then select the user that you want to assign the deployment request to from the list.
4. Select the **Source environment** field, then select your development environment from the list.
5. Select the check box for **On demand**.
6. **Optional:** To select a custom pipeline for your deployment request, select the **Pipeline** field, then select the pipeline from the list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

By default, the deployment request deploys through the base system ReleaseOps pipeline.

7. Select **Submit** to create your deployment request.
   Your deployment request is created. You can make additional changes to the deployment request as needed, such as adding additional update sets to the request.
8. When you're ready to submit your deployment request through the pipeline, select the **Ready to assess** button.

> ⚠ **Warning:** When you select **Ready to assess**, assessment of the deployment request begins immediately, and making additional changes to the deployment request isn't recommended. Only select **Ready to assess** when you're ready to deploy your changes to the pipeline.

When you have selected **Ready to assess**, ReleaseOps automatically begins assessing your deployment request. You can see how the assessment is running and any issues that the assessment encountered in the **Activity** section of the deployment request form. Once your deployment request has passed the assessment checks, your deployment request will be moved into the pipeline that your organization has configured.

Attach an update set to an existing deployment request

If you want to attach multiple update sets to a deployment request, you can attach an update set to an existing deployment request on the **Deploy an update set** form.

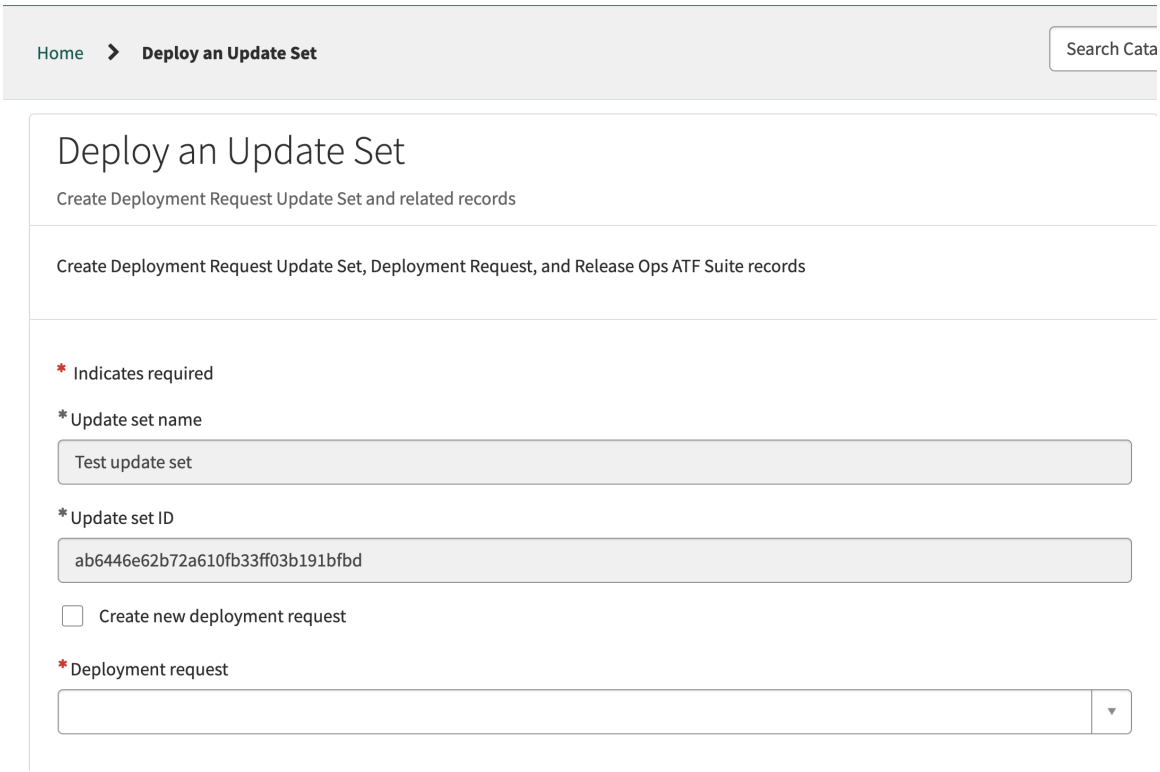
Before you begin

Role required: developer or release_admin

Procedure

1. On the **Deploy an update set** form, select the **Deployment request** field, then select the deployment request that you want to attach the changes to from the list.

> ℹ **Note:** You can only select deployment requests that are in the draft state.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

2. Select **Submit** to create your deployment request.
   Your update set is attached to the existing deployment request. You can make additional changes to the deployment request as needed, such as adding additional update sets to the request.
3. When you're ready to submit your deployment request through the pipeline, select the **Ready to assess** button.

   > ⚠ **Warning:** When you select **Ready to assess**, assessment of the deployment request begins immediately, and making additional changes to the deployment request isn't recommended. Only select **Ready to assess** when you're ready to deploy your changes to the pipeline.

   When you have selected **Ready to assess**, ReleaseOps automatically begins assessing your deployment request. You can see how the assessment is running and any issues that the assessment encountered in the **Activity** section of the deployment request form. Once your deployment request has passed the assessment checks, your deployment request will be moved into the pipeline that your organization has configured.

Create a release

Specify the details for your release, including the target instance that the changes will be deployed to and when the release should occur.

Before you begin

Role required: release_admin

Procedure

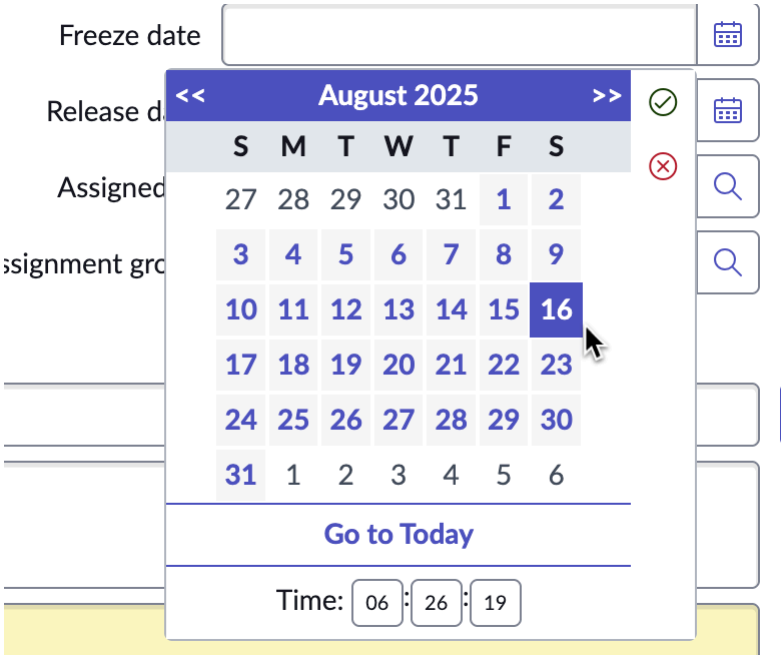
1. On your production instance, navigate to **All** > *ReleaseOps* > **Releases**.
2. Select **New**.
3. On the new release form, select the **Destination environment** field, then select your target destination instance from the list.
   Typically, the target destination instance is your production instance. However, some organizations might have different target destination instances, depending on their pipeline. For more information about the pipeline environments, see Pipelines in ReleaseOps.
4. Select the **Pipeline** field, then select the pipeline that you want the release to use.
5. **Optional:** Enter a short description for the release in the **Short description** field.
6. **Optional:** Enter a description for the release in the **Description** field.
7. **Optional:** Set a freeze date for the release.
   A freeze date is the date by which all changes for the release must be submitted via deployment requests and ready for deployment. Any changes that aren't ready for deployment are deferred at the time of the freeze date.

   a. Select the Select freeze date and time icon .

   b. Select the Previous month icon << or Next month icon >> to locate the month and year for your intended freeze date.

   c. Select your freeze date by selecting the date.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



d. Set the time for the freeze date by entering a time in the **Time** fields.

e. Select the Enter icon .

8. **Optional:** Set a release date for the release.
   A release date is the date by which the changes that apply to the release are deployed to the target destination instance, typically the production instance.

   Note: If your release is associated with a change request, the **Release date** field is automatically populated by the scheduled date in the change request.

   a. Select the Select release date and time icon .

   b. Select the Previous month icon << or Next month icon >> to locate the month and year for your intended release date.

   c. Select the date for your release date by selecting the date.

   d. Set the time for the release date by entering a time in the **Time** fields.

   e. Select the Enter icon .

9. Select **Activate release** to activate the release.
   Once a release is activated, you can select the release as an available option when creating deployment requests for scheduled releases.

Add a release to a deployment request

If you created a deployment request and associated it with only a pipeline, you must add a release to enable the deployment request to move to production.

Before you begin
Role required: developer or release_admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **ReleaseOps** > **Deployment Requests**.
2. From the list, select the deployment request you intend to update.
3. In the **Release** field, select the release the deployment request applies to.
4. Select **Update** or select and hold (or right-click) in the header and select **Save**.

Create a runbook task

Create a runbook task to pause deployment and define the steps required to proceed.

Before you begin

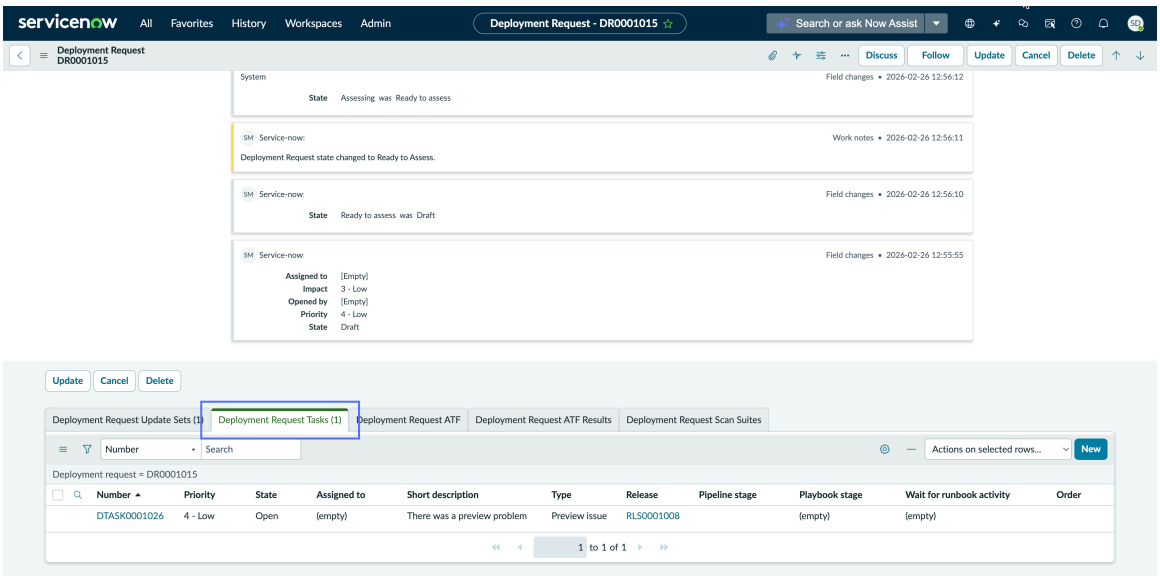
Role required: sn_releaseops.releaseops_tester or sn_releaseops.releaseops_developer

About this task

Runbook tasks enable you to pause playbook progress until a task is resolved. To learn more about runbook tasks, see Runbook tasks in ReleaseOps.

Procedure

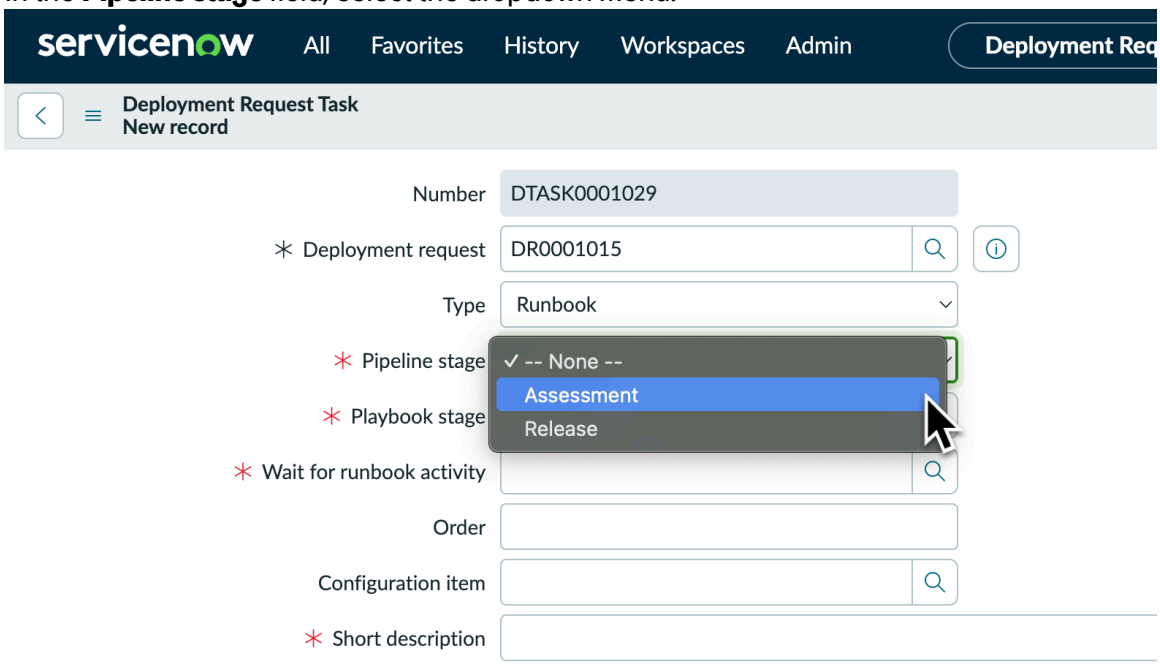
1. On a deployment request record, select the **Deployment request tasks** tab in the Related lists section.



2. Select **New**.
3. On the new deployment task record, select the playbook where you want the runbook task to occur.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. In the **Pipeline stage** field, select the dropdown menu.



b. Select the playbook that you want the runbook task to occur in, either assessment or release.

4. Select the playbook stage where you want the runbook task to occur.

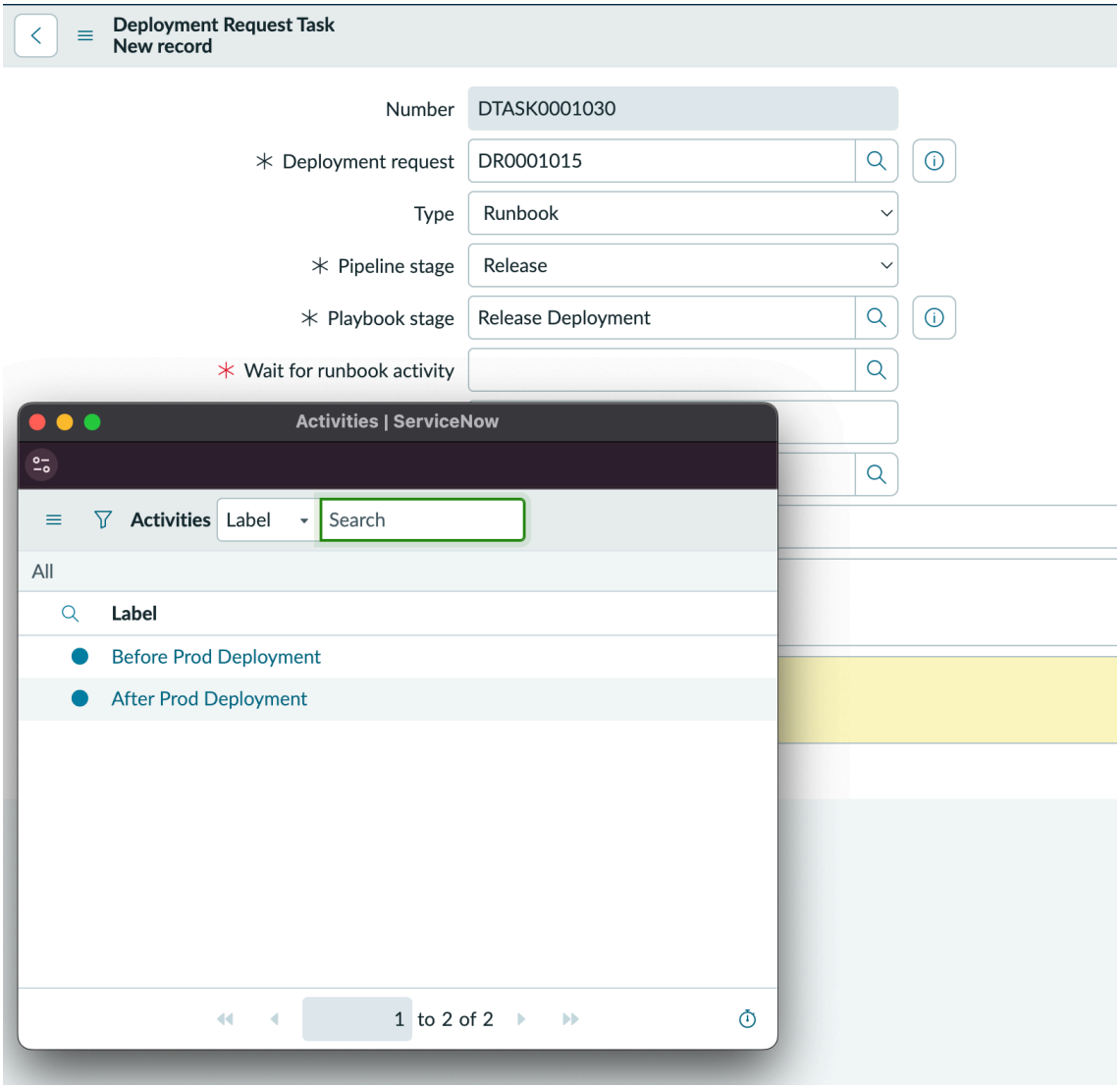
   a. In the **Playbook stage** field, enter the name of the playbook stage or find the available playbook stages using the lookup using list icon.

   b. Select the playbook stage.

5. Define when you want the runbook task to occur, either at the beginning or end of the selected playbook stage.

   a. In the **Wait for runbook activity** field, select the Lookup using list icon.
   A dialog appears that displays the selected playbook stage and the options available for when the runbook task can occur.

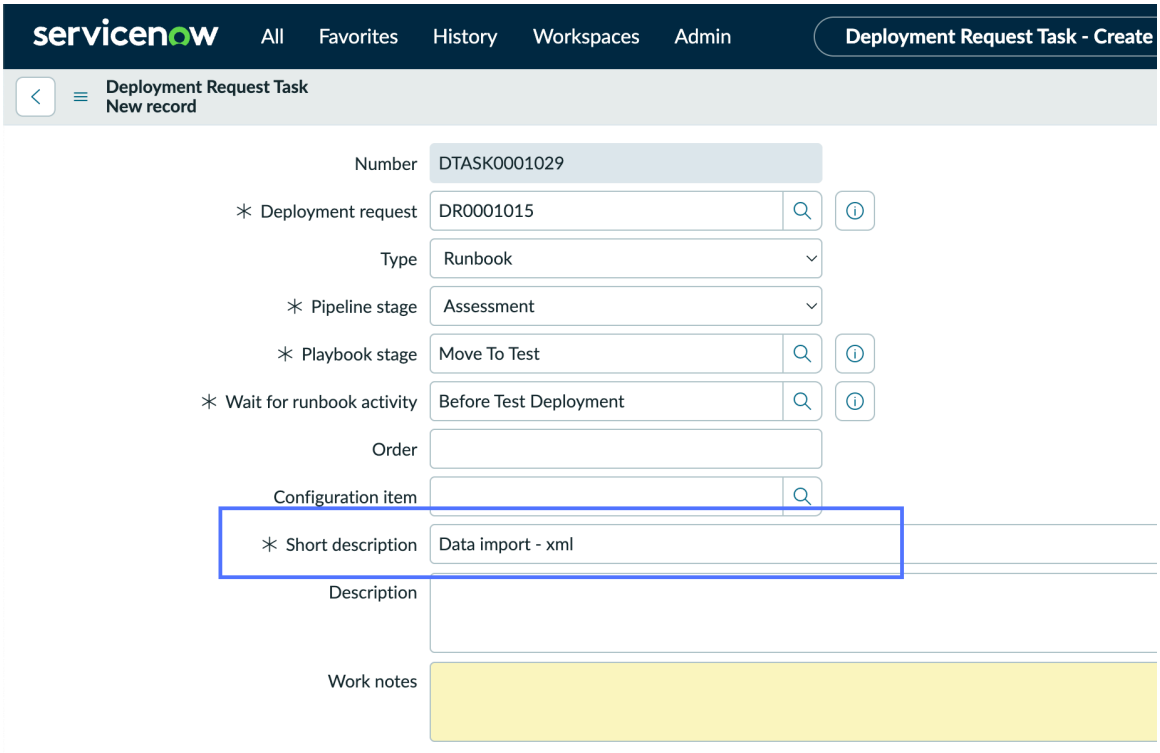
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



b. In the dialog, select either **Before** the playbook stage or **After** the playbook stage.

6. In the **Short description** field, describe the runbook activity that must occur.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



7. Select **Submit**.

Result

The runbook task is added to the playbook stage that you selected. When the playbook is executed, progress will stop until the runbook task is completed.

Reconcile a deployment task

Reconcile a deployment task to enable a deployment request to move to the next stage in deployment.

Before you begin

Role required: developer or release_admin

Procedure

1. Navigate to **All** > *ReleaseOps* > **Deployment tasks**.
2. Select the deployment task that you want to reconcile from the list.
3. On the deployment task record, review the record to see what type of issue the deployment request encountered.
4. Reconcile the deployment task, depending on the type of issue.

Issue	Description
Needs code changes	Additional coding effort is needed. The deployment request is returned to the Draft state for rework.
Rerun tests	ATF re-assessment is needed as the result of insufficient code coverage or incomplete unit tests. The deployment request is returned to the Assessing state, which begins immediately.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Issue	Description
Sign off	The release manager, or someone with the appropriate permissions, must approve an exception for the deployment request to mark it as Ready for Deployment.

5. Once you've addressed the issues in the deployment task, select **Resolve** to enable the deployment request to be reassessed.

Generate release notes

Generate release notes to document app changes and versions over time.

Before you begin

You must have the release lifecycle documentation AI agent turned on in AI Agent Studio. For more information, see .

A release must be in the **Complete** state in order to generate release notes using the release lifecycle documentation AI agent.

Role required: sn_aia.viewer, update_set_admin, or release_notes.admin

Procedure

1. Navigate to **All** > **ReleaseOps** > **Releases**.
2. Select the release from the list.
3. On the release record page, select **Generate release notes**.

> **Important:** Each time you generate release notes using the release lifecycle documentation AI agent, the operation counts as an assist that is tracked by your Now Assist subscription. To track your Now Assist usage, see Monitoring Now Assist usage in Subscription Management.

The release lifecycle documentation AI agent generates the release notes, which might take several minutes. Once release notes have been generated, you can view them in the **Release notes** tab of the release.

4. **Optional:** Edit the release notes manually as needed.
   a. On the release record page, select the Release notes tab.
   b. Select the most current version of the release notes, or the version that you want to edit, by selecting the link in the **Version** column.
   c. Edit the release notes as needed.
   d. Select **Update**.
5. Regenerate the release notes as needed.

> **Note:** If you have an App Engine Management Center (AEMC) subscription, you can regenerate (not generate) release notes using the release lifecycle documentation AI agent inside AEMC.

   a. If you aren't already on a release notes record page, navigate to a release record through either of the following navigation paths.
      - From App Engine Management Center (AEMC), navigate to the **Release management** tab.
      - From ReleaseOps, navigate to **All** > **ReleaseOps** > **Releases**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

b. Select the release from the list.

c. Navigate to the release notes.

- In AEMC, select **View notes**.
- In ReleaseOps, select the **Release notes** tab.

d. Select the most current version of the release notes, or the version that you want to edit, by selecting the link in the **Version** column.

e. Select **Regenerate**.

Generate an update set description

Generate an update set description to quickly communicate the intent and scope of changes.

Before you begin

You must have the release lifecycle documentation AI agent turned on in AI Agent Studio. For more information, see .

Update sets must be in certain states for you to generate descriptions using the release lifecycle documentation AI agent. The following table explains what update set locations and states support description generation.

Update set location	Required state
Local update sets	• In progress • Complete
Remote update sets	• Loaded • Reviewed

You can't generate an update set description for an update set marked as the default update set.

Role required: sn_aia.viewer, update_set_admin, or release_notes.admin

Procedure

1. Navigate to an update set.
   - To navigate to a local update set, navigate to **All** > **System update sets** > **Local update sets**.
   - To navigate to a remote update set, navigate to **All** > **System update sets** > **Retrieved update sets**.
2. Select the update set from the list.
3. On the update set record page, select **Generate description**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Important: Each time you generate a description using the release lifecycle documentation AI agent, the operation counts as an assist that is tracked by your Now Assist subscription. To track your Now Assist usage, see Monitoring Now Assist usage in Subscription Management.

The release lifecycle documentation AI agent generates the update set description, which might take several minutes. Once a description has been generated, the description is automatically added to the update set record.

ReleaseOps reference

Review information about deployment and release states.

ReleaseOps terminology

Before getting started with ReleaseOps, it's important to understand some key concepts used in the application.

deployment analyzer

A service within ReleaseOps that compares changes represented in an update set with the target environment. The deployment analyzer then produces results based on defined rules, such as "contains metadata of type X" or "contains script changes". These results can then be consumed to decide different actions in the deployment process.

deployment request

A record type that represents a portion of a deployment, typically containing a unit of work that was developed and can be deployed independently from other deployment requests. That unit of work is encapsulated in one or more update sets.

deployment request update set

A mapping of an update set to a given deployment request, with some additional data. Although it’s visible to the user for verification purposes, there should be no need to edit these records.

deployment task

As events (such as Automatic Test Framework results or Instance Scan findings) occur, the system can generate deployment tasks. These deployment tasks are assigned to an applicable person (the Developer, Release Admin, and so on) and must be resolved for the deployment or release to continue. Each task presents available resolution options.

pipeline

The flow of a deployment within ReleaseOps, from a developer’s keyboard to the target instance. Pipelines are built on top of playbooks and represent the activities and flows for an application or update to get installed on that target instance. These activities and flows include required scans and tests and potentially other ServiceNow and third-party processes.

release

The release record type and object that represents a full set of changes is destined to be deployed to production or another target environment. The set of changes includes information dictating the schedule of when a deployment should occur. It can contain one or more changes from one or more developers for one or more applications that might be related to each other. A release can be integrated with Change Management.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

update set

A low-level foundational concept that captures and groups configuration changes made in a development instance (such as form customizations, workflows, scripts, or UI elements). Update sets enable these changes to be tracked, bundled, and migrated to new instances. Update sets can contain the instruction for an application install.

Deployment request states

A deployment request might be in one of several different states during the release process.

State	Description
Draft	The deployment request hasn't yet been associated with a release. Draft is the only state in which an update set can be added to a deployment request. A developer keeps their deployment request in Draft state until they believe the work associated with it is complete. It shouldn't be moved into the Ready for Assessment state until the developer is comfortable with it being deployed as-is.
Ready for Assessment	The developer creating the deployment request has determined that it's a functional and complete unit and is ready to be deployed. This state is the triggering condition for the assessment playbook. Immediately after the developer selects Ready to Assess, assessment will begin, running the processes, tests, and checks defined in the release pipeline.
Assessing	Automated Test Framework (ATF) tests are running, instance scans are running, and any Playbooks (PAD) process that will determine the suitability of this deployment request to be deployed is executed. Update sets contained within the deployment request are being moved from the development instance to the test instance. The Assessment phase results in one of two conclusions: • The deployment request is ready, with no findings requiring action (by means of a deployment task). • The assessment surfaced one or more findings that must be reconciled for the deployment request to be ready.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

State	Description
	Deployment tasks will be generated and assigned to the user/group nominated in the request. The outcomes of these tasks are recorded against the request, and the request will be reassessed. This loop continues until the request is ready or canceled.
Reconciling	Open deployment tasks have been generated for action. These tasks must have an outcome for the assessment to be completed and the deployment request to move into the next stage. The sample pipelines included in ReleaseOps will create deployment tasks for preview conflicts and ATF test failures. Some outcomes might involve code changes or additions, updates to test suites or configuration, or approvals or sign-offs. After all open deployment tasks have an outcome, the request is reassessed.
Ready for Deployment	All assessments have been performed, and all reconciliations are complete. No new update sets can be added in this state, and the deployment request is effectively locked. If changes must be made at this point, the deployment request must be canceled. At this point, if the deployment request is an on-demand deployment, it will continue immediately into deployment. Otherwise, it waits for the scheduled time of the release before continuing.
Deploying	A release is in the process of deploying the update sets associated with this deployment request.
Complete	The deployment request was successfully deployed as part of a release.
Failed	The deployment request had a problem that requires manual intervention or resolution to be resubmitted for a release.
Cancelled	The deployment request was manually canceled.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Release states

A release might be in one of several different states during the release process.

State	Description
Draft	The initial state of a release. ReleaseOps doesn't act on any release in this state, regardless of the release schedule set. Draft releases don't appear in a deployment request as a target. This enables the creation of schedules into the future, but they must be manually moved into the Active state before deployment requests can be added and deployed.
Active	The release is open in this state, and waiting for the freeze time to arrive. While in this state, attached deployment requests are actively in the Assessing or Reconciling state. Deployment requests can still be added at this point, or removed (by the developer or release admin).
Preparing	The deployment is frozen. All deployment requests attached to the release are validated to be Ready for Deployment, or else they're deferred. The release ordering has been generated. A release administrator can remove a deployment request. This change of state is derived from the scheduler based on the freeze time. **Note:** If the release is associated with a change management record, the release time is determined by that change management record.
Ready for Release	All deployment requests are finalized. No deployment requests can be added to or removed from the release, and the release has passed all checks and is eligible for deployment. Any manual steps within the pipeline playbook have been completed.
Deploying	The act of finalizing the move of all build artifacts to the destination instance. This change of state is derived from the scheduler, based on the release time.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

State	Description
	Note: If the release is associated with a change management record, the release time is determined by that change management record.
Reconciling	There was a problem with at least one deployment request in the release that requires manual intervention. From the playbook, a deployment task will be created for a preview conflict.
Complete	The release completed without issue.
Cancelled	The deployment of the release was manually canceled before commencement.
Failed	The deployment was either unable to be completed or completed with issues. Common reasons for deployment failure include: • An update set is unable to be retrieved. • An update set that was planned to be retrieved was already retrieved and committed. The commitment is deemed a failure as the defined order of update set deployments has been broken (unless this update set was the first in the ordering). • An update set was committed unexpectedly after retrieval.

Release form

Create a release for the deployment requests to be deployed to target instances.

Field	Description
Release type	Indicates if the release is on-demand or scheduled.
Change request	If a change request is associated with the release, the date on the change request determines the Release date of the release.
State	The state of the release, which will be moved automatically after the release admin selects Activate Release and the release process begins. See Release states.
Release window	Set automatically by the system and shouldn't be changed.
Destination environment	Indicates the deployment instance to which the release will be deployed.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Assigned to	Indicates the person who should receive task notifications related to the deployment request. Either Assigned to or Assignment group is required.
Assignment group	Indicates the group who should receive task notifications related to the deployment request. Either Assignment group or Assigned to is required.
Configuration item	An optional association with a configuration item can be selected. This association doesn't impact the behavior of ReleaseOps.
Pipeline	Indicates the pipeline that the release will run on. This field is editable only while the release is in the Draft state. Once the release leaves Draft, the Pipeline field becomes read-only.
Freeze date	The date on which all deployment requests associated with this release must be in the Ready for Deployment state. All deployment requests must be ready to release and deployed by this date, or else they'll be deferred (and removed from the release). During the freeze window, the system orders the update sets be deployed in the order that they were deployed in the last instance.
Release date	The date on which the release will be pushed to the destination environment.

Deployment Request form

Create a deployment request for the update sets to be deployed to target instances.

Field	Description
Release	The release that the deployment request is associated with. A deployment request can only be attached to one release. **Note:** Deployment requests are deferred if they aren't in the Ready for Deployment state at the freeze date. If the deployment request is deferred, a new release must be associated with the deployment request to reset its state back to Draft and successfully deploy.
Pipeline	The pipeline that the deployment request will be assessed and deployed against, when no release is assigned. Only active pipelines can be selected.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Destination environment	The instance that the deployment request's update sets will be deployed to when no release is assigned. Used to run Code Diff for the deployment request and, for on-demand deployment requests, to determine where the resulting release deploys.
State	The state of the deployment request, which is changed throughout the release process. See Deployment request states.
Source environment	The instance that contains the deployment request and its associated update sets.
Assigned to	Indicates the person who should receive task notifications related to the deployment request. Either Assigned to or Assignment group is required.
Assignment group	Indicates the group who should receive task notifications related to the deployment request. Either Assignment group or Assigned to is required.
On demand	Indicates if the deployment request should be released on demand, instead of scheduled. When On demand is selected, ReleaseOps creates a dedicated release that can only have this one deployment request attached. As soon as the deployment request's state is Ready for Deployment, it will be deployed. No freeze or release dates are followed for an on-demand release.
Configuration item	An optional association with a configuration item can be selected. This association doesn't impact the behavior of ReleaseOps.
Deferred	Checked if the deployment request was deferred due to not being in the Ready for Deployment state by its associated release's freeze date.

Deployment analyzer rules

The deployment analyzer contains five rules that are included by default with ReleaseOps.

Rule name	Rule set	Match Type	Description
Has Code Change	Default	Any	Detects any script change in the update sets.
Has Security ACL Roles	Security	Any	Detects any changes to roles or new roles in the update sets.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Rule name	Rule set	Match Type	Description
Has Security ACLs	Security	Any	Detects any changes to ACLs or new ACLs in the update sets.
Only Catalog Item Changes	Default	All	Verifies that the update sets contain only data from specific catalog tables.
Schema Change	Default	Any	Detects if new schema columns are added. Configured to only trigger if the table has more than 10,000 rows.

Note: Any means that the rule will act as an or search with other rules. All means that the rule will match only if no other rules match.

ServiceNow Otto for App Engine

Use ServiceNow Otto for App Engine to supercharge custom applications with AI agents and skills.

Get started

The ServiceNow AI Platform now brings you a new AI experience with three licensing tiers available:

- Foundation: AI basics to deliver insights
- Advanced: AI to boost productivity across relevant use cases
- Prime: Act autonomously with all AI assets, and create your own

For more information, see ServiceNow product tiers.

Explore



Learn about ServiceNow Otto for App Engine concepts and features.

Install



Install ServiceNow Otto for App Engine.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Use	
 Use ServiceNow Otto for App Engine.	

Important:

- Not all model providers are available for customers with in-country SKUs, and some AI products/features are currently unavailable for in-country customers. For more information, see the KB1584492 article in the Now Support Knowledge Base. Be sure to check for model provider availability updates in future releases.
- Some AI products/features are currently unavailable for customers in the FedRAMP, NSC DOD IL5, or Australia IRAP-Protected data centers, self-hosted customers, or in other restricted environments. For more information, see the KB0743854 article in the Now Support Knowledge Base. Be sure to check for availability updates in future releases.
- Some AI products/features are currently available only for customers in some regions. Be sure to check for availability updates in future releases.
- Some AI products and skills are not available in Regulated Markets. For more information, see KB2593939: Regulated Markets AI Products/Skills Not Available. Be sure to check for availability updates in future releases.

Troubleshoot and get help

- Ask questions and explore other resources for ServiceNow Otto for App Engine in the ServiceNow Community
- Search the Known Error Portal for known error articles
- Contact Customer Service and Support

AI limitations

This application uses artificial intelligence (AI) and machine learning, which are rapidly evolving fields of study that generate predictions based on patterns in data. As a result, this application may not always produce accurate, complete, or appropriate information. Furthermore, there is no guarantee that this application has been fully trained or tested for your use case. To mitigate these issues, it is your responsibility to test and evaluate your use of this application for accuracy, harm, and appropriateness for your use case, employ human oversight of output, and refrain from relying solely on AI-generated outputs for decision-making purposes. This is especially important if you choose to deploy this application in areas with consequential impacts such as healthcare, finance, legal, employment, security, or infrastructure. You agree to abide by ServiceNow's AI Acceptable Use Policy, which may be updated by ServiceNow.

Data processing

This application requires data to be transferred from ServiceNow customers' individual instances to a centralized ServiceNow environment, which may be located in a different data center region from the one where your instance is, and potentially to a third-party cloud provider, such as Microsoft Azure. This data is handled per ServiceNow's internal policies and procedures, including our policies available through our CORE Compliance Portal.

Data collection

ServiceNow collects and uses the inputs, outputs, and edits to outputs of this application to develop and improve ServiceNow technologies including ServiceNow models and AI products. Customers can opt out of future data collection at any time, as described in the AI Opt-Out page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Exploring ServiceNow Otto for App Engine

Explore the AI capabilities, users, benefits, and workflow for using ServiceNow Otto for App Engine to enhance your custom applications with AI.

ServiceNow Otto for App Engine overview

ServiceNow Otto for App Engine is designed to empower developers with the tools needed to add AI to custom applications. Whether you're enhancing an existing custom application with AI capabilities, or creating a new application that features AI capabilities from the start, ServiceNow Otto for App Engine has everything that you need to make developing AI experiences seamless.

With ServiceNow Otto for App Engine, you can leverage the generative and agentic AI capabilities of the ServiceNow AI Platform. For more information about the tools available with ServiceNow Otto for App Engine, see Tools included with ServiceNow Otto for App Engine.

Using ServiceNow Otto for App Engine, you can enhance your custom applications with several AI capabilities: skills, AI agents, and agentic workflows. Starting with version 28.2.4 of ServiceNow Otto for App Engine, you can also use the custom app record summarization skill, which enables you to summarize records within custom apps and tables. Use the following resources to learn more about each of these capabilities and which one might be the best fit for your custom application.

- AI capabilities for enhancing custom applications
- Custom app record summarization skill
- Choosing the right AI capability

ServiceNow Otto for App Engine users

ServiceNow Otto for App Engine users

Users	Tasks
Developers	• Build custom AI agents, skills, and agentic workflows and embed them within existing custom applications. • Configure custom app record summarization to enable summary generation in-product and through conversation. • Automate tasks and work within custom app workflows through AI skills, agents, and agentic workflows.
Requesters and fulfillers	Leverage AI agents and skills within custom applications at runtime to complete tasks.

ServiceNow Otto for App Engine workflow

The following list outlines the developer workflow for creating and embedding a custom skill within an application with ServiceNow Otto for App Engine:

1. Identify a task to automate as a skill within an existing or new custom application.
2. Engineer a skill using AI Skill Kit.
3. Choose a large language model (LLM), define inputs, and compose prompt text for the skill.
4. Add tools to the skill, such as pulling in data or automating a task within a workflow or subflow.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

5. Test and validate the skill using AI Data Kit to optimize the correctness of the skill.
6. Activate and embed the skill within a custom application using an AI agent, UI action, flow action, or the ServiceNow Otto panel.

The process for building and embedding an AI agent or an agentic workflow within a custom application is similar. Developers use AI Agent Studio to create, define, and add tools to the AI agent. They can then deploy the AI agent for use in the ServiceNow Otto panel or within an agentic workflow.

For more information about creating skills, AI agents, and agentic workflows with ServiceNow Otto for App Engine, see Using ServiceNow Otto for App Engine.

ServiceNow Otto for App Engine benefits

ServiceNow Otto for App Engine benefits

Benefit	Users
Modernize existing custom applications with generative AI agents and skills, including the custom app record summarization skill..	Developers
Create skills, AI agents, and agentic workflows capable of handling complex, non-deterministic decision logic.	Developers
Improve productivity and efficiency by leveraging AI agents and skills to complete tasks.	Requesters and fulfillers
Understand the contents of a record quickly using the custom app record summarization skill.	Requesters and fulfillers
Offload manual, repetitive tasks to AI agents and agentic workflows.	Fulfillers

What to explore next

To learn more about ServiceNow Otto for App Engine, see Installing ServiceNow Otto for App Engine and Use ServiceNow Otto for App Engine.

AI capabilities for enhancing custom applications

Learn about the AI capabilities available with ServiceNow Otto for App Engine that you can use to enhance custom applications.

There are several types of AI capabilities that you can add to custom applications:

- Skills
- AI agents
- Agentic workflows

The following sections describe what each capability is and how you can use it to enhance a custom application. To learn about which AI capability might be best for your custom application use case, see Choosing the right AI capability.

Skills

On the ServiceNow AI Platform, a skill is a generative AI capability that contains instruction on how to complete a specific task for a use case, such as generating a summary.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

There are several important components of a skill.

- The input is the data that the skill analyzes to generate an output. Typically on the ServiceNow AI Platform, inputs are records in tables.
- The prompt is the instruction to the underlying large language model (LLM) that tells the skill what to do with the input. Prompts define the task, tone, and structure of the output.
- The output is the generated response from the LLM based on the prompt and input. Outputs can be plain text, structured data, or even formatted responses depending on how the skill is configured.

Skills also contain activation methods, which define how the skill can be accessed on the ServiceNow AI Platform. Skills can be called conversationally in the ServiceNow Otto® panel, through UI actions such as buttons, and through conversations with Virtual Agent. To learn more about activating skills, see Activate a skill.

Note: Some skills might need to be reviewed and approved by a data steward before you can activate them. To learn more, see Governing AI assets.

Platform skills vs. custom skills

With ServiceNow Otto for App Engine, you have access to several kinds of skills: Platform generative AI skills and custom skills. While you might be able to use both kinds of skills in custom applications, the following table provides more information about each skill type and the ease of using the skill in custom applications.

Differences between Platform generative AI skills and custom skills

Skill type	Description	Ease of use in custom applications	More information
Platform generative AI skills	Preconfigured skills that are designed for existing use cases,such as document extraction and dashboard and visualization export	Must be duplicated and reconfigured significantly for use within custom apps	Now Assist skills in the Platform workflow
Custom skills	Skills that you create custom for your use case with AI Skill Kit	Can be designed to work within custom apps during the creation process, when you define the skill inputs and outputs	• General guidelines for AI Skill Kit • Create a skill

ServiceNow Otto for App Engine custom app record summarization skill

Starting with version 28.2.4 of ServiceNow Otto for App Engine, you can use the custom app record summarization skill. The skill is a template skill available with ServiceNow Otto for App Engine that enables you to generate AI summaries of records within custom apps and tables. For more information, see Custom app record summarization skill.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

AI agents

An AI agent is a generative AI capability that contains a set of large language model (LLM) instructions with the tools to perform specific tasks. On the ServiceNow AI Platform, AI agents are often part of agentic workflows and can be orchestrated to perform complex tasks. More information about agentic workflows is included in the following section.

There are several important components of an AI agent.

- The AI Agent role defines the capabilities and responsibilities for the AI agent. Roles enable the AI agent to perform its required actions.
- The list of steps are the instructions are sent to the large language model (LLM) that describe the necessary steps the AI agent follows while carrying out its role.
- The tools are the additional resources, such as skills, and data sources that you can provide your AI agent to accomplish its role.
- The trigger defines how the AI agent is activated.
- The availability determines how your AI agent is displayed, either in the ServiceNow Otto® panel or Virtual Agent.

For more information about AI agents, see AI Agent Studio overview.

Agentic workflows

An agentic workflow contains a set of large language model (LLM) instructions with one or more AI agents that can execute an objective. Agentic workflows are tailored to achieve specific business outcomes, such as classifying tasks or processing emails for tasks.

An agentic workflow is controlled by an AI Agent Orchestrator that coordinates the flow of work between AI agents. Additionally, an AI Agent Communicator helps to facilitate communication between the AI Agent Orchestrator and AI agents in the workflow.

For more information about agentic workflows, see General guidelines for writing prompts for agentic AI.

Platform AI agents and agentic workflows vs. custom AI agents and agentic workflows

With ServiceNow Otto for App Engine, you enhance custom applications with Platform and custom AI agents and agentic workflows. Similar to adding generative AI skills, configuring Platform AI agents and agentic workflows to fit custom applications often takes more time than just building custom AI agents and agentic workflows from the start. The following table provides additional information about Platform and custom AI agents agentic workflows.

Differences between Platform AI agents and agentic workflows vs. custom AI agents and agentic workflows

		Ease of use within custom applications	More information
Platform AI agents and agentic workflows	Preconfigured AI agents and agentic workflows that are designed for existing use cases, such as incident resolution	Must be duplicated and modified significantly for use in custom applications	• Explore AI agents • Platform agentic workflows

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Differences between Platform AI agents and agentic workflows vs. custom AI agents and agentic workflows (continued)

		Ease of use within custom applications	More information
Custom AI agents and agentic workflows	AI agents that you create custom for your use case using AI Agent Studio	Can be designed to work within custom apps during the creation process, when you define the AI agent and agentic workflow	• General guidelines for writing prompts for agentic AI • Create an AI agent • Create an agentic workflow

Custom app record summarization skill

The ServiceNow Otto for App Engine custom app record summarization skill can generate summaries for the contents of records in custom applications and tables.

Record summarization overview

Starting with version 28.2.4 of ServiceNow Otto for App Engine, you can use the custom app record summarization skill. The skill uses generative AI to generate summaries for records in custom tables and applications. Record summarization enables you to learn important details and context about a record quickly, helping to save you time and reduce repeated efforts.

The custom app record summarization skill is configurable, enabling you to define the data, experience, and access for the summaries that it generates. You can specify the tables and fields that you want to summarize. You can also include additional data sources, such as emails and attachments, to use when generating summaries. You can then define which users have access to the summaries. Learn more about the custom app record summarization skill and how you can configure it to work for your custom applications in the following sections.

Data sources

The custom app record summarization skill can generate summaries for records in any table that's created within a custom application. The application can be global or scoped.

You can configure the skill to summarize as many tables as needed for your custom application. During the configuration process, you define which table or tables you want the skill to summarize. For each table that you select, you must define the fields that should be included in the summaries and provide additional context about the fields that you selected.

For example, let's say that you want to summarize the records in a table for your paid time off (PTO) request application. You select the table that stores details about each employee's PTO request, such as the dates that the employee is requesting time off from work and the approval status of the request. When you select the fields that you want to include in the AI-generated summaries, you must enter a description for each field. Field descriptions provide additional information and context about the purpose of the field, helping the skill to generate summaries that are more accurate. The following table illustrates how to write field descriptions for generative AI-powered summaries.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example fields and field descriptions for configuring the custom app record summarization skill

Field in the PTO request table	Field description
PTO start date	`Date that the employee's PTO begins. Date is in MM-DD-YYYY format.`
PTO end date	`Date that the employee's PTO ends. Date is in MM-DD-YYYY format.`
Approval status	`Field that indicates whether the PTO request has been approved, denied, or hasn't been reviewed yet.`

Additional data sources

The custom app record summarization skill can include additional data sources when generating a summary of a record, such as:

- Related tables
- Relationships
- Emails
- Attachments

For each related table or relationship that you add, you must provide the fields and field descriptions that you want included in the generated summaries. Activities, such as emails and attachments, don't have fields that you can select and configure. If you select either email or attachment as an additional data source to include in a summary, the skill attempts to fetch that data at the time that the summary is generated. If an email or attachment is present within the record, the skill includes that information in the summary that it generates. Otherwise, the summary is generated without the email or attachment.

Important: Record summarization that includes an attachment has a greater assist usage. To track your AI usage, see Monitoring Now Assist usage in Subscription Management.

Display options

You can choose where you want record summarization to be available to your application users during the skill configuration process. The display options that you can select are: in-product, ServiceNow Otto® panel, or both.

Note: The display options that you choose applies to every application that uses the ServiceNow Otto for App Engine custom app record summarization skill.

In-product

When selected, AI-generated summaries are displayed on forms and workspaces.

For more information about adding record summarization to a workspace, see the following resources:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Customize UI Builder pages using components
- Components documentation for the AI summary card
- UI Builder setup documentation for the AI summary card

ServiceNow Otto® panel

When selected, AI summaries are accessible through chat in the ServiceNow Otto® panel. If you don't see this option when configuring the skill, you must activate the ServiceNow Otto® panel. For more information, see Turn on the Now Assist panel.

Access

You can define which users and groups have access to generate AI summaries. During configuration, you can add roles and conditions that restrict when and for whom the skill is available. You can also customize whether certain roles have access to a particular display option.

Tools included with ServiceNow Otto for App Engine

Learn about the tools and applications included with ServiceNow Otto for App Engine.

ServiceNow Otto for App Engine enables users to use the generative and agentic AI capabilities of the ServiceNow AI Platform. With ServiceNow Otto for App Engine, you can create and activate custom skills using AI Skill Kit, build custom AI agents using AI Agent Studio, generate synthetic data for testing skills and AI agents using AI Data Kit, and more.

The following table highlights some of the ServiceNow AI Platform tools and applications that you can use to enhance custom applications with ServiceNow Otto for App Engine.

Note: Check your entitlements to determine whether you have access to the following tools and applications.

Tools available with ServiceNow Otto for App Engine

Tool	Description	Resources for additional information
AI Agent Studio	Create flexible, adaptive AI agents for App Engine use cases.	AI Agent Studio overview
AI Control Tower	Govern AI assets.	AI Control Tower
AI Skill Kit	Develop custom skills that can be used in App Engine applications.	AI Skill Kit
AI Data Kit	Build and maintain datasets to evaluate your custom skills.	AI Data Kit

For a complete list of the tools and skills in ServiceNow Otto for App Engine, see the product listing in the ServiceNow Store.

Choosing the right AI capability

Choosing the right AI capability for your custom application is important. Learn about the differences between skills, AI agents, and agentic workflows, use cases for each, and when to choose one capability over another.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

When adding AI capabilities to your custom applications, the capability that you choose depends on several factors, including the complexity of the workflow and the degree of human oversight required for the tasks.

When to add skills

Skills are suited for specific, well-defined tasks, such as summarizing a record or classifying a request. Because skills can be reused in different contexts and applications, skills are ideal for situations where the input and output are predictable and don't require complex decision-making. Skills also provide a high degree of human oversight and control, which makes them a practical way to introduce AI capabilities within your organization's workflow.

When to add AI agents

AI agents are suited for more complex, specific tasks that involve some degree of reasoning, planning, and responding. For example, an AI agent for a retail company app could be trained to identify recurring issues with product merchandise and escalate issues to the appropriate teams responsible for resolution. Because AI agents can contain multiple tools and skills, they are useful in situations where you want an entirely automated solution for a task or to offload manual work. AI agents involve less human oversight than skills, while still providing transparency and human intervention when necessary.

When to add agentic workflows

Agentic workflows are suited for complex, knowledge-intensive tasks where the solution requires human oversight at several steps in the process. Because agentic workflows can complete processes from end-to-end, agentic workflows are useful for processes that are low risk, such as handling routine approvals or requests.

Configuring ServiceNow Otto for App Engine

Learn about the configuration process for ServiceNow Otto for App Engine.

ServiceNow Otto for App Engine configuration overview

To start using ServiceNow Otto for App Engine, you must install ServiceNow Otto for App Engine on your instance. For more information, see Installing ServiceNow Otto for App Engine.

To use the custom app record summarization skill, you must activate and configure the skill in the AI Admin Hub. For more information, see Activate the custom app record summarization skill and Configure the custom app record summarization skill.

Installing ServiceNow Otto for App Engine

You can install ServiceNow Otto for App Engine if you have the admin role.

Before you begin
Role required: admin

The ServiceNow Store enables you to download core products and applications. A product contains one or more applications that are licensed as a group. For example, ServiceNow Otto for App Engine is a product that contains numerous tools and skills.

Procedure

1. Open the ServiceNow Store website.
2. In the **Search** field, enter `ServiceNow Otto for App Engine` and select enter or return on your keyboard.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

3. Select the *ServiceNow Otto for App Engine* tile.
4. Read through the information about ServiceNow Otto for App Engine.
5. Select **Buy**.
6. Log in with your ServiceNow user ID.

Activate the custom app record summarization skill

Before you can generate AI-powered summaries for custom application records, activate the custom app record summarization skill in the *AI Admin Hub*.

Before you begin
Role required: admin or sn_generative_ai.nsa_admin

Procedure
1. Navigate to **All** > *AI Admin Hub* > **Skills**.
2. On the Now Assist Skills page, select App Engine from the side panel.
3. Select **Activate skill** in the custom app record summarization card.

Result
The custom app record summarization skill is active.

What to do next
Configure the skill to generate summaries for tables in your custom applications. See Configure the custom app record summarization skill for more information.

Configure the custom app record summarization skill

Configure the custom app record summarization skill to define the records and outputs for ServiceNow Otto for App Engine summaries.

Before you begin
Role required: admin

You must have the custom app record summarization skill activated to configure the skill. For more information, see Activate the custom app record summarization skill.

Procedure
1. Navigate to **All** > *AI Admin Hub* > **Skills**.
2. On the *AI Admin Hub* Skills page, select App Engine from the side panel.
3. In the custom app record summarization card, select the actions icon  .
4. Select **Edit**.
5. On the Select an app page, choose the app containing the data that you want to summarize by selecting the check box next to the app name.
6. Select **Save and continue**.
7. Choose the table that contains the data you want to generate AI summaries for.
   You can generate summaries for records in most custom tables on the ServiceNow AI Platform. However, some tables are restricted. For more information, see Custom app record summarization skill.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. On the Choose data source page, select the **Base input table** field.

b. Select the table containing the data you want to summarize.
When you select a table from the list, the required fields from the table are automatically included in the AI summary.

c. **Optional:** To include additional fields from the table in the summary, select **+ New base input field** and select the field that you want to add.

d. For each field in the table that you want included in the AI summary, enter a description and context for the field in the **Field description** field.

e. **Optional:** To add related data sources that you want included in summaries, such as related tables, select **+ New data source**, then select the additional data source from the list.
For more information about the kinds of data sources that can be included in AI summaries, see Custom app record summarization skill.

f. **Optional:** To add more tables, select **+ New input table**. Then repeat the process in steps a through e to define the fields that you want to be included in the summary.

g. When you're finished, select **Save and continue**.

8. Test the summaries that are generated.

> **Important:** Each time you test the output of generative AI skills, the operation counts as an assist that is tracked by your subscription. To track your AI usage, see Monitoring Now Assist usage in Subscription Management.

a. On the Test summary output page, in the **Base input table** field, select the table that you want to test from the list.

b. In the **Additional instructions for LLM** field, enter additional information about the context and purpose of the table.

c. In the **Record** field, select the record that you want included in the AI summary.

d. Select **Run test**.

e. Review the AI-generated summary in the Result panel.

f. **Optional:** Test each table that you want to summarize and adjust the instructions and field descriptions as needed to improve the summary quality.

g. When you have completed testing, select **Save and continue**.

9. Define which users have access to the skill.

a. **Optional:** To adjust which roles have access to the skill, select the edit icon ( ).

b. Follow the instructions outlined in Configure security controls for a skill to control who has access to the skill.

c. When finished, select **Save and continue**.

10. Choose the display option that determines how users access the skill.
For more information about display options, see Custom app record summarization skill.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. Select the toggle on icon ( ) for the display option that you want to enable.

b. To enable access for certain roles, select the expand icon ( ) for the display option that you want to define access to.

c. In the **user roles** field, enter the name of the role that you want to grant access to, then select the role from the list.

> **Note:** You can grant access to multiple roles. Select all roles that you want to grant access to from the list.

d. When finished, select **Save and continue**.

11. Review and verify that the skill details are correct.
12. When you're ready to activate the skill, select **Done**.

Result

The custom app record summarization skill is now ready for use in the custom app that you defined.

What to do next

To use the custom app record summarization skill, see Using ServiceNow Otto for App Engine.

Using ServiceNow Otto for App Engine

Enhance custom applications with generative and agentic AI capabilities.

With ServiceNow Otto for App Engine, you can enhance your custom applications with several AI capabilities: skills, AI agents, and agentic workflows. To learn more about the types of AI capabilities that you can implement with ServiceNow Otto for App Engine, see AI capabilities for enhancing custom applications.

Certain app workflows might be well suited for AI agents and agentic workflows. Other workflows might be best handled with skills. To learn about which AI capability might be best for your use case, see Choosing the right AI capability.

Implementing skills

The following list outlines the process for implementing a skill within a custom application:

1. Create a skill or duplicate and reconfigure a Platform generative AI skill for your app's use case.
2. Create a prompt to choose what skill inputs to use and the type of tool.
3. Test the prompt to verify that you're seeing the expected prompt results before the skill is activated.
4. Evaluate the prompt to assess the effectiveness of your skill prompts.
5. Finalize and publish the skill to enable a ServiceNow Otto admin to activate it.
6. Activate the skill, enabling you to trigger the skill within the UI.
7. Use the skill in your custom application.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: Some skills might need to be reviewed and approved by a data steward before you can activate them. To learn more about approvals for custom skills, see Governing AI assets.

Implementing AI agents and agentic workflows

The following list outlines the process for implementing an AI agent and agentic workflow within a custom application:

1. Create an AI agent or duplicate an AI agent and modify it for your app's use case.
2. Add a tool to the AI agent to enable different functionalities and help your AI agent to achieve their objective.
3. Test the AI agent to see that it functions the way that you defined it.
4. Create an agentic workflow or modify an agentic workflow so that AI agents can coordinate to solve complex problems.
5. Test the agentic workflow to analyze its performance while it runs the instructions that you defined.
6. Evaluate the agentic workflow to monitor performance and evaluate it against different benchmarks.
7. Use the AI agent or agentic workflow in your custom application.

Summarize a record in-product using ServiceNow Otto for App Engine

Quickly understand the contents of a record in a custom application within Core UI or a custom workspace.

Before you begin

Role required: admin

Verify that the custom app record summarization skill is activated and configured. For more information, see Activate the custom app record summarization skill.

The following procedure describes how to summarize a record from a custom application in-product in both Core UI and in a custom workspace. To learn how to add a record summarization card to your custom workspace, see the following resources:

- Customize UI Builder pages using components
- Components documentation for the AI summary card
- UI Builder setup documentation for the AI summary card

You can also summarize the contents of a record through chat in the ServiceNow Otto panel. For more information, see Summarize a record through chat using ServiceNow Otto for App Engine.

Procedure

1. In Core UI or a custom workspace, open the record that you want to summarize.
2. Select **Summarize**.

   Note: Generating and displaying the summary may take several seconds.

3. **Optional:** When you're finished summarizing a record, you can copy the summary, expand or collapse the summary, or provide feedback about the summary.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Procedure
Expand or collapse the summary	Select the expand card icon ( ) to view the complete summary or the collapse card icon ( ) to view a collapsed summary.
Provide feedback about the summary	If you think that the summary was helpful, se lect the helpful icon ( ). If you think that the summary wasn't helpful, select the not helpful icon ( ). **Note:** This feedback improves the generative AI model and can help to im prove future versions of this skill.
Copy the record summary	If you want to reuse the summary, select the copy to clipboard icon ( ).

Summarize a record through chat using ServiceNow Otto for App Engine

Quickly understand the contents of a record through a conversation with ServiceNow Otto.

Before you begin

Role required: admin

Verify that the custom app record summarization skill is activated and configured. For more information, see Activate the custom app record summarization skill.

The following procedure describes how to summarize a record through chat in the ServiceNow Otto panel. You can also summarize a record in-product within Core UI or a custom workspace. For more information, see Summarize a record in-product using ServiceNow Otto for App Engine.

Procedure

1. Open the ServiceNow Otto panel by selecting the ServiceNow Otto icon ( ).
2. In the ServiceNow Otto panel, select **Summarize a record** or enter `summarize a record`.
3. Enter the record number for the record that you want to summarize.
4. Select the Enter key.

   Note: Generating and displaying the summary may take several seconds.

5. **Optional:** When you're finished summarizing a record, you can copy the summary or provide feedback about the summary.

Option	Procedure
Provide feedback about the summary	If you think that the summary was helpful, se lect the helpful icon ( ). If you think that the summary wasn't helpful, select the not helpful icon ( ).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Procedure
	Note: This feedback improves the generative AI model and can help to im prove future versions of this skill.
Copy the record summary	If you want to reuse the summary, select the copy to clipboard icon ( ).

AI-led software development life cycle (SDLC) on the ServiceNow AI Platform

Build, test, source-control, and release ServiceNow applications end to end using the recommended ServiceNow stack.

The advent of agentic software development has changed the software development life cycle across industries and organizations. To keep pace with these rapidly-evolving technologies and enhanced ServiceNow tooling, ServiceNow has developed a new approach to the software development life cycle (SDLC). This new, AI-led SDLC covers all stages of development for both net new apps and customizations or extensions to existing ServiceNow applications. To learn more about the ServiceNow AI-led SDLC, see SDLC on ServiceNow.

The recommended ServiceNow stack includes:

- Build Agent
- ServiceNow Studio
- ServiceNow SDK
- Developer Sandboxes
- Automated Test Framework (ATF)
- ReleaseOps

Together, these components carry an application through the stages of the SDLC:

Planning

Define build-ready requirements with entities, roles, and Given-When-Then criteria.

Building

Author ServiceNow Fluent metadata and logic, with or without AI assistance.

Testing

Generate and run ATF coverage alongside development.

Managing

Branch, review, and merge changes in GitHub, then publish to the Application Repository.

Releasing

Assess and promote the deployment through ReleaseOps.

For the full walkthrough of this model, see the SDLC on ServiceNow.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Learning about developing on the ServiceNow AI Platform

Using the ServiceNow Creator Workflow products, you can create intuitive experiences that run on the powerful ServiceNow AI Platform.

In this document

Use this document to discover how to:

- Find all ServiceNow developer products
- Understand low-code and pro-code development
- Understand ServiceNow licenses
- Get support as a developer
- Understand the various ServiceNow user interfaces
- Learn the basics of being a developer

Please use the reader feedback feature to suggest additional topics.

What's new

Build apps smarter and deliver them faster with the new ServiceNow Studio. ServiceNow Studio empowers platform developers with a modern, unified environment for building on the ServiceNow AI Platform. ServiceNow Studio features streamlined navigation to applications and metadata, integrated low-code tools, efficient tracking and packaging of development work that accelerates development processes and enhances productivity.

Find ServiceNow developer products quickly

Find the ServiceNow developer product that can help you build amazing products on the ServiceNow AI Platform.

Find the developer product you need

Use the following tools to find ServiceNow developer products:

- ServiceNow Widget that guides you to developer products
- ServiceNow Table that lists developer products

If you're still not sure where to start, contact your sales representative. They can help you figure out what problems you're trying to solve and how to solve them.

Finding ServiceNow developer products

Use the following widget to find ServiceNow developer products that help you create applications that run on the powerful ServiceNow AI Platform.

Find the products that help you build applications

To find the ServiceNow developer products and information that are right for you, select one of the following tiles.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

I am comfortable coding	I do not code or write scripts
	
Build ServiceNow apps with custom code.	Build ServiceNow apps within a low code environment.

App Engine developer products

For a list of App Engine developer products, see App Engine products and offerings.

Pro-code tools: What you can do to build apps

Select an option below.

Build apps	Manage my apps	Engage with my customers
		
Learn about developing on the ServiceNow platform.	Customize your ServiceNow applications.	Learn how you can communicate with your customers using the ServiceNow platform.
Create reports	**Learn about the ServiceNow platform**	**Get help**
		
Create reports with your data on the ServiceNow platform.	See what you can do with the ServiceNow platform.	Find resources to help you with the ServiceNow platform.

Build apps with pro-code tools

Use the ServiceNow platform to build apps.

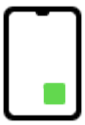
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ServiceNow Studio



Build custom applications using integrated builders and tools in one place.

Mobile App Builder



Build ServiceNow mobile applications.

VS Code Extensions



Edit your ServiceNow applications in Visual Studio Code with the help of the ServiceNow Extensions for VS Code.

AES



Build apps in a structured environment for fast development and deployment.

Manage apps with pro-code tools

Manage your ServiceNow apps.

Customize my apps



Customize your apps on the ServiceNow platform.

Automate my apps



Automate your ServiceNow applications.

Secure my apps



Secure your ServiceNow applications.

Test my apps



Test your apps on the ServiceNow platform.

Deploy my apps



Deploy your apps with the ServiceNow platform.

Maintain my apps



Maintain your apps on the ServiceNow platform.

Customize my apps with pro-code tools

Customize your ServiceNow apps.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

UI Builder



Learn how to create custom experiences with the ServiceNow platform.

Table Builder



Customize your data with Table Builder.

Workspace Builder



Customize your data with Workspace Builder.

Scripts



Use custom scripts with the ServiceNow platform.

Automate apps with pro-code tools

Automate your ServiceNow apps.

Automation Discovery



Identify automation opportunities for your workflows on the ServiceNow platform.

Flows in Workflow Studio



Enables process owners to automate work with the ServiceNow platform.

Decision Builder



Enable developers to decouple decision logic from their code by creating and maintaining decision rules.

Playbooks



Enable process owners to author cross-enterprise workflows and create a single, unified process.

Business rules



Use business rules to accomplish tasks like automatically changing values in form fields when certain conditions are met.

Secure apps with pro-code tools

Secure your ServiceNow apps.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

App Engine Studio roles and permissions



Control who is permitted to use or edit your apps.

Edge Encryption



Encrypt sensitive data on your company premises before sending it over the Internet.

UI Policies



Dynamically change the behavior of information on a form and control custom process flows for tasks.

Roles



Control access to features and capabilities in applications and modules.

Access Control



Control which Customer Service and Support employees can access your instance, and when.

Users and Groups



Manage the individuals who can access your instance by defining them as users in the system.

Test apps with pro-code tools

Test your ServiceNow apps.

Automated Test Framework



Create and run automated tests to confirm that your instance works after making a change.

Test Management



Streamline the management of testing processes to help you deliver software products more efficiently and with fewer errors.

Script Debugger



Debug scripts using session logs and ServiceNow AI Platform debugging tools such as a walk-through script debugger and error messages that display in the UI.

Script Tracer



Filter your debugging search to quickly narrow down script problems.

Session Log



View and download the session log for business rules, script includes, and a custom UI.

User impersonation



Impersonate other authenticated users for testing purposes and view impersonation logs.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

NOW Command Line Interface (CLI)



Perform instance operations from your local system with a command-line interface.

Deploy apps with pro-code tools

Deploy your ServiceNow apps.

The Next Experience

App Engine Management Center



Track and manage your app requests, deployments, applications, and collaborative developers using the App Engine Management Center (AEMC) in your production instance.

Core UI Tech Stack

Application Repository



Learn about developing on the ServiceNow platform.

Cloud Provisioning and Governance



The ServiceNow® Cloud Provisioning and Governance application provides a single interface to access cloud resources, publish cloud offerings to a catalog, and manage the usage of those resources.



Maintain apps with pro-code tools

Maintain your ServiceNow apps.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The Next Experience

System Update Sets



Allow administrators to group a series of changes into a named set and then move them as a unit to other systems for testing or deployment.

Service Mapping



The ServiceNow® Service Mapping application discovers all application services in your organization and builds a comprehensive map of all devices, applications, and configuration profiles used in these application services.

Engage with customers with pro-code tools

Engage with your customers using ServiceNow apps.

Chat bot

Virtual Agent Designer



Create and manage virtual agent topics using the ServiceNow platform.

Artificial intelligence

Natural Language Understanding Workbench



Use the NLU Workbench to create morphological representations of human language.

Platform Analytics



Optimize processes and increase productivity with Platform Analytics, virtual agents, and machine learning.

Predictive Intelligence



Provide a layer of artificial intelligence that empowers features and capabilities across ServiceNow applications to provide better work experiences

Reporting on data from pro-code apps

Most applications that you create have some level of reporting requirements. Reports should be actionable to drive change.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

"Reporting" generally refers to showing the data inside facts tables like Incident [incident]. You can also create key performance indicators (KPIs) to track changes in this data over time, through the Performance Analytics application.

Currently the ServiceNow AI Platform® is in transition between two user interfaces for showing this information:

- The older Core UI technology. This UI includes the Reporting application, which only shows data directly from tables, and PA Widgets, which show data from Performance Analytics indicators. Both reports and PA widgets can be placed on Core UI responsive dashboards. For more information, see Reporting, dashboards, and Performance Analytics in the Core UI.
- The newer Platform Analytics technology. This UI includes Data Visualizations, which let you report on data from any source. These visualizations can be placed on Platform Analytics dashboards, along with Platform Analytics filters. For more information, see Platform Analytics experience.

  All Platform Analytics objects are rooted in the Next Experience UI Framework and are available to developers. However, a non-developer can still build their own objects through the Platform Analytics experience, without using UI Builder. For more information, see Platform Analytics experience.

Note: Although "reporting" is a general term, this documentation usually uses "report" and "reporting" to refer to the Core UI Reporting application and "visualization" or "visualize" to refer to Platform Analytics data visualizations, for disambiguation.

Follow these guidelines when building reports or data visualizations:

- Creating reports or visualizations on large tables can negatively impact performance. Be sure to filter by a date range or other limiting criteria rather than showing all records on the table.
- Grouping by fields that contain many possible values can negatively impact performance.
- If loading a report or visualization gives a Long running transaction timer message, consider adding more data filters to reduce the load time.
- If you need to export a report, data visualization, or dashboard on a regular basis, schedule the export and email.

In Platform Analytics, you have several possibilities for showing multiple data visualizations on one page:

- Create the data visualizations and the dashboards entirely inside the Platform Analytics experience. This approach does not require a developer role or special technical knowledge, and probably should be explored before you try a more complex solution.
- Create data visualization components in a generic UI Builder page, along with filters and other components. This approach gives you the most freedom as a developer, but also requires the most configuration.
- Create the data visualizations and the dashboards inside the Platform Analytics experience, but then place the dashboards inside UI Builder pages using the Dashboard page template. This approach lets you use the convenience and special features of the Platform Analytics experience to create dashboards, data visualizations, and filters in your own experiences/workspaces. This approach also gives you the freedom to customize the page configuration partially. For more information, see Creating Platform Analytics pages in your own workspace.
- Create a technical dashboard and populate it inside UI Builder. This approach is almost the same as creating your own UI Builder page from scratch, but the dashboard is available in the dashboard library, has dashboard details, and can be shared like other dashboards. For more information, see Technical dashboards.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Low-code and no-code tools: What you can do to build apps

When building low-code apps, you can customize them, create reports on their usage, and more.

Build apps



Learn about developing on the ServiceNow platform.

Manage my apps



Customize your ServiceNow applications.

Engage with my customers



Learn how you can communicate with your customers using the ServiceNow platform.

Create reports



Create reports with your data on the ServiceNow platform.

Learn about the ServiceNow platform



See what you can do with the ServiceNow platform.

Get help



Find resources to help you with the ServiceNow platform.

Build apps with low-code and no-code tools

Use the ServiceNow platform to build apps.

Creator Studio



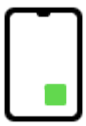
Build request-fulfill apps with no coding experience required.

AES



Build apps in a structured environment for fast development and deployment.

Mobile App Builder



Build ServiceNow mobile applications.

Manage apps with low-code tools

Manage your ServiceNow apps.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Customize my apps  Customize your apps on the ServiceNow platform.	**Automate my apps**  Automate your ServiceNow applications.	**Secure my apps**  Secure your ServiceNow applications.
Test my apps  Test your apps on the ServiceNow platform.	**Deploy my apps**  Deploy your apps with the ServiceNow platform.	**Maintain my apps**  Maintain your apps on the ServiceNow platform.

Customize my apps with low-code tools

Customize your ServiceNow apps.

UI Builder  Learn how to create custom experiences with the ServiceNow platform.	**Table Builder**  Customize your data with Table Builder.
Workspace Builder  Customize your data with Workspace Builder.	

Automate apps with low-code tools

Automate your ServiceNow apps.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Automation Discovery



Identify automation opportunities for your workflows on the ServiceNow platform.

Flows in Workflow Studio



Enables process owners to automate work with the ServiceNow platform.

Decision Builder



Enable developers to decouple decision logic from their code by creating and maintaining decision rules.

Playbooks



Enable process owners to author cross-enterprise workflows and create a single, unified process.

Business rules



Use business rules to accomplish tasks like automatically changing values in form fields when certain conditions are met.

Secure apps with low-code tools

Secure your ServiceNow apps.

App Engine Studio roles and permissions



Control who is permitted to use or edit your apps.

Edge Encryption



Encrypt sensitive data on your company premises before sending it over the Internet.

UI Policies



Dynamically change the behavior of information on a form and control custom process flows for tasks.

Roles



Control access to features and capabilities in applications and modules.

Access Control



Control which Customer Service and Support employees can access your instance, and when.

Users and Groups



Manage the individuals who can access your instance by defining them as users in the system.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Test apps with low-code tools

Test your ServiceNow apps.

Automated Test Framework



Create and run automated tests to confirm that your instance works after making a change.

Test Management



Streamline the management of testing processes to help you deliver software products more efficiently and with fewer errors.

Script Debugger



Debug scripts using session logs and ServiceNow AI Platform debugging tools, such as a walk-through script debugger and error messages that display in the UI.

Script Tracer



Filter your debugging search to quickly narrow down script problems.

Session Log



View and download the session log for business rules, script includes, and a custom UI.

User impersonation



Impersonate other authenticated users for testing purposes and view impersonation logs.

NOW Command Line Interface (CLI)



Perform instance operations from your local system with a command-line interface.

Deploy apps with low-code tools

Deploy your ServiceNow apps.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The Next Experience

App Engine Management Center



Track and manage your App Engine Studio, Creator Studio, and ServiceNow Studio requests, deployments, applications, and collaborative developers using the App Engine Management Center (AEMC) in your production instance.

Core UI Tech Stack

Application Repository



Learn about developing on the ServiceNow platform.

Cloud Provisioning and Governance



The ServiceNow® Cloud Provisioning and Governance application provides a single interface to access cloud resources, publish cloud offerings to a catalog, and manage the usage of those resources.



Maintain apps with low-code tools

Maintain your ServiceNow apps.

The Next Experience

System Update Sets



Allow administrators to group a series of changes into a named set and then move them as a unit to other systems for testing or deployment.

Service Mapping



The ServiceNow® Service Mapping application discovers all application services in your organization and builds a comprehensive map of all devices, applications, and configuration profiles used in these application services.



Engage with customers with low-code tools

Engage with your customers using ServiceNow apps.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Chat bot

Virtual Agent Designer



Create and manage virtual agent topics using the ServiceNow platform.

Artificial intelligence

Natural Language Understanding Workbench



Use the NLU Workbench to create morphological representations of human language.

Platform Analytics



Optimize processes and increase productivity with Platform Analytics, virtual agents, and machine learning.

Predictive Intelligence



Provide a layer of artificial intelligence that empowers features and capabilities across ServiceNow applications to provide better work experiences

Reporting on data from low-code apps

Most applications that you create have some level of reporting requirements. Reports should be actionable to drive change.

"Reporting" generally refers to showing the data inside facts tables like Incident [incident]. You can also create key performance indicators (KPIs) to track changes in this data over time, through the Performance Analytics application.

Currently the ServiceNow AI Platform® is in transition between two user interfaces for showing this information:

- The older Core UI technology. This UI includes the Reporting application, which only shows data directly from tables, and PA Widgets, which show data from Performance Analytics indicators. Both reports and PA widgets can be placed on Core UI responsive dashboards. For more information, see Reporting, dashboards, and Performance Analytics in the Core UI.
- The newer Platform Analytics technology. This UI includes Data Visualizations, which let you report on data from any source. These visualizations can be placed on Platform Analytics dashboards, along with Platform Analytics filters. For more information, see Platform Analytics experience.

  All Platform Analytics objects are rooted in the Next Experience UI Framework and are available to developers. However, a non-developer can still build their own objects through the Platform Analytics experience, without using UI Builder. For more information, see Platform Analytics experience.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: Although "reporting" is a general term, this documentation usually uses "report" and "reporting" to refer to the Core UI Reporting application and "visualization" or "visualize" to refer to Platform Analytics data visualizations, for disambiguation.

Follow these guidelines when building reports or data visualizations:

- Creating reports or visualizations on large tables can negatively impact performance. Be sure to filter by a date range or other limiting criteria rather than showing all records on the table.
- Grouping by fields that contain many possible values can negatively impact performance.
- If loading a report or visualization gives a Long running transaction timer message, consider adding more data filters to reduce the load time.
- If you need to export a report, data visualization, or dashboard on a regular basis, schedule the export and email.

In Platform Analytics, you have several possibilities for showing multiple data visualizations on one page:

- Create the data visualizations and the dashboards entirely inside the Platform Analytics experience. This approach does not require a developer role or special technical knowledge, and probably should be explored before you try a more complex solution.
- Create data visualization components in a generic UI Builder page, along with filters and other components. This approach gives you the most freedom as a developer, but also requires the most configuration.
- Create the data visualizations and the dashboards inside the Platform Analytics experience, but then place the dashboards inside UI Builder pages using the Dashboard page template. This approach lets you use the convenience and special features of the Platform Analytics experience to create dashboards, data visualizations, and filters in your own experiences/workspaces. This approach also gives you the freedom to customize the page configuration partially. For more information, see Creating Platform Analytics pages in your own workspace.
- Create a technical dashboard and populate it inside UI Builder. This approach is almost the same as creating your own UI Builder page from scratch, but the dashboard is available in the dashboard library, has dashboard details, and can be shared like other dashboards. For more information, see Technical dashboards.

What ServiceNow product solves your problem

Use this table to find ServiceNow products that help solve your problem.

I want to	Using	ServiceNow product or topic
Get an overview of developing		• What is ServiceNow? • Get a development instance • Demos • How the ServiceNow AI Platform works • Licensing • What is low-code development?
Get help developing applications		• Getting started on the ServiceNow AI Platform • ServiceNow developer community

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

I want to	Using	ServiceNow product or topic
		• Customer support • Knowledge base • Customer Success Center • ServiceNow training • Additional support • Known error portal
Plan to create an application		• Delegated development • Delegated development in Creator Studio • Delegated development in App Engine Studio • Link an app to source control in ServiceNow Studio • Common Service Data Model
Create an application	• No code • Next Experience	• Creator Studio • App Engine Studio
	• Coding and scripting • Core UI tech stack • Forms and lists	• Build Agent • Autonomous Engineer in Build Agent • ServiceNow Studio • ServiceNow Extensions for Visual Studio Code • REST APIs • ServiceNow AI Platform® forms, fields, and lists
	Mobile platforms	• Mobile App Builder • Mobile Card Builder
Customize an application	UI pages	• UI Builder • Workspace Builder (in App Engine Studio only)
	Data, tables	Table Builder

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

I want to	Using	ServiceNow product or topic
	Scripts	• No-code scripting (Workflow Studio - Building custom actions) • ServiceNow Studio • Client-side scripting • Server-side scripting • Script Debugger • ServiceNow CLI
	Service Portals	• Create and edit a page using the Service Portal Designer • Service Portal widgets • Configuring Next Experience themes and preferences
Add automation		• Automation Discovery • Workflow Studio • Exploring Decision Tables • Playbooks • Journey designer • Classic Business rules
Integrate my applications with ServiceNow	Automation and flows	• Integration Hub • Flows in Workflow Studio
	APIs	• REST API reference • Server API reference • Client API reference • Browse APIs by product • Workflow Studio - Building custom actions • Integration Hub Remote Process Sync
Secure an application	No code (App Engine Studio only)	Configure AES personas and roles
	Coding and scripting, Core UI tech stack	• Edge Encryption • Using UI policies • Managing roles

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

I want to	Using	ServiceNow product or topic
		• ServiceNow® access control • User administration
Engage customers	Artificial intelligence	• Analyze and optimize business processes NOW Intelligence • Predictive Intelligence
Test an application	For App Engine Studio apps	Automated Test Framework
	Coding, Core UI tech stack	• Automated Test Framework • Test Management applications • Script Debugger • Script Tracer • Session Log • Impersonate a user • ServiceNow CLI
Deploy an application	No code (App Engine Studio only)	App Engine Management Center
	Coding, Core UI tech stack	• ServiceNow application repository • Cloud Provisioning and Governance
Maintain an application		• Maintaining and monitoring the ServiceNow AI Platform • Maintaining and monitoring the ServiceNow AI Platform System update sets • Service Mapping • Instance Data Replication • MetricBase
Create reports		• CMDB 360 view in CMDB Workspace • Analyze and optimize business processes - Performance Analytic Reports • Analytics Hub • Analytics Center
Copy an instance		Instance Data Replication

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

I want to	Using	ServiceNow product or topic
Personalize your experience		Personalize your experience

Programming basics

Software development usually encompasses a standard set of steps. If you're new to developing applications, read on.

Your job as a developer

If you're new to developing applications, if you don't even know how to code, ServiceNow has a low-code development platform called Creator Studio that makes creating a basic request-fulfill application possible for you. Development might be new waters for you to swim in but you will find the low-code tools easy to use. So, don't be shy about developing an app!

Software developers do many things but what they have in common is solving problems using computer code. ServiceNow offers coding platforms that require no coding experience, mid-level coding experience, and expert coding experience. The less coding required, the more work the coding platform does for you. So, even experienced software developers will appreciate low- and mid-level coding platforms, such as Creator Studio and App Engine Studio.

Software development life cycle

The common stages of developing an application are:

1. Define the problem.
2. Plan the solution.
3. Code the solution.
4. Test the application.
5. Deploy the application.
6. Document the application.

These steps are commonly referred to as the software development life cycle. ServiceNow developer documentation is grouped according to these steps.

Do you really need to create a new application

Your first step is to determine if creating an application is necessary. Consider:

- Can you extend an existing application instead of creating a new one?
- How many people will the application serve? If not many, is it worth the effort?
- Are you willing to maintain the application as things change over time?
- How often will anyone use this application? If not often, is it worth the effort?

If creating a new application makes sense, then move forward.

Defining the problem

If you're planning to create an application, you must have some notion of what you want it to do. You might not have a detailed idea of the solution, however. At a minimum, you need to define in detail:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Desired application output
- Data going into the application

The data going in typically comes from a table of data. You need to know the table name and the type of data in it. You can use ServiceNow tools, like Table Builder, to store and customize your data.

The desired output might be a dashboard, data stored in a table, or a piece of equipment being sent to an employee. Before you decide on the output, show a mockup of your output to a lot of people to see if they can suggest improvements to the output. Do not skip this step.

Planning the solution

There are various ways to create an algorithm that produces the output you want. The easiest methodology is to use a flowchart that starts with the input data and shows each step in the process that leads to the output. Using a flowchart enables you to focus on the big picture of what you want to do and avoid how to do it. Look at your flowchart critically to optimize the process.

You may need to get permissions to use tables of data or even permission to create an application. For example, citizen developers are granted app development permissions through App Engine Management Center.

For more information about ServiceNow planning tools, see Planning your application.

Coding the application

To code the application, you can use:

- Low-code ServiceNow tools, such as App Engine Studio.
- Mid-level and pro-code ServiceNow tools, such as ServiceNow Studio.
- Pro-dev source code tools, such as ServiceNow IDE and ServiceNow SDK.

For more information about ServiceNow coding platforms, see Developing your application.

Testing your application

Testing your application requires inputting data to make sure the outcome is accurate. It is important to input a lot of data, even incorrect data, to see how well your application handles correct and incorrect inputs. Put in data that is too large or the wrong type. Put in no data at all. Try to break your application to prevent your users from doing so. Handling error conditions gracefully is critical for a good customer experience.

For more information about ServiceNow testing tools, see Testing and debugging applications.

Deploying your application

Now that you've tested your application, it's time to deploy it so customers can use it. Application deployment is done by system administrators not developers.

For information about ServiceNow deployment tools, see Deploying applications.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Maintaining your application

Maintaining the application deals with issues such as overloading a server, application errors, and feature updates. Accounting for feature updates is a developer's job. Seeing how well an application is working is a system administrator's job.

For more information about ServiceNow tools that help you maintain your application, see Maintaining your application.

Documenting your application

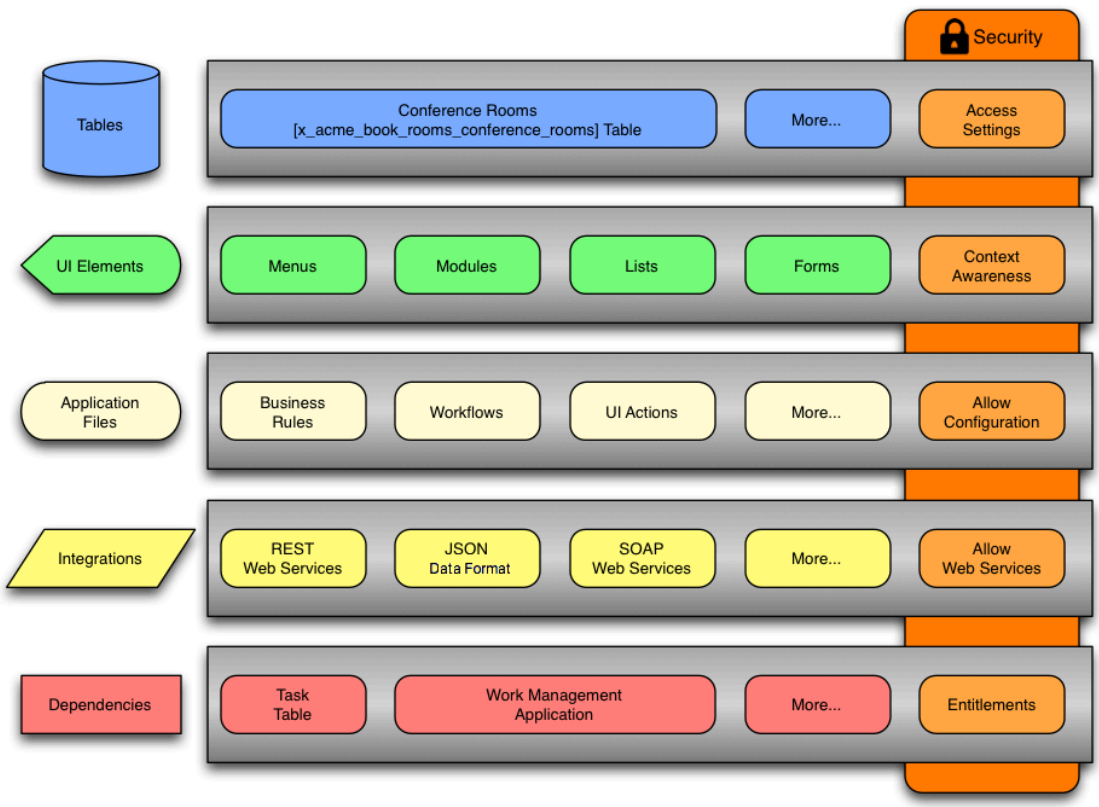
Application development is never complete until the application is documented. There are many reasons to document your application:

- Internal notes help other developers work on your application. It takes a lot of time to figure out the reasoning behind programming decisions and methodology. Documenting those helps future developers maintain your application.
- Users need to know how to use your application. As transparent as you believe the application is, you will find many people who get frustrated trying to use it. Do a usability test on your application and documentation to see where users run into problems. The broader the audience, the more important the application, the more important the documentation. Consider using an experienced technical writer to write the documentation.

Anatomy of an application

Applications consist of several types of files and records that collectively deliver a service.

Application model



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Custom application record

The custom application record defines and identifies an application and all its associated artifacts.

It is similar to a system dictionary record for a table or column in that it stores the most current configuration of an application. The system automatically creates a custom application record during the application creation process. Application developers can use this record to perform the following tasks.

- Change the application name
- Change the application version
- View the scope the system uses to identify application files and configuration records
- Enable scoped administration
- Manage design and runtime access to the application
  - Select what JavaScript standards the application supports
  - Select how the system tracks runtime API operations
  - Permit or restrict access to tables from other applications
- Monitor or enforce subscriptions
- Select the default menu in which to display application modules
- Set the user role required to access the application
- Add or update a logo
- View all application files
- View resources from other applications on which the application depends
- View the run-time resource to which the application has been granted access
- View the design-time resources to which the application has been granted access

Application versions

Each installed application has a version as defined by its application developer in the custom application record.

The system uses this version information to determine whether there are updates available from the ServiceNow application repository or ServiceNow Store.

Application scope

Application scoping protects applications by identifying and restricting access to application files and data.

Administrators can specify what parts of an application are accessible to other applications from:

- The Custom application record
- Each application Table record

For example, suppose that you create a conference room booking application in its own application scope. By default, the application can access and change its own tables and business logic but other applications can't unless you give them explicit permission. The application scope ensures:

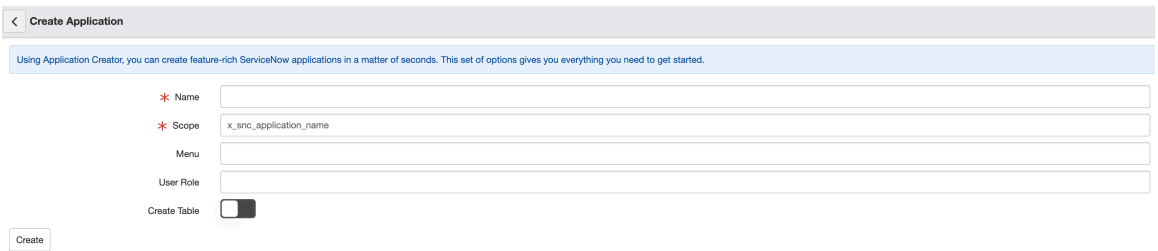
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- The conference room booking application does not interrupt core business services.
- Other applications do not interfere with its normal functioning.

By default, all custom applications have a private scope that uniquely identifies them and their associated artifacts with a namespace identifier. The application scope prevents naming conflicts and allows the contextual development environment to determine what changes, if any, are permitted. Application developers specify an application scope when they create an application.

Tip: Global apps can alter data that you don't intend to alter. You should leverage scoped apps to create new tables, and consider transitioning old ones to scoped apps. This allows you to split responsibilities with Delegated Development. To learn more about delegated development, see Delegated development and deployment.

Selecting the **Can Edit Application in Studio** option does not affect any custom or global applications in development on an instance. If you are the owner and choose to publish the application, you can restrict the development of customizations in Studio of. If you set **Can Edit Application in Studio** to false and then publish it, those who download the application to their own instances won't be able to edit the application in Studio. But they will have access to Source Control features inside of Studio.



Related topics

Runtime access to applications tables

Global scope

The global scope is a special application scope that identifies applications developed prior to application scoping, or applications intended to be accessible to all other global applications.

Applications in the global scope do not append a unique namespace identifier to the application name. Global applications can have naming conflicts and data collisions when developers create multiple global applications with the same name.

Since all global applications are in the same scope, they bypass scope protections. Global applications allow other global applications access to their tables to

- Read records
- Run API requests
- Create configuration records

Typically, only applications provided by ServiceNow are in the global scope. However, any custom applications created before application scope was implemented are also in the global scope. Additionally, developers with the sn_g_app_creator.app_creator role can create applications in the global scope.

Applications in the global scope are eligible for upload to the application repository, but not to the ServiceNow Store.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Related topics

- Publish an application to the application repository
- Publish an application to the ServiceNow Store

Namespace identifier

The system adds a namespace identifier to the front of application artifacts such as tables, scripts, and configuration records.

The identifier cannot be changed or removed from application artifacts to ensure that they are always associated to the proper application and that they have a unique name.

The system generates a namespace identifier from the following information:

Elements used to generate a namespace identifier

Element	Requirements	Sample Value
The prefix characters for a scoped application.	Scoped applications always start with an x_ prefix.	x_
The instance customer prefix (glide.appcreator.company.code)	This string is two to five characters long. ServiceNow generates this prefix for each customer. The instance stores the prefix in the *glide.appcreator.company.code* system property.	acme
The application ID	This string can be up to 40 characters long. If the string is longer than 18 characters, the system truncates the application name and appends it to prefix x_<glide.appcreator.company.code>_. Application developers set this ID when they create the application. The system uses the application name by default.	book_rooms

The example values generate a namespace identifier of x_acme_book_rooms.

Namespace identifier examples

The following examples illustrate generating namespace identifiers for applications, tables, and fields.

Example namespace identifiers

Action	Element generated	Explanation
1. Generate a namespace identifier for a private scope application called Book Rooms.	x_acme_book_rooms	This is a combination of the vendor prefix and applicatio

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example namespace identifiers (continued)

Action	Element generated	Explanation
2. Generate a namespace identifier for a global scope application called Marketing Events.	None	The system does not generate namespace prefixes for g applications.
3. Add the conference rooms table to the Book Rooms application.	x_acme_book_rooms_conference_rooms	This table is in the Book Rooms scope so begins with th namespace identifier.
4. Add a Marketing Event table to a global application.	u_marketing_event	Custom tables in the global scope always use the `u_` namespace identifier.
5. Select Book Rooms in the application picker and add the **Capacity** field on the Conference Rooms table.	capacity	The field is in the same scope as the table so it does not have its own namespace identifier. However, to dot-walk to the field in a script, you would use the full path including the table namespace identifier: `x_acme_book_rooms_conference_rooms.cap`
6. Select Book Rooms in the application picker and add the **Theme** field to the Marketing Event table.	x_acme_book_rooms_theme	The field is in a different scope from the table so it is prefixed with the `x_acme_book_rooms` namespace identifier. To dot-walk to the field in a script, you would use full path including the field namespace identifier: `u_marketing_event.x_acme_book_rooms_th` **Note:** This example assumes that the Marketing E table allows other application scopes to add fields.
7. Select Marketing Events in the application	u_theme	Custom fields in the global scope use the **u_** prefix. To dot-walk to the field in a script, you would use `u_marketing_event.u_theme`.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example namespace identifiers (continued)

Action	Element generated	Explanation
picker and add the **Theme** field to the Marketing Event table.		

Application tables

Application developers create tables and their associated lists and forms for users to add and update records.

An application owns its tables and determines whether other applications can access resources from them. For example, the Book Rooms application can store conference room data in the Conference Rooms [x_acme_book_rooms_conference_rooms] table and permit other applications to read this data.

The system uses standard access controls to manage user access to application data. During application creation, developers can specify an application-specific user role for these access controls. They can also use application access settings to manage runtime and design time access to application tables.

Note: Certain ServiceNow AI Platform subscriptions include custom table entitlements. You can create custom tables for any purpose, up to the entitlement limit in the subscription. To learn more about how your usage administrator maps the custom tables that you create to subscriptions, see Map custom tables to a product subscription in Subscription Management.

Be aware of these database limitations:

- The system can only have a maximum of 1000 columns per table. Although 1000 columns is a specified limit, this limit doesn't mean you can physically have 1000 columns within a table. The number of columns within a table is defined by the database used in the ServiceNow datacenter, not by the ServiceNow AI Platform.
- Every table, regardless of storage engine, has a maximum row size of 65,535 bytes. Storage engines may place additional constraints on this limit, reducing the effective maximum row size.
- The system can't have more than 10 medium-length or longer **String** fields to a single table. Attempting to save a large number of characters in 11 or more String fields can result in the following error: Syntax Error or Access Rule Violation detected by database [Row size too large (> `8126`)].
- When you create fields, the u_ prefix is automatically added to the column name. If the column label that you enter contains leading numeric characters, they are replaced by the u_ prefix.

For more information on database limitations and general questions on tables in your ServiceNow instance, see KB0749585.

Related topics

Application access settings

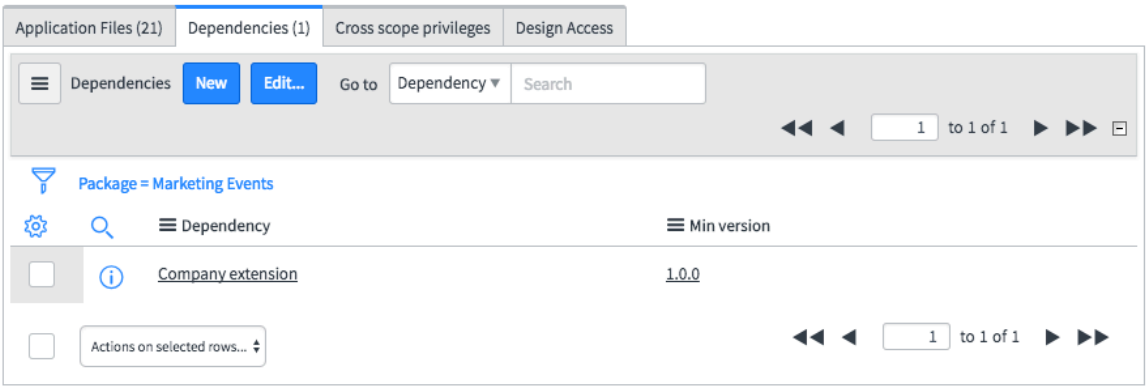
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Dependencies for custom applications

Every custom application record includes a related list identifying its dependencies on other applications.

Administrators can review this list to determine whether an application poses any risk to existing processes or data. Application developers can use this list to ensure that their applications have the proper access to other applications.

Sample application dependencies



Application files

Application files are configuration records that allow developers to extend application functionality.

Application developers create application files when they add configuration records for application logic such as business rules, workflows, and script includes. Scoped applications do not own these tables, but they do own the records (files) within these tables. For example, adding a business rule to check for available rooms from the Conference Room table adds an application file to the Business Rule [sys_script] table. Application developers can view the complete list of application files from the custom application record.

The Application File [sys_metadata] table is the parent table for all tables that contain configuration records. It provides a series of standard fields that define the attributes for a configuration record. Tables that contain configuration records extend the Application File table. For example, the Business Rule [sys_script] table extends the Application File table.

Developers do not create application file records directly from the Application File table. Instead, they create or modify configuration records, and the system creates or modifies the associated application file record. Most configuration record tables do not display a reference field or related list for the Application File table. By default, only Applications [sys_scope] and Tables [sys_db_object] have a reference to the Application File table.

The system also tracks application file records in Update Sets. Whenever you change an application file record or its related configuration record, the system adds a corresponding record in the Customer Updates [sys_update_xml] table. The system uses a single update record, ensuring that a configuration record and its application file are kept in sync when transferring applications between instances. To avoid collision, users are warned when they edit an application file that has been previously edited in another Update Set.

Administrators can:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- View file properties for configuration records.
- Protect application files from changes during upgrades.
- View parent-child relationships between configuration records.

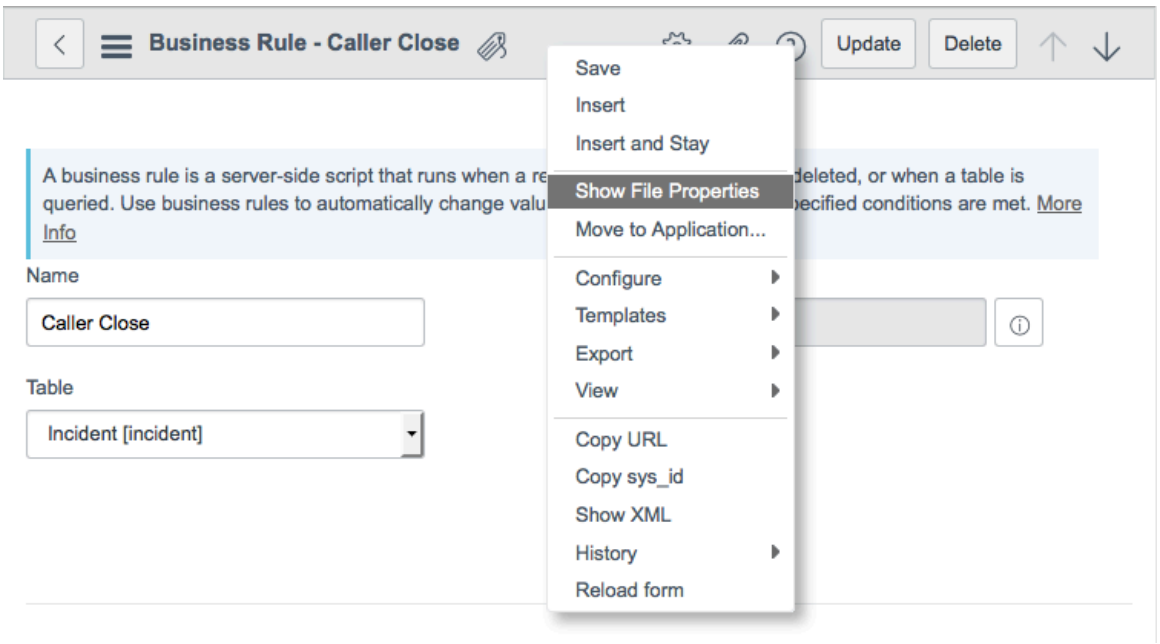
View file properties

Administrators can view the application file properties of a single record.

Before you begin
Role required: admin

Procedure

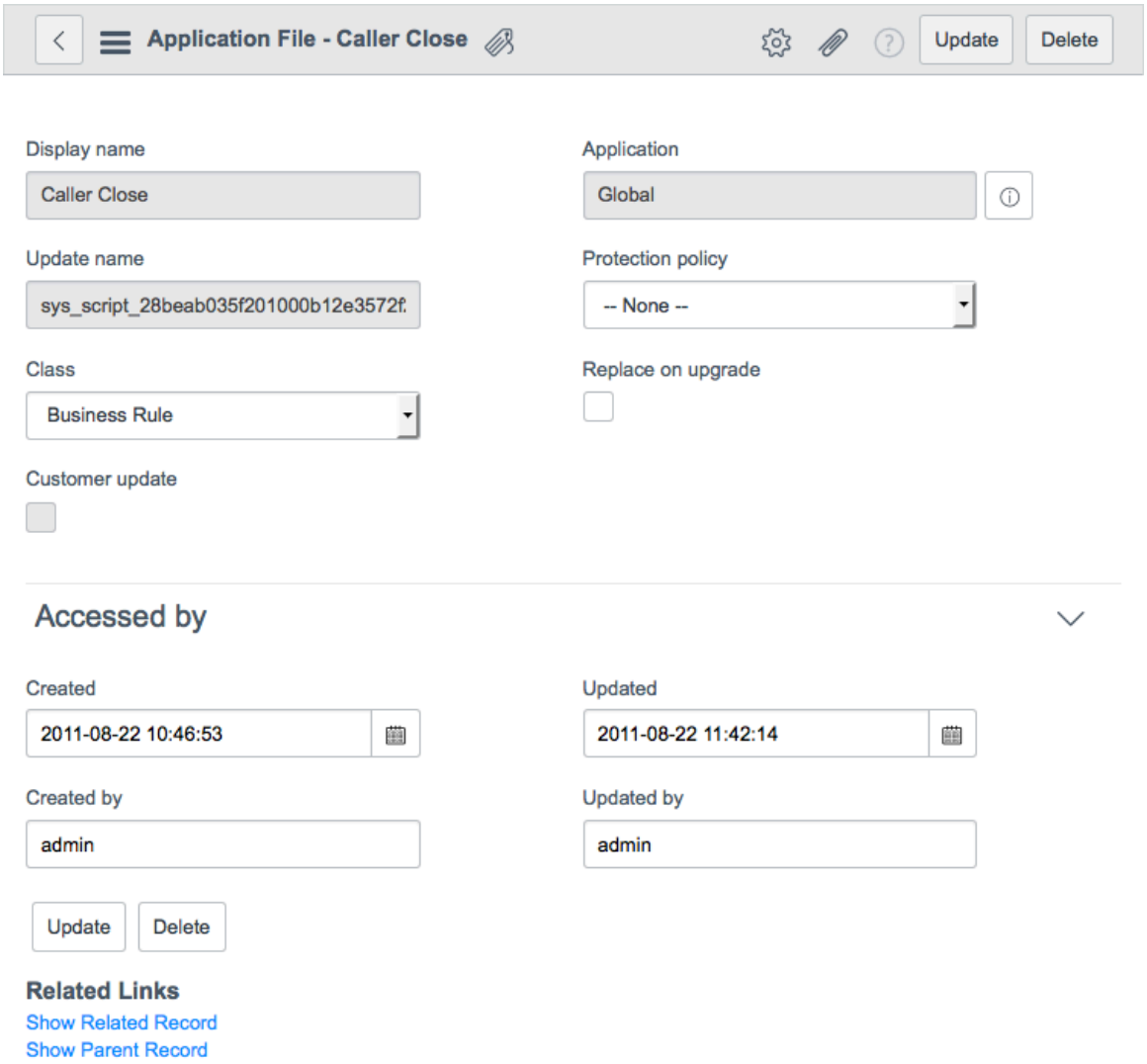
1. Navigate to the form view of the configuration record.
   For example, navigate to **All** > **System Definition** > **Business Rules** and select a business rule for the Incident table.
2. Right-click the form header and select **Show File Properties**.



   The Application File table provides the standard fields that define the attributes for the configuration record.

3. To return to the configuration record view, click the **Show Related Record** related link.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



4. Navigate between a customer update record, the file properties view, and the configuration record view.

Option	Description
Show Related Record related link	Navigate to the configuration record
Show Parent Record related link	Navigate to the parent record of the current configuration record.
Descendants related link	View child configuration records, such as a field label translation.

Application File form

Use the Application File form to view relationships between applications and configuration records.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Application File form fields

Field	Description
Display name	Display name for the configuration record.
Update name	Unique identifier for the configuration record. This value is used to identify versions and updates of the record.
Class	Table that contains the configuration record.
Application	Application that contains the configuration record.
Protection policy	Policy that determines if the configuration record is protected from changes. See Protected Application Files.
Created	Creation date of the configuration record.
Created by	User who created the configuration record.
Updated	Last update date for the configuration record.
Updated by	User who last updated the configuration record.
Related Record Versions	Version records for the related configuration record. A version record is created every time a user changes the related record. Use this list to compare versions of the configuration record or to revert to a previous version. See Versions.
Related Record Updates	Local update records for the related configuration record. An update record is created for the most recent change to the related record in a given Update Set. See Update Sets.

Note: You may find it useful to manually add an **Owned By** (owned_by) field to this form. It indicates, for each application, who in IT owns the application and is responsible for maintaining information about it. The Data Certification function uses this field to send certification tasks every quarter, or at any time interval that is configured. You can assign certification tasks to other users, but using this field limits configuration effort. To learn more about adding a field to a form, and about Data Certification, see Using the form designer and Data Certification.

Application file protection policy

A read-only protection policy prevents anyone from modifying an application file or its related record.

Some application code shipped with the ServiceNow system requires special protection. Only a ServiceNow employee can alter the protection policy and then modify the application file or its related record. A ServiceNow employee cannot edit protected files without changing the policy designation first.

To prevent customizations from being overwritten by system upgrades, the upgrade process automatically skips changes to customer-updated records. If you modify an application file or related record that is later designated as *Read-only* in an upgrade, the application file maintains the default protection policy of *Write*. You keep the existing modifications and can continue modifying the records.

Note: Reverting a customized file to its baseline state causes the record to inherit the new protection policy as well. For example, a record going from a *Write* protection policy to a *Read-only* protection policy.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Relationships between configuration records

The Application File Types table defines parent-child relationships between configuration records.

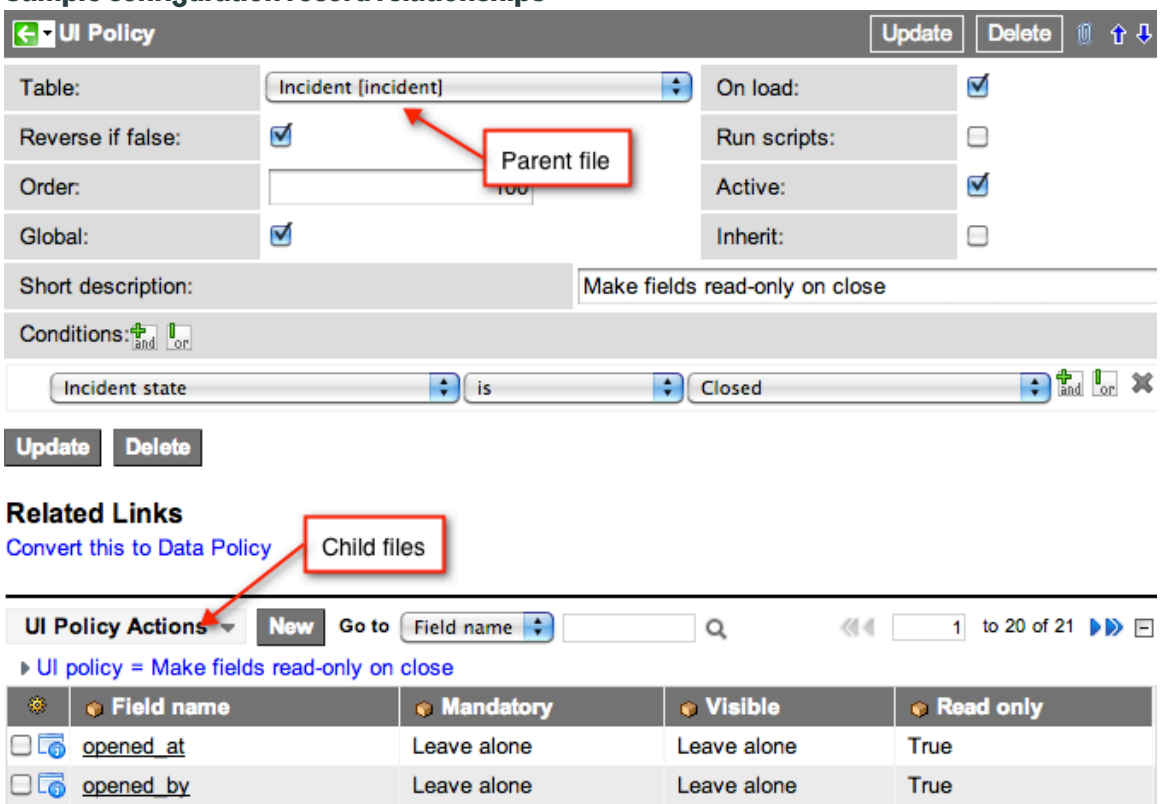
The system uses this structure to keep configuration records that normally belong together in the same application.

Note: Do not modify the Application File Types table as it provides system functionality.

For example, consider the parent-child relationships for a UI policy.

- The UI policy is a child of the application table.
- UI policy actions are children of the UI policy.
- UI policy actions have a parent UI policy and a grandparent application table.
- The UI policy actions and the UI policy are all descendants of the application table.

Sample configuration record relationships



Fix scripts

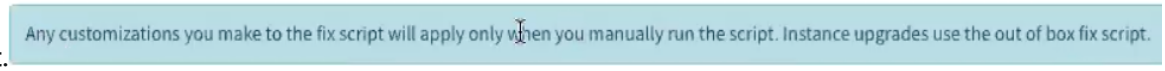
A fix script is server-side JavaScript code that you run after an application is installed or upgraded.

Include fix scripts to make changes that are necessary for the data integrity or product stability of an application.

Administrators can create, manage, and run fix scripts. Users with the script_fix_admin role can create and manage fix scripts but cannot run fix scripts.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: An annotation with the following message shows up when you open an existing fix script or create a new fix script.



Create a fix script

Create fix scripts to ensure the system installs or updates an application properly.

Before you begin
Role required: script_fix_admin or admin

About this task
Use fix scripts to add, update, and delete data, including rules, scripts, and property settings.

Procedure

1. Navigate to **All** > **System Definition** > **Fix Scripts**.
2. Click **New**.
3. Define the fix script by filling in the fields on the form.

Field	Description
Name	Unique, descriptive name for the fix script.
Unloadable	Option to create Customer Update records [sys_update_xml] when the fix script runs. Fix script tests enforce the Unloadable option.
Before	Option to run the fix script before installing or upgrading the application.
Record for rollback	Option to record execution of the script so that it can be reverted if needed.
Description	Description of the fix script.
Script	Code for the fix script.

4. Click **Submit**.
5. Test the fix script and make any necessary changes.

Test a fix script

Test your fix scripts to ensure they install or update applications as expected.

Before you begin
Role required: admin

About this task
Fix scripts add, update, and delete data, including rules, scripts, and property settings.

Note: Do not test fix scripts on production instances.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

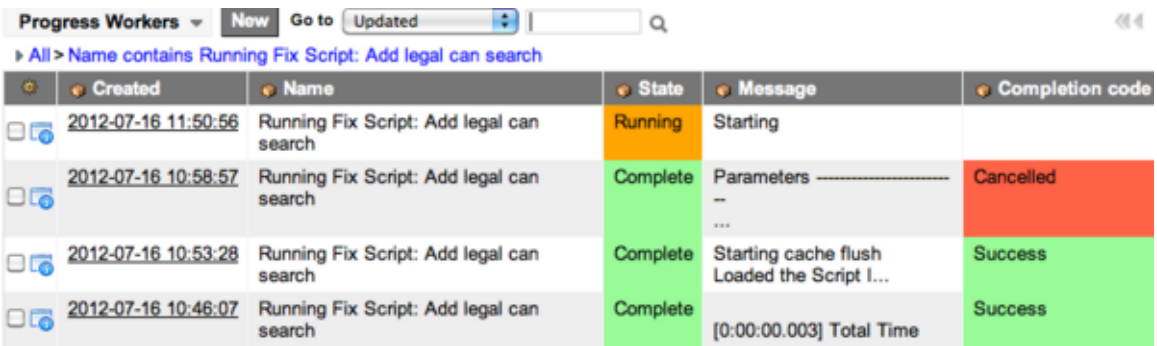
Procedure

1. Open the fix script record.
2. Review the code design to ensure that it can run more than once on the same system without causing damage.
   For example, you may write a fix script that adds a role to a property by default. Design the script so that it can run multiple times on the same system without overwriting the existing data, even if it is not necessary to run the script again after the initial installation.
3. Click the **Run Fix Script** related link.
   Tests enforce the Unloadable option.
4. Confirm how to run the script.

Option	Description
Proceed in Background	Use this option for long-running scripts, or if you do not know the expected execution time.
Proceed	Use this option to run the script immediately and display the results in a confirmation win dow.



5. Review the results from the Progress Workers related list, and make any necessary changes.
6. To manually stop a running fix script:
   a. From the **Progress Workers** related list, select a worker in the `Running` state.
   b. Select the **Cancel job** related link.



Run fix scripts

Fix scripts run at the first installation or updated after they have been added to an application. However, previous fix scripts do not run on subsequent application updates, and you must manually run any necessary fix scripts.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin
Role required: admin

About this task
To run a fix script:

Procedure
1. Navigate to **All** > **System Definition** > **Fix Scripts**.
2. Edit the filter to search for your application name.
   For example, `Application | is | Book Rooms`).
3. Open the fix script record.
4. Click **Run Fix Script**.

Fulfillment tables
To enable a production instance to enforce entitled usage of your ServiceNow Store App, you configure the tables where only record owners or subscribed app users can make updates.

For any table that you, the developer, create or extend to support a custom application, you can specify that the table is a fulfillment table. In a fulfillment table, only a subscribed fulfiller user can perform a fulfiller action (typically, create/update/delete a not-own record).

In contrast, for a table that is not a fulfillment table, any user—even a user who is not subscribed —can act as a requester. The intent is to allow the usage admin to enable subscription enforcement on any production instance that implements the application.

Ownership of records in a fulfillment table
To enable the system to identify a fulfiller action, you define how to determine ownership of any record in the table. The developer of the application specifies the set of conditions that determine whether a user owns the record. For example, *`UserA`* owns a record in a task table if *`UserA`* opened the record or another resource opened the record on behalf of *`UserA`*.

For task-extended tables, time cards, and apps that require a subscription, the system sets the table as a fulfillment table by default and auto-assigns the ownership condition. For tables that you create to support your app, you can mark the table as a fulfillment table and can specify the ownership condition (for example, use the filter **[opened_by][is][currentUser] OR [caller_id][is] [currentUser]**).

System default conditions for ownership

Action	Ownership condition [owner_condition]
task extension	opened_by (read-only)
catalog request	requested_for (read-only)
other tables in apps that require a subscription	sys_created_by (read-only)
tables created by developer for app that requires a subscription	Specified by developer

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Specify that a table is a fulfillment table

You configure a table as a fulfillment table to enable the system to prevent updates by users who are not subscribed to the app. For ServiceNow Store apps, you configure a table as a fulfillment table to enforce that fulfillment usage complies with your subscription use policy.

Before you begin
Role required: usage_admin, admin

Procedure

1. While working on a table, open the **Table Subscription Configuration** related list.
2. Select **Fulfillment table**.
3. Specify how the system determines ownership of records in the table so that both end users who own a record and subscribed fulfiller users can update the record: Specify the **Ownership condition**.
   For example, set the filter as **[opened_by][is][currentUser] OR [caller_id][is][currentUser]**.

Personal developer instance guide

ServiceNow® offers free personal developer instances (PDIs) to Developer Program members to learn, explore, and experiment with the ServiceNow AI Platform®.

ServiceNow® offers free personal developer instances (PDIs) to Developer Program members who want to develop applications or improve their skills on the ServiceNow AI Platform®. PDIs provide members with a sandbox environment to experiment with the ServiceNow AI Platform and build greater value for their businesses. The ServiceNow® Developer Program allows each member to have only one instance to maximize this valuable resource for all Developer Program members.

Personal Developer Instances (PDIs) can be used to test application ideas without impacting a customer or partner instance. PDIs are intended for you to use to learn, explore, and experiment with the ServiceNow AI Platform®. PDIs are not intended to be used for business or production purposes. Using a PDI for business or production is a violation of the ServiceNow® Developer Program Agreement. With a PDI, you receive admin-level access to all of the features the ServiceNow AI Platform® has to offer. Use your PDI on a continuous basis and it is yours forever.

Learn more on the ServiceNow Developer Site.

Understanding PDIs

PDIs require you to select a specific release and maintain consistent usage to avoid hibernation.

Sign in to the Developer Site to keep your PDI active. After a period of inactivity, your instance is reclaimed. Back up your work regularly to prevent data loss.

Note: To save your work, commit your work to source control or create update sets. Work that is not saved may be lost due to instance reclamation.

When a PDI is reclaimed, the instance is unassigned from you, reset to its original state, and reassigned to another Developer Program member. Reclaiming instances allows the Developer Program to support active program members with free instances. The **Account** menu on the Developer Site header includes a reminder in the Activity section that you have ten days of inactivity.

Keep your PDI active by signing in or performing developer-type activities on your instance such as:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Creating an application
- Creating a table
- Adding a field to a table
- Creating a Script Include

Hibernation

To keep the developer program free and provide PDIs to everyone who wants one, instances hibernate when they are idle. When a PDI hibernates, the database and application server shut down to conserve computing resources. All your data is preserved when the PDI hibernates. You can wake up a hibernating instance by signing in to the Developer Site. Waking an instance from hibernation resets the ten-day inactivity countdown.

In addition, you should also be aware of the following:

- PDIs are returned to the pool of available instances if they go unused for ten days. Duration may change due to availability.
- Availability is not guaranteed.
- PDIs cannot be clone targets or clone sources for customer or partner instances.
- PDIs cannot be linked via team development to customer or partner instances since they belong to ServiceNow®.
- PDIs cannot publish to the ServiceNow Application Repository or the ServiceNow® Store.
- PDIs cannot work with Machine Learning, Instance Data Replication, or MetricBase.
- Many ServiceNow® Store applications cannot be installed on PDIs.

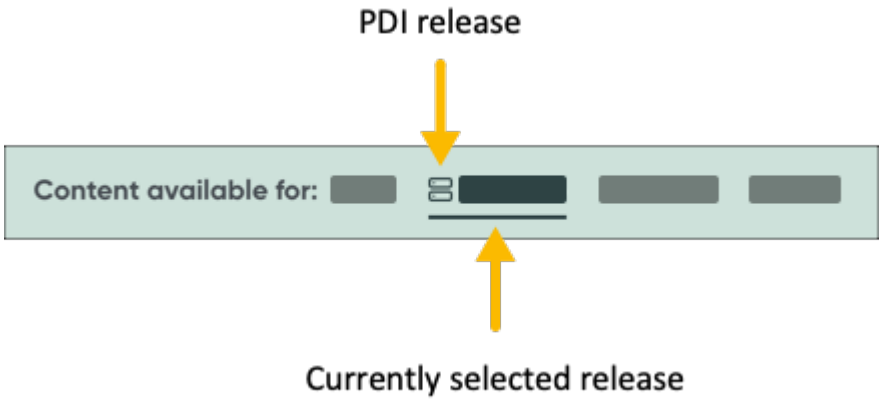
Mainentance

ServiceNow® occasionally needs to perform maintenance on the underlying infrastructure associated with PDIs. Maintenance includes updating and patching software, firmware, and replacing hardware. During maintenance, your PDI may not be accessible. The length of time required for maintenance depends on the type of maintenance. If available, the My Instance page indicates the expected completion time for planned maintenance. Your work and data on your PDI is unaffected by planned maintenance.

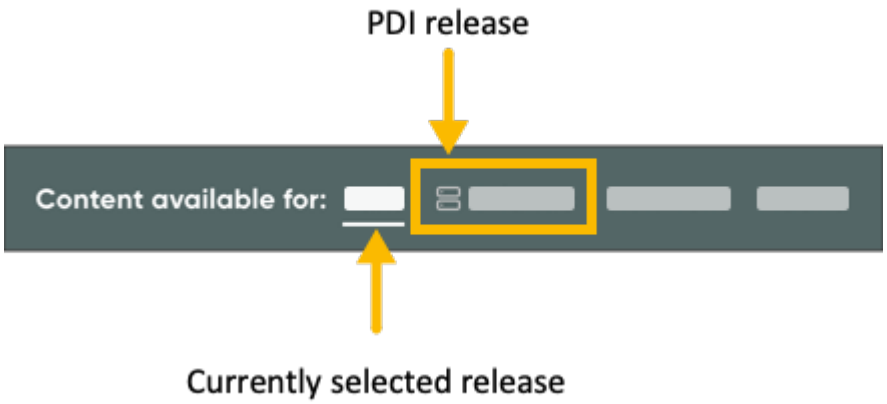
PDIs and the release selector

When content is available for different releases, use the release selector to select a release. The server icon next to the release name identifies the release of your PDI. The bar under the release name identifies the selected release. When working on your PDI, be sure the content matches your PDI.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



If the content does not match your PDI release, click the release with the server icon to view the content that matches the release of your PDI.



The selector only shows releases for which the current page of content is available except when the content is not available for your PDI release. When the content is not available for your PDI release, the PDI release is grayed out and cannot be selected.



If your PDI is not on the latest release, follow the steps for upgrading your PDI to get the latest release.

Get a development instance

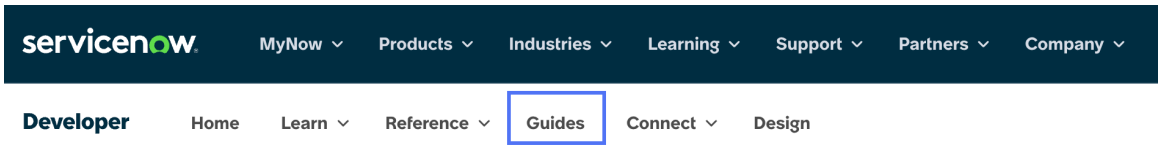
Request a personal development instance to build and test applications.

Before you begin
Role required: none

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

About this task
Get a free ServiceNow personal developer instance (PDI). Build applications within a sandbox to test application builders and deployment strategies.

Procedure
1. Navigate to the Developer Site in your browser.
2. Select **Sign In** or **Sign up and Start Building** if you don't have a ServiceNow developer account.
3. After logging into the ServiceNow Developer Site, select **Guides**.



4. Select **Personal Developer Instance (PDI) Guide** in the side navigation (you may need to scroll down).
5. Follow the instructions to obtain your personal developer instance.

Obtaining a PDI

Get a new instance to run a PDI on the latest generally available release.

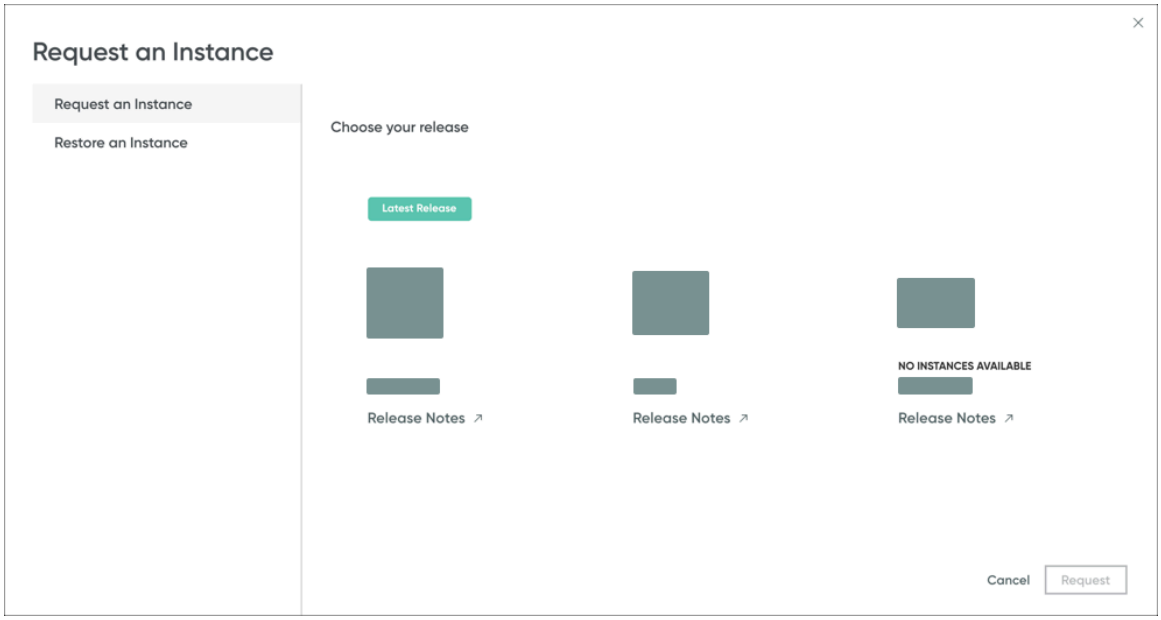
Before you begin
Role required: none

About this task
The first time you sign in to the Developer Site, you are given a PDI running the latest generally available release. You can use this PDI as long as you keep it active. If you want a PDI running an earlier release, release your PDI and request a new instance.

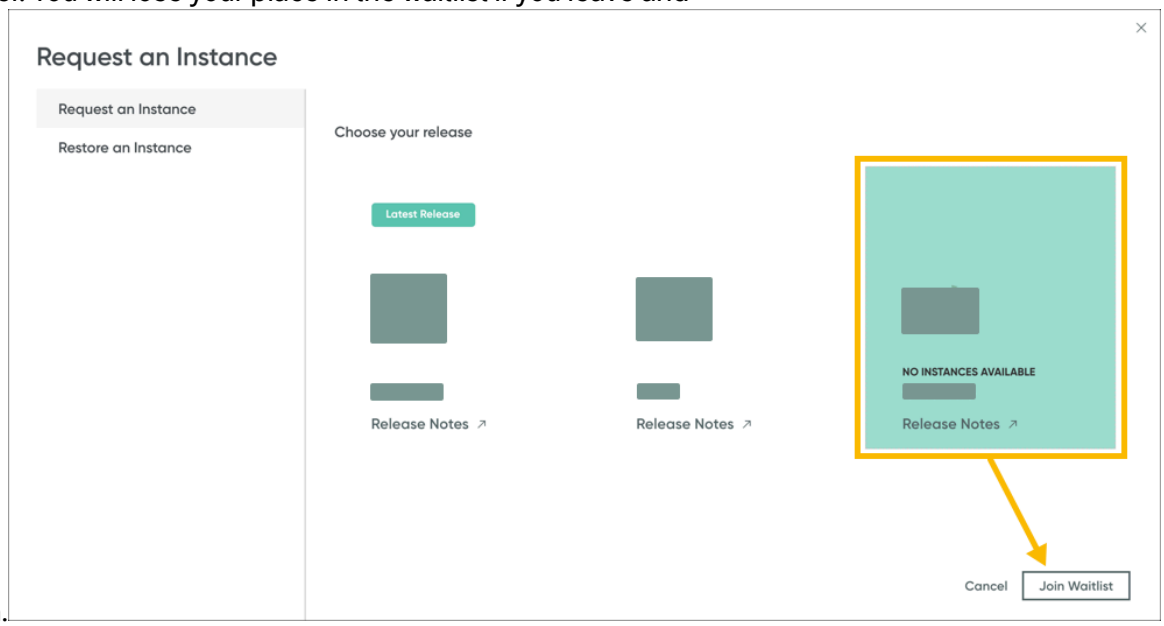
If you do not have a PDI because you released your PDI or went longer than ten days without activity, can request a new instance.

Procedure
1. Sign in to the Developer Site.
2. Click the **Request Instance** button in the header.
3. In the **Request an Instance** dialog, select a ServiceNow® release for the instance and click the **Request** button.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

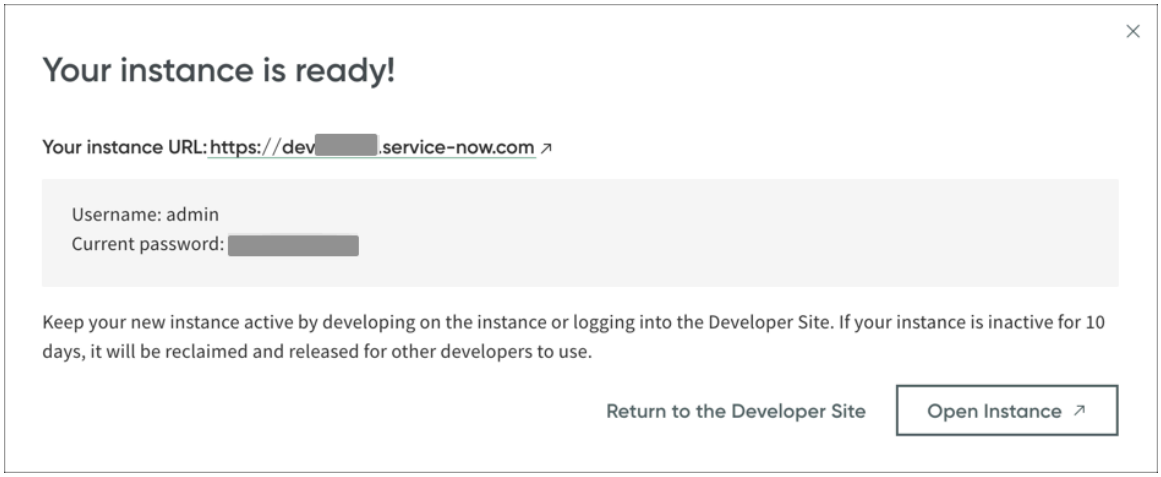


- PDIs are a limited resource. If no PDIs are available for the desired release, select the release card for the release and click the **Join Waitlist** button. You will be notified when a PDI becomes available.
- To leave the waitlist, click the **Leave Waitlist** link in the header. You will lose your place in the waitlist if you leave and rejoin.



4. When an instance is assigned to you, the **Your instance is ready!** dialog supplies your instance URL and password.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



5. Click the **Open Instance** button to open the instance in a new browser tab.

Accessing your PDI

Open your PDI and start building in the instance.

Before you begin

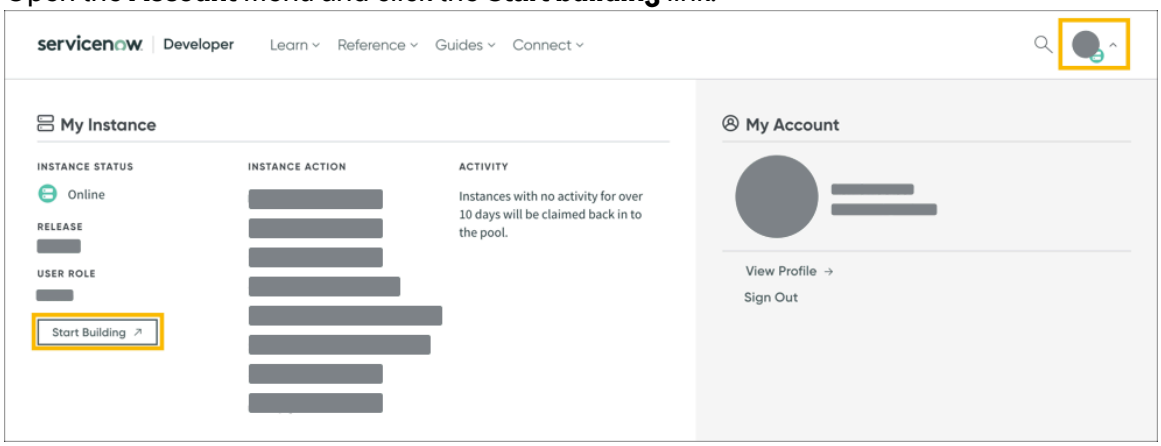
Role required: none

About this task

Once you have obtained a PDI, you can open the instance from **Account** menu.

Procedure

1. Sign in to the Developer Site.
2. Open the **Account** menu and click the **Start building** link.



3. If your PDI is hibernating, wait a couple minutes for your PDI to wake up before you can access it.
   Your PDI wakes from hibernation automatically when you sign in to the Developer site.
4. If your PDI hibernates while you are still signed in to the Developer Site, select the **Refresh Instance** instance action, then click the **Wake Instance** button to wake your PDI from hibernation.
5. If the instance is undergoing maintenance, you will need to wait for maintenance to complete before you can access your PDI.
6. If you have any issues accessing your PDI, review the Getting instance assistance section of this guide for steps you can take.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Activating a plugin from your PDI

One option to get started using an instance is to activate a plugin directly for your PDI.

Before you begin

Role required: none

About this task

You can activate most plugins directly in your PDI. Follow the instructions to Activate a plugin.

Procedure

1. On your PDI, open **All** > **System Definitions** > **Plugins**.
2. In the **Search** field, type the `<name of the plugin to install>`, for example, `app engine studio`.
3. On the plugin's card, click the **Install/Update All** button.
4. *Optional:*In the **Install** dialog, select the **Load demo data** option to install the plugin's sample data.
5. Select the **Install** button.
6. When the installation is complete, click the **Close** button in the **Install** dialog.

Activating a PDI plugin from the developer site

Activate your PDI plugin from the Developer Site to start working on an instance.

Before you begin

Role required: none

About this task

Some plugins require activation by ServiceNow® personnel on company and partner instances. You cannot activate these plugins from the plugins list. You may be able to activate these plugins from the **My Instance** section of the **Account** menu.

Note: Many ServiceNow® Store applications cannot be installed to PDIs.

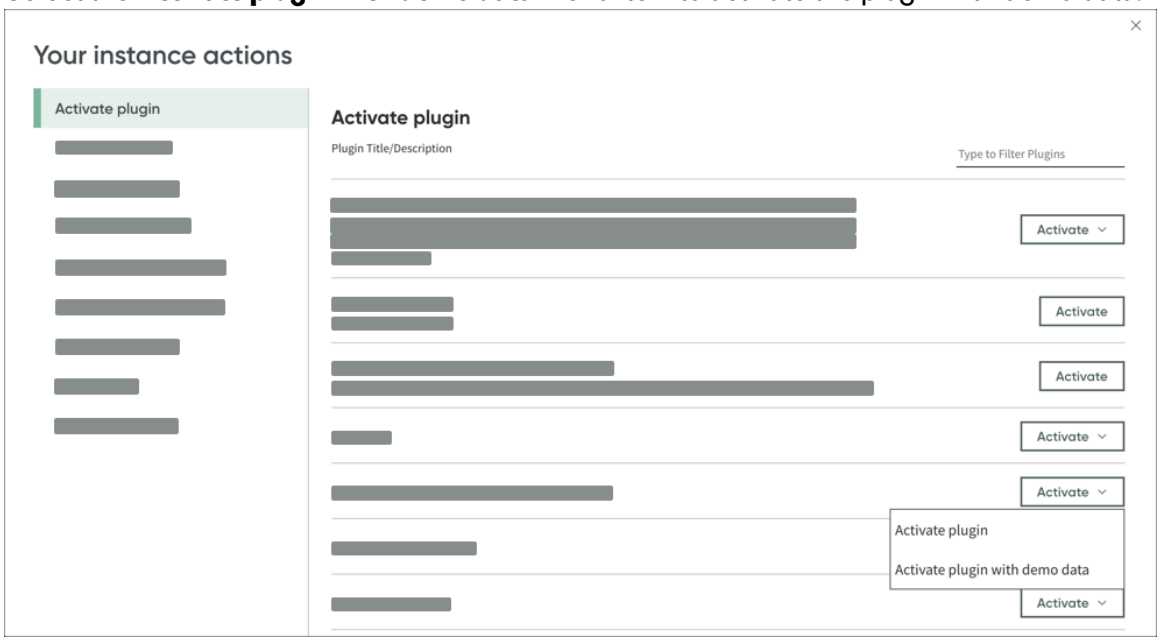
Procedure

1. Open the **Account** menu to access the **My Instance** section.
2. Click the **Activate plugin** instance action.
3. In the **Your instance actions** dialog, find the plugin and click the **Activate** button.
4. If your plugin has demo data:

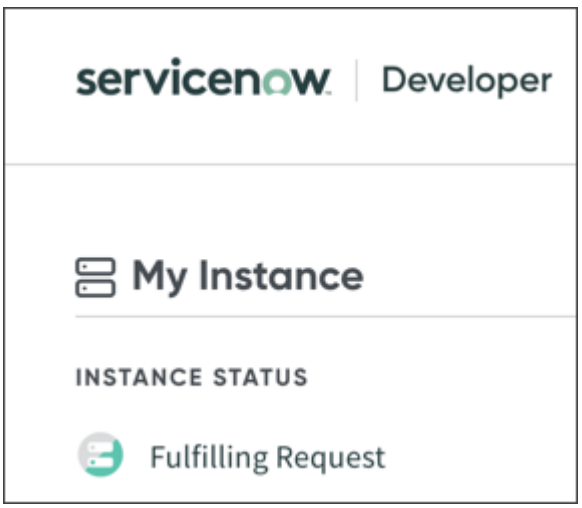
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. Select the **Activate plugin** menu item to activate the plugin without demo data.

b. Select the **Activate plugin with demo data** menu item to activate the plugin with demo data.



You cannot perform other instance actions while the plugin is activating. The **Instance Status** is **Fulfilling Request** while the plugin is activated.



Result

You will receive an email from the Developer Site when the plugin activation is complete. Plugins that require activation by ServiceNow® personnel do not appear in the plugins list in your PDI. The **Available Plugins** dialog does not show plugin activation status.

To use the plugin, reload the browser window where you logged in to your PDI.

Managing email properties for your PDI

Configure properties within your PDI to start sending and receiving emails.

Before you begin

Role required: none

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

About this task

ServiceNow® instances can send and receive email. Configuring email is typically the responsibility of a system administrator, not an application developer. In order to send and receive email with your PDI, email sending and receiving needs to be configured.

Unfortunately, personal developer instances (PDI) have been used both accidentally and maliciously to send email spam. In order to prevent spamming but still allow notifications on PDIs, we have restricted our email servers to sending to a single domain, guerrillamail.com. PDIs can receive email from any domain except guerrillamail.com.

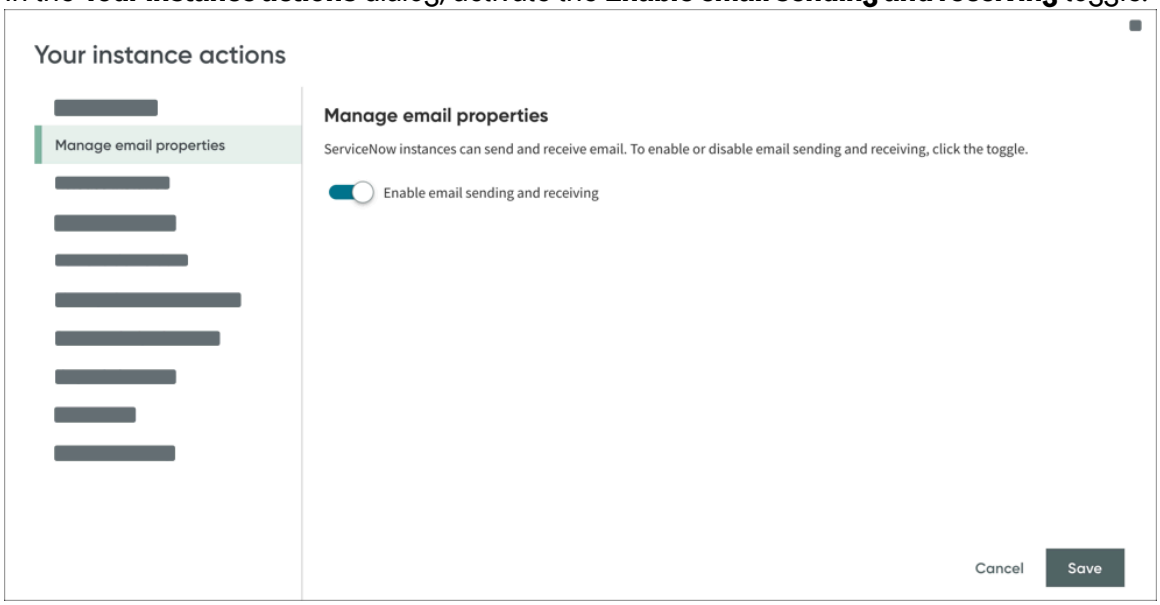
Procedure

1. To enable email sending and receiving from the Developer Site:

   a. Open the **Account** menu to access the **My Instance** section.

   b. Click the **Manage Email Properties** instance action.

   c. In the **Your instance actions** dialog, activate the **Enable email sending and receiving** toggle.



   d. Click the **Save** button.

2. To disable email sending and receiving, deactivate the **Enable email sending and receiving** toggle.

Releasing your PDI

Release your PDI to unassign the instance from you, reset to its original state, and reassign it to another Developer Program member.

Before you begin

Role required: none

About this task

If you no longer need your PDI or you are not going to be able to use it for an extended period of time, save your work and release the instance. When you release your PDI, the instance is unassigned from you, reset to its original state, and reassigned to another Developer Program

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

member. When you request an instance or restore an instance, you will receive a PDI with a different instance name and URL.

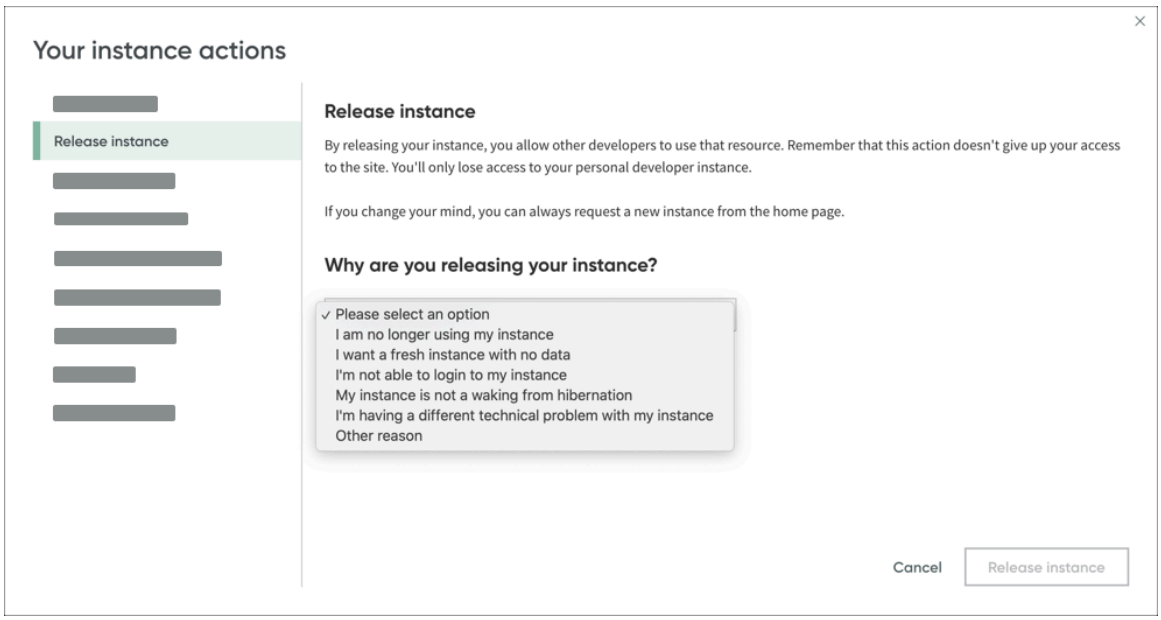
Procedure

To release your PDI:

a. Open the **Account** menu to access the **My Instance** section.

b. Click the **Release instance** action.

c. In the **Your instance actions** dialog, read the details and select a reason for releasing your instance from the choice list.



d. Click the **Release instance** button.

Changing your instance user role

Change your PDI user role to either App Engine Studio Creator or Admin for different views and levels of access.

Before you begin

Role required: none

About this task

Depending on your learning or testing needs, you may need to change the user role for your PDI.

Procedure

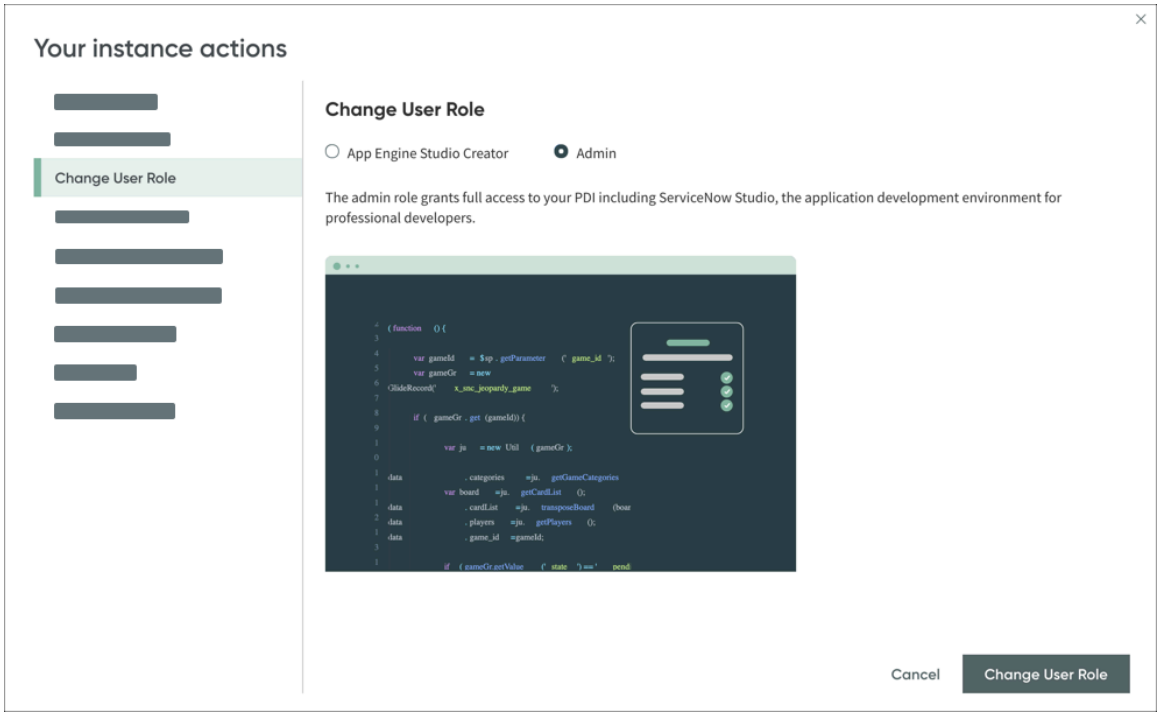
1. To change your user role:

a. Open the **Account** menu to access the **My Instance** section.

b. Click the **Change User Role** instance action.

c. In the **Your instance actions** dialog, select **App Engine Studio Creator** or **Admin**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



d. Click the **Change User Role** button.

e. A message indicates when your change user role request is completed successfully.

2. To see the change in your user role:

a. Close any open PDI browser tabs.

b. Relaunch your PDI by clicking the **Start building** button in the **Account** menu.

Removing demo data from your PDI

If you do not need the demo data in your PDI included by default, remove it from your PDI.

Before you begin

Role required: none

About this task

By default, PDIs include demo data. You can remove the demo data from your PDI on the Developer Site.

Procedure

1. Open the **Account** menu to access the **My Instance** section.

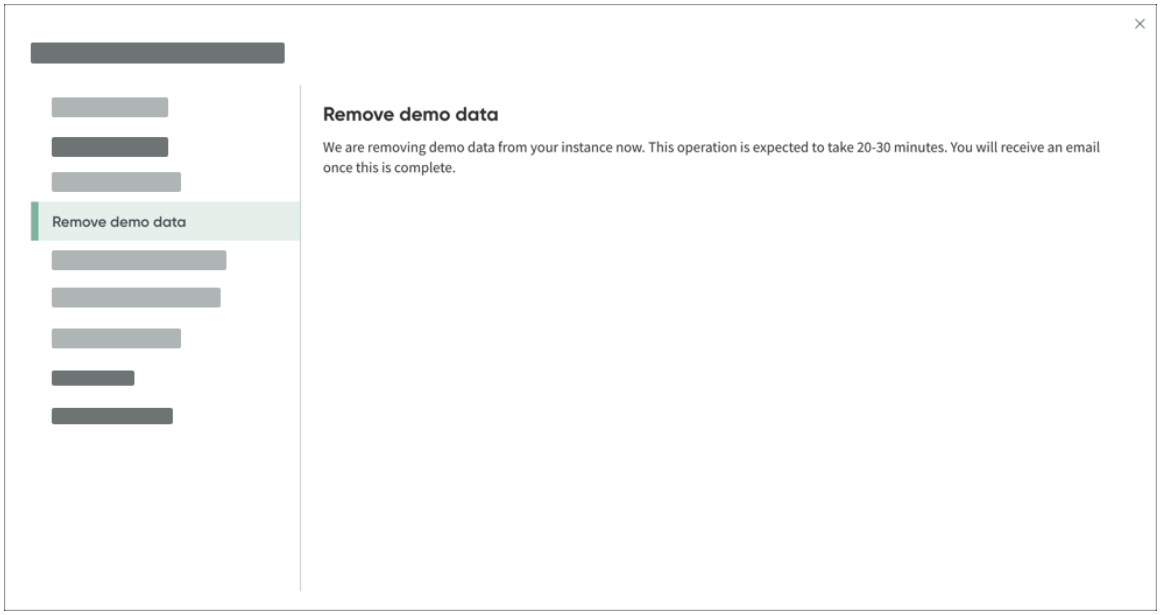
2. Click the **Remove demo data** instance action.

3. In the **Your instance actions** dialog, read the description and acknowledge that demo data cannot be restored click the **Remove demo data** button.

Result

Demo data removal takes 20-30 minutes. The Developer Site will send you an email when the process is complete.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Resetting your PDI to its initial state

Reset your PDI to its initial state to start work with a fresh instance.

Before you begin

Role required: none

About this task

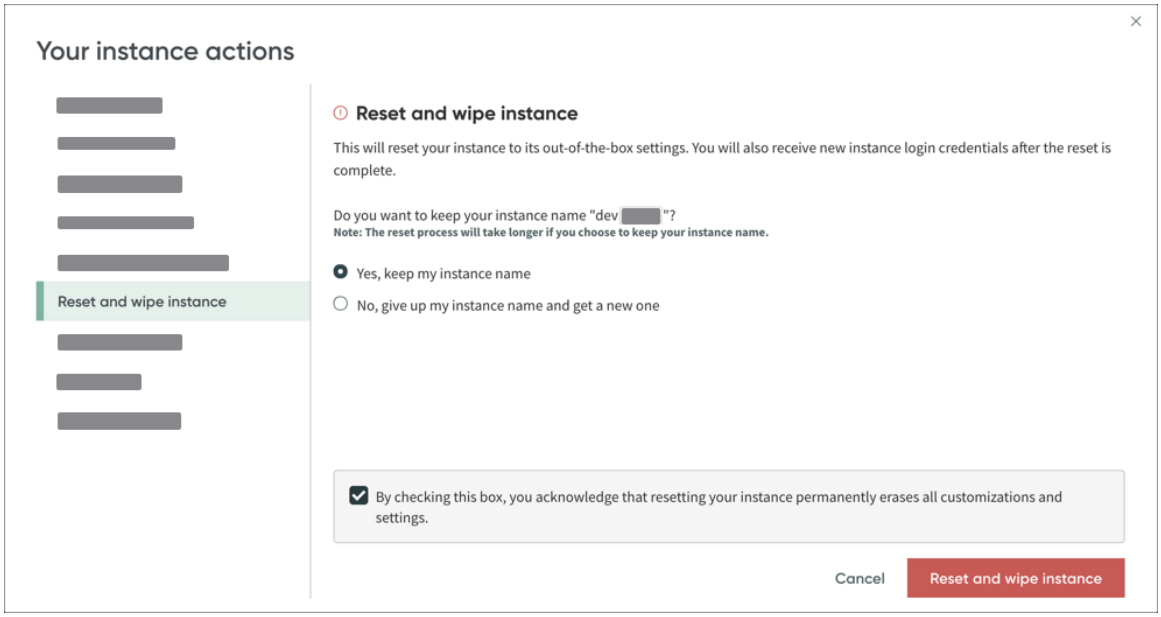
If you want to start work on a new application or want to work with a fresh instance, you can reset your PDI to its initial state.

Note: If you reset and wipe your instance, you will lose your customizations and configurations you made to the PDI. Be sure to back up any work you want to save by committing your work to source control or creating update sets.

Procedure

1. Open the **Account** menu to access the **My Instance** section.
2. Click the **Reset and wipe instance** instance action.
3. In the **Your instance actions** dialog, select how to reset your instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



- **Yes, keep my instance name**: Reset your instance to its default state. Resetting and wiping your instance may take a couple of hours to complete. You will receive an email when the process is complete.
- **No, give up my instance name and get a new one**: Get a new instance without waiting for the reset process.

4. Select the **By checking this box, you acknowledge that resetting your instance permanently erases all customizations and settings** disclaimer.
5. Click the **Reset and wipe instance** button.

Upgrading your PDI

Apply a patch or new release to your instance to upgrade your PDI.

Before you begin

Role required: none

About this task

When ServiceNow® releases patches and new releases, you can apply the patch or release to your PDI from the Developer Site. You have two options for upgrading a PDI.

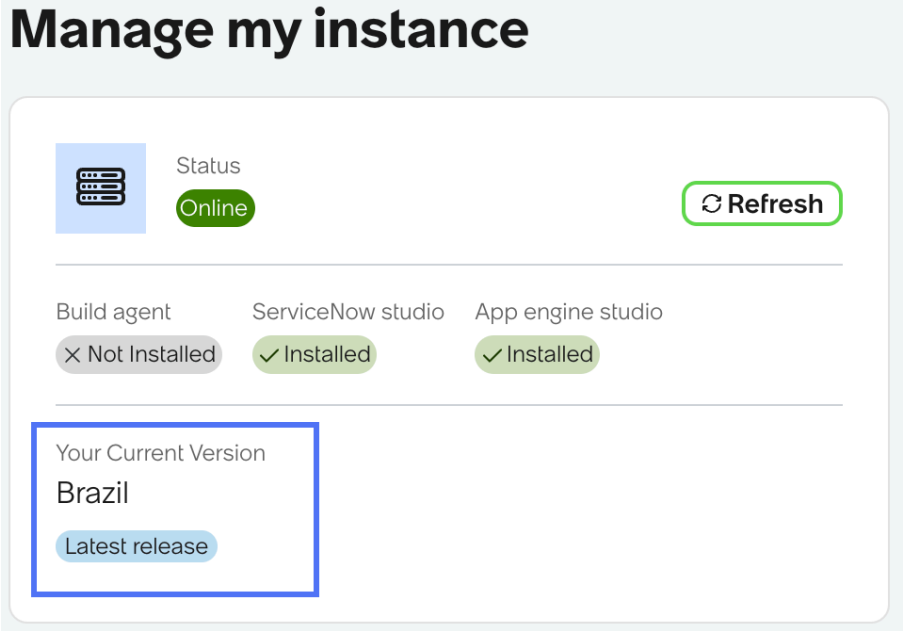
Note: Patches for a release are applied to your PDI automatically up to once a month to maintain compliance with the ServiceNow Patching Program. Sometimes patches may be made available to apply outside the regular patching schedule. Use the **Upgrade instance** option to apply a patch manually.

Procedure

1. Log in to the Developer Site.
2. Select **Manage my instance**.
3. In the first panel, select **Upgrade release**.

   The upgrade release button is only displayed if a later release version is available. A label is displayed if your current version is the latest release.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



4. Select the patch or release to apply to your PDI.
5. Select **Upgrade**.

Result
You receive an email when the upgrade is complete.

Getting instance assistance

Get assistance for common issues accessing an instance and give feedback.

If you cannot access your PDI or get the **Your instance is currently offline** page, your PDI may be in the process of waking up from hibernation or your PDI just completed patching and needs to restart. Your PDI should be available within five minutes.

If your PDI is not available after five minutes, sign in to the Developer Site, select the arrow next to your avatar, and check the **My Instance** page to see if your PDI is still undergoing maintenance.

Early Availability guide

Early Availability provides you with exclusive access to the latest release over a month before market launch makes the release available for any customer to use.

The ServiceNow Developer Program provides Early Availability to pre-release versions. You have access to pre-release versions and resources through the ServiceNow Developer Site. As a Developer Program member, Early Availability provides you with exclusive access to the latest ServiceNow release over a month before market launch makes the release available for any customer to use. Early Availability includes:

- Personal developer instances (PDI)
- Learning plans
- Training courses
- APIs
- Documentation

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

As a Developer Program member, you have the first opportunity to become familiar with the latest ServiceNow release through Early Availability. Early Availability gives you an opportunity to learn skills that will set you apart from other developers in the industry. You can leverage the latest features and upgrades to build new applications.

Early Availability learning plans and training courses

Early Availability gives you access to learning plans and training courses specifically for the latest release.

The Developer Program provides you with the training needed to successfully explore releases at your own pace. Early Availability gives you access to learning plans and training courses specifically for the latest release. Pick from a selection of easy-to-follow training courses or learning plans based on your current knowledge and experience. With either choice, you are guaranteed to build or improve your application development skills on the latest ServiceNow AI Platform®.

Early availability API documentation

Early Availability includes access to API documentation for the latest release.

The Developer Program also contains API documentation on:

- Server-side APIs (scoped and global)
- Client-side APIs
- REST APIs
- Now Experience UI Framework

Server-side scoped APIs are for use within scoped applications, and may behave differently within the global scope. Server-side legacy APIs are documented for development work in global scope. New applications should be built using scoped APIs.

Early Availability includes access to API documentation for the latest release.

Low-code versus pro-code development

Learn the difference between low-code and pro-code solutions on the ServiceNow AI Platform.

Which app builder should I use?

Use the following table to choose the app building experience that matches your role and technical background.

App builders

Tool	Users	Features
Build Agent: Build an app using natural language prompts.	Developers of all experience levels (from low-code to Platform and source code)	Build full-stack applications via a conversational interface. You can use Build Agent in ServiceNow Studio and the ServiceNow IDE to generate a complete ServiceNow scoped or global application. For more

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

App builders (continued)

Tool	Users	Features
		information, see Exploring Build Agent.
ServiceNow Studio: Build and deliver apps in a unified environment.	Citizen developers, Platform developers	Build apps in a unified development environment. ServiceNow Studio provides streamlined navigation, integrated low-code tools, and built-in tracking and packaging so you can develop and ship apps faster. Use Build Agent in ServiceNow Studio to create and update apps with a conversational interaction. For more information, see Exploring ServiceNow Studio.

What is low-code development

Low-code development is a new approach to app creation that allows users with limited coding experience to create powerful apps. Low-code development platforms rely on graphical interfaces and configuration instead of manual coding. These new low-code development platforms enable more people to create and deploy apps quickly and efficiently.

Benefits of low-code development

Low-code app development streamlines the development process to build more apps faster. Low-code solutions require fewer developers, and allow non-developers to build apps. Pre-built templates provide developers a head start building apps. System administrators can manage app development from a single location and collaborate with other developers. Decrease the time that it takes to deploy apps using predefined workflows in the ServiceNow AI Platform.

ServiceNow no-code and low-code development tools

- Creator Studio
- App Engine Studio
- UI Builder
- Table Builder
- Flows in Workflow Studio
- Workspace Builder
- Exploring Decision Tables

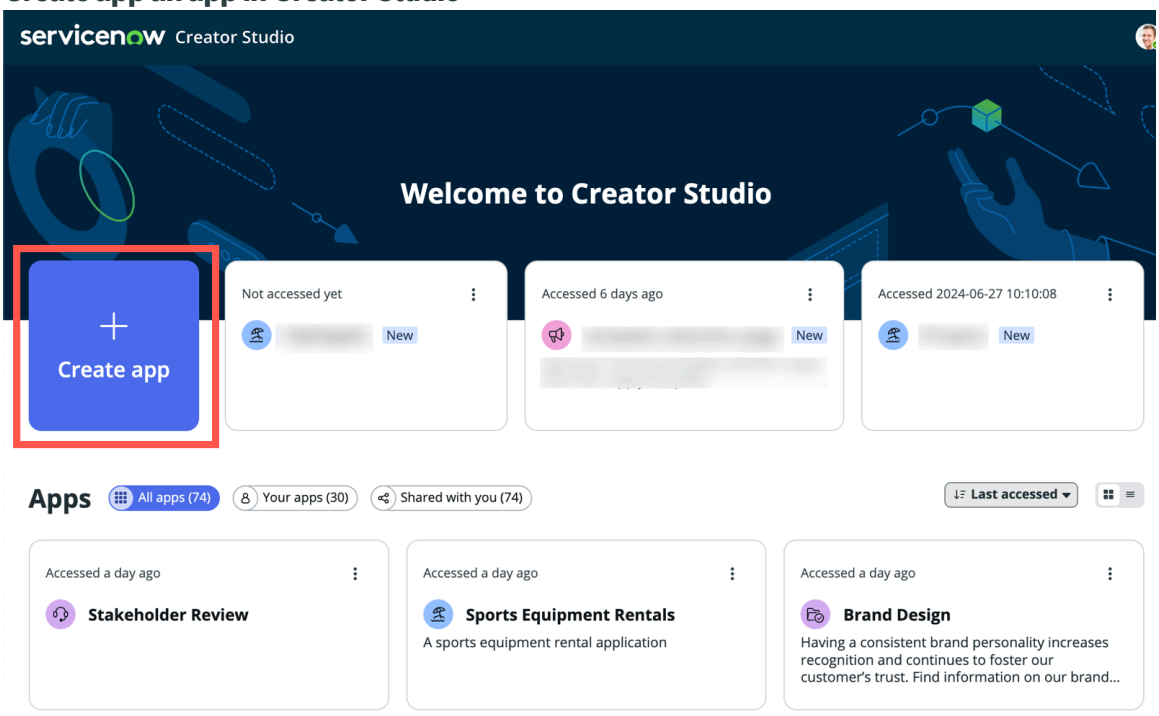
For more information on low-code development tools, see .

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

No-code development tool example

Creator Studio makes creating basic request-fufillment apps easier by dividing their creation into simple steps. You can create forms for users to request catalog items and use form submissions to initiate automated playbooks. Find out more in Creator Studio.

Create app an app in Creator Studio



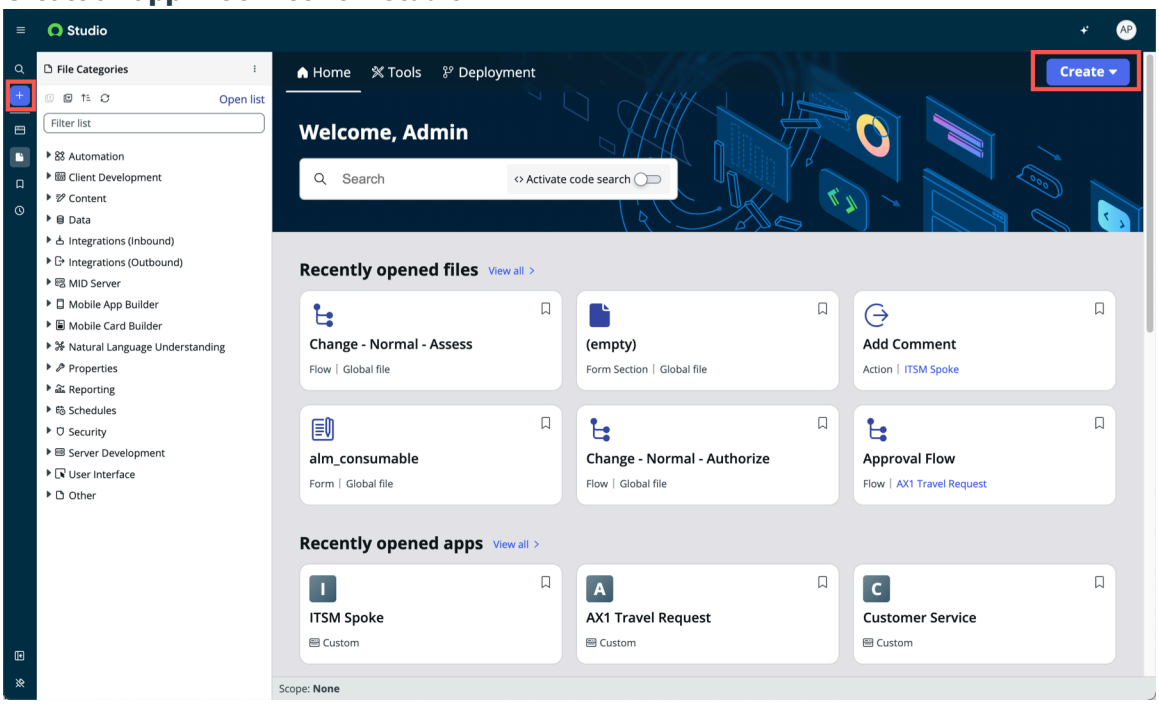
What is mid-code development

If you're comfortable with the ServiceNow AI Platform and some development tools, use the new ServiceNow Studio to access all of the builders and development tools in one place.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Mid-code development tool example

Create an app in ServiceNow Studio



What is pro-code development

Pro-code development is used by developers to create complex apps that can't be built with a low-code tool. Traditionally, pro-code development is used to create apps from scratch using custom code to solve a business need. Developers need to have knowledge of coding and how to use programming languages to build apps.

Benefits of pro-code development

The advantage of pro-code development is being able to create custom apps without the limitation of a tool. You can build custom apps unique to your business needs without limits. Developers can create apps with a custom look and feel to match your company's branding.

ServiceNow pro-code development tools

- ServiceNow IDE
- ServiceNow SDK
- Scripting
- REST APIs
- ServiceNow Extensions for Visual Studio Code

For more information on pro-code development tools, see Building pro-code applications.

Low-code versus no-code

The terms low-code and no-code tend to get used interchangeably, but they aren't exactly the same thing. While both low-code and no-code solutions provide tools for simplified app development, the differences are worth considering.

Low-code

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Low-code platforms are designed for professional developers and non-technical business users. They require very little training or experience and use visual-based modeling to streamline the development process. They also allow people with coding experience to dive deeper, coding by hand when needed.

No-code

No-code platforms require no development experience, and are designed specifically for citizen developers and business users. No-code solutions open app development up to essentially everyone, but can lead to shadow IT—unsanctioned app development within an organization.

Modifying versus building an application

At a certain point in the lifetime of your applications, it might make sense to create an application to replace the old one. Cost savings, maintenance effort, and features available in new apps can help you determine if it's better to build a new application or modify an existing one.

Consider building a new application for the following reasons:

- The existing application is outdated and doesn't run well on newer hardware.
- Building a new application would produce cost savings over updating and maintaining the existing application.
- The existing application doesn't fundamentally meet your business needs.
- The existing application's features can easily transfer to a new application.
- Updating the existing application would cause significant disruptions to your UI and workflow because the existing application does not support the features you want to add.
- A new application would resonate with more users through an updated UI, more features, and ease of use than the existing application.
- Building a new application would be a quicker and easier process on low-code platforms like App Engine Studio (AES) than maintaining the existing application.

For more information, see the following topics:

- Use App Engine instead of customizations
- Customization vs configuration with ServiceNow Studio

Understand the ServiceNow® UI experiences

You can develop in the following ServiceNow AI Platform user interfaces: the Next Experience UI, the Workspace UI, the Classic Environment, and the Core UI. This topic explains these user interfaces.

Use the information in the following table to select the best UI experience for you and your team.

UI experiences on the ServiceNow AI Platform

UI experience	Benefits
Next Experience UI	**Unified Navigation** • Access app shell items—such as the contextual app pill, menu items, and search—all in one place across the entire

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

UI experiences on the ServiceNow AI Platform (continued)

UI experience	Benefits
	platform, no matter which application you're using. • Pin and unpin navigation menus to access menu items or free up screen real estate. **Additional menus** • Use the Favorites, History, and Workspaces menu items to navigate to and pin important resources. • Easily switch between Core UI applications and configurable workspaces. • Find your recently viewed items in a single menu. • Create custom menus for your end users. For more information, see Configure custom menus for Unified Navigation.
Configurable Workspace UI	**Improved agent efficiency** Work on multiple issues concurrently in an optimized, intuitive layout. **Speedy resolution** Resolve issues faster with automated suggestions powered by machine learning. **Proactive identification of major incidents** Get notified of potential major incidents based on issue frequency and impact. **Greater visibility for agents** Keep informed of updates and surface important insights with a live activity feed and analytics.
Working in the classic environment	**Work in lists and forms** Work in the classic ServiceNow AI Platform environment, creating lists from records. **Powerful app creation and development**

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

UI experiences on the ServiceNow AI Platform (continued)

UI experience	Benefits
	Create applications and develop with the full power of the ServiceNow AI Platform.
Working in Core UI	The Core UI interface provides an updated look and usability improvements. Notable features include real-time form updates, user presence, a redesigned application navigator with Favorites and History tabs, and improved activity streams. Core UI is the default user interface for new instances. For upgraded instances, an administrator must activate Core UI.

Support for developers

Here's where ServiceNow developers get support.

Where to find developer support

Resource	Description
ServiceNow Community	• Product hubs - Choose a product area and: ○ Connect to resources and learn about integrations. ○ Get expert and peer advise. ○ Network with peers. • Developer discussions - Connect with developers like you to ask questions, offer solutions, or build together. • Developer blog - See what the developer advocate team has to say.
Developer Site	• Sign up and start building on the ServiceNow AI Platform with a free account. • Join the ServiceNow developer program. • Access learning resources. • Access product and release documentation, APIs, CLI, and Component docs. • Access guides with an overview of the ServiceNow AI Platform, as well as

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Where to find developer support (continued)

Resource	Description
	information on how to get started whether you're a new or experienced developer. • Learn ways to connect and collaborate with others working on similar problems.
ServiceNow training	Get started with training and certification.
ServiceNow developer videos	Access the ServiceNow Developer Program YouTube channel.

Application collaboration

People with admin privileges can manage the permissions of developers collaborating on an application.

Collaboration descriptors

You can use the predefined collaboration descriptors that are standard with activation, or create a custom collaboration descriptor. By using a collaboration descriptor, you can assign, manage, and monitor delegated development permissions.

Collaboration descriptors

Descriptors	Description
Owner	Owner of the application that manages other collaborators and can delete the application.
Editor	Standard descriptor to invite collaborators.
Custom	Non-standard (custom) descriptor created by the user.

Development permissions

By setting permissions, you can retain control over the system. You assign permissions to developers (or users who deploy applications) so that they can develop or deploy applications.

Development permissions

Permissions	Description
Application Management	Capabilities to delete the application and version source control access.
File type Access	File types that the user has access to.
Security / Entitlement	Capabilities to manage application security.
Programming Tools	Capabilities for script fields access.
Deployment	Deployment capabilities.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Application collaboration users

You can assign a collaboration descriptor for an individual by creating an app collaboration user. The user can only be assigned one collaboration descriptor for any application.

Application collaboration groups

You can assign a collaboration descriptor for a group of users by creating an app collaboration group. The group can only be assigned one collaboration descriptor for any application.

Install the app collaboration application

Install the app collaboration application so that you can view the collaboration feature in the UI.

Before you begin

Download the app collaboration application so that you can view the collaboration feature.

You must have an App Engine license.

Role required: admin

Procedure

1. Navigate to **System Applications** > **All Available Applications** > **All**.
2. Click the search icon () in the middle of the screen to search for the application collaboration application.
3. Click **Install**.

   Note: If app collaboration is already installed, the **Install** button is inactive.

4. Click **Activate**.

Result

The app collaboration feature is active.
After you install and activate the app collaboration plugin, all delegated developers will receive custom collaboration descriptors based on their current permissions and applications. The base system table provides standard Owner and Editor collaboration descriptors which are used in the collaboration feature.

If you install this plugin without App Engine Studio (AES) for an application, you should manage the delegated development permissions with this plugin or use the old Manage Developers screen to manage the permissions. Do not use both the plugin and the Manage Developers screen to manage the delegated development permissions.

To use the old Manage Developers screen, select your application and select Manage Developers in the related links on the form.

Create collaboration descriptors to assign permissions

Create descriptors in the collaboration application so that you can assign permissions to users or groups.

Before you begin

Role required: admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

About this task

Note: You should create collaboration descriptors in addition to Owner and Editor in the global scope. If you want collaboration descriptors to appear and be used in AES, you should also set them to `standard = TRUE`. AES doesn't support collaboration descriptors that are created in custom scopes, and non-standard collaboration descriptors don't render in AES.

Use the predefined collaboration descriptors that are standard with activation, or create a custom collaboration descriptor. Collaboration descriptors enable you to assign, manage, and monitor delegated development permissions for each application or uniformly across multiple applications. You can assign each collaborator with one collaboration descriptor for an application. However, users can have multiple collaboration descriptors simultaneously if they belong to groups where collaboration descriptors have been assigned.

Procedure

1. Navigate to **All** > **App Engine** > **Collaboration** > **Descriptors**.
2. Click **New**.
3. On the form, fill in the fields.

App Collaboration Descriptor form

Field	Description
Name	Name for the descriptor.
Description	Description of the descriptor that includes permissions capabilities. Examples are as follows: ○ Owner ○ Editor
Application	Scope of the collaboration descriptor. Currently, all collaboration descriptors must be created into the global scope.
Standard	Option for inviting other collaborators with this role.

4. Click **Submit**.

Add permissions to collaboration descriptors

Add permissions to collaboration descriptors to manage your user's capabilities, such as the ability to delete the application or manage collaborators.

Before you begin
Role required: admin

Procedure

1. Navigate to **All** > **App Engine** > **Collaboration** > **Descriptors**.
2. To modify the permissions, select a collaboration descriptor from the related list.
3. To view the permission's options, click **Edit...** in the related list.
4. To add or remove permissions to the descriptor, click the arrows.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For a list of permissions, see Collaboration permissions.

5. Click **Save** and **Update**.

Assign collaboration descriptors to users

Assign collaboration descriptors to your users for a specific application so that you can define specific permissions for these users.

Before you begin

Role required: admin

Procedure

1. In the related list, click **Application collaboration users**.
2. Click **New**.
3. On the form, fill in the fields.

Collaboration users form

Field	Description
Collaboration Application	Application that the user works on.
Collaboration User	User that works on the application.
Collaboration Descriptor	Descriptor for the user on this application.

4. Click **Submit**.

Assign collaboration descriptors to groups

Assign collaboration descriptors to user groups for a specific application so that you can define specific permissions for these user groups.

Before you begin

Role required: admin

Procedure

1. In the related list, click **Application collaboration groups**.
2. Click **New**.
3. On the form, fill in the fields.

Collaboration groups form

Field	Description
Collaboration Application	Application that the group will work on.
Collaboration Group	Name of the collaboration group.
Collaboration Descriptor	Descriptor for the group on this application.

4. Click **Submit**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Collaboration permissions

Assign permissions to define what collaborators can do.

Collaboration permissions

Permission	Description
Playbooks	Grants access to work with the Playbooks design environment to create processes. Editing activity subflows or actions requires the **Flow Designer** permission.
All Metadata	Grants access to work with all metadata.
Integrations	Grants access to web service APIs, REST APIs, data sources, and Integration Hub - Import.
UI Builder	Grants access to work with UI Builder to build out more complex interfaces.
Mobile builders	Grants access to build mobile experiences, such as with Mobile App Builder.
Security Management	Grants access to security management files, such as Access Control Lists and roles.
Service Portal	Grants access to work with Service Portal editors and tools.
Notifications	Grants access to create automatic email notifications and work with email templates.
Invite collaborators	Grants access to invite developers to collaborate on an app.
Workflow	Grants access to the Workflow Editor and Activity Creator.
Reporting	Grants access to reports and scheduled reports.
Workflow Studio	Grants access to the Workflow Studio design environment to create flows and actions.
Service Catalog	Grants access to work with catalog-related file types such as catalog items, record producers, and variables to add catalog items to apps.
Tables & forms	Grants access to model and layout-related file types such as table columns, form layout, and list layout.
Decision Tables	Grants access to work with Decision Tables to create decision logic based on multiple if-then rules.

Contextual development environment

The platform is a contextual development environment that displays the currently selected application, identifies the scope of every application artifact, and prevents any changes that violate the access settings for an application.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Application developers can use the contextual development environment to:

- Determine the application context
- View and select applications
- Enforce application version standards
- Enforce application resource throttling
- Enforce script protection policies

Application context

When application developers create new records, the system automatically assigns the records to the currently selected application in the application picker.

When application developers attempt to change existing records, the platform checks whether the currently selected application matches the scope of the application artifact. If they match, the application developer can save changes to the artifact. If they differ, the system makes the following changes to the user interface:

- Makes all the fields on the current record read-only.
- Displays a warning message that the application artifact belongs to another scope.

Application access settings

Application access settings determine whether one application can access resources from another application.

Application access settings are similar to access controls (ACLs) in that they allow you to restrict access to certain resources, but instead of restricting tables and records from users they restrict applications resources from other applications. There are several ways to set cross-scope access.

Setting type	Use	Description	Access setting location
Application design and runtime settings	Your application or application script requires access to a cross-scope resource.	Application designers can configure the following application access settings for the entire application. • Specify whether cross-scope tables can be selected during design-time activities. • Specify when scripts can run on cross-scope resources. • Specify what JavaScript standard the application supports.	Access determined by the cross-scope privilege record owned by the calling application.
Table design and runtime settings	A cross-scope application or web service requires access to perform CRUD operations on a table.	Application developers can also configure application access settings for individual tables.	Access determined by settings on the target table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Setting type	Use	Description	Access setting location
		• Specify whether the table is available to other application scopes. • Specify what runtime operations from other application scopes the table supports. • Specify whether the table can be selected during design-time activities. • Specify whether the table can be accessed from web services.	
Restricted caller access privilege settings	A cross-scope application or script requires access to your application or application resource.	Admin users can configure the following application access settings for an entire application scope or application resource. • Specify whether a cross-scope script can access your application scope or an application resource. • Track cross-scope requests for access to application resources. • Approve or deny cross-scope requests to access application resources.	Access determined by the restricted caller access record owned by the target application.

Application design and runtime settings

The application design and runtime settings determine whether an application can access cross-scope resources.

Design and Runtime fields

Field	Description
JavaScript Mode	The JavaScript standard that the application supports. Select **ECMAScript 2021 (ES12)** to support features in ECMAScript 12th edition or **ES5 Standards Mode** to support features in ECMAScript 5th edition. Select **Compatibility Mode** to support earlier ECMAScript editions. For more information, see Set the JavaScript mode for an application.
Runtime Access Tracking	The application's handling of script access requests to resources in other applications. Select **None** to authorize all access requests to cross-scope resources without logging them. Select **Tracking** to log and authorize all access requests to cross-scope resources. Select **Enforcing** to log access requests to cross-scope resources but require an administrator to authorize each request.
Restrict Table Choices	The availability of cross-scope tables when designing the application. Clear the option to allow the application to see tables from other application scopes. Select the option to restrict design choices to only tables in the same application.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Runtime access tracking

Runtime access tracking allows administrators to manage script access to application resources by creating a list of script operations and targets that the system authorizes to run.

Runtime access tracking provides the following options:

Runtime access tracking options

Option	Tracking done	Authorization done	Results
None	The system does not track runtime access requests.	The system does not require authorization to run access requests.	Application scripts can access resources from other applications as long as the table-level access settings allow it.
Tracking	During development, the system creates a Cross-Scope Privilege record for each runtime access request. After installation, the system no longer tracks new runtime access requests.	The system sets the status of the Cross-Scope Privilege record to **Allowed**.	The system runs the tracked operation as long as the table-level access settings allow it.
Enforcing	During development, the system creates a Cross-Scope Privilege record for each runtime access request. After installation, the system no longer tracks new runtime access requests.	The system sets the status of the Cross-Scope Privilege record to **Requested**.	The system blocks the tracked operation from running until an Administrator manually changes the status to **Allowed** and the table-level access settings allow it.

During development, application designers must run all of an application's script logic to ensure the system tracks and authorizes the access requests to other applications.

Cross-scope privilege record

Runtime access tracking uses cross-scope privilege records to determine which script operations and targets the system allows to run.

The system creates cross-scope privilege records when:

- Runtime access tracking is set to **Tracking** or **Enforcing**.
- A script attempts to access another application.

Each cross-scope privilege record in the Cross scope privileges [sys_scope_privilege] table contains the following information.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Cross-scope privilege fields

Field	Description
Source Scope	The application requesting runtime access to another application's resources.
Target Scope	The application whose resources are being requested.
Target Name	The name of the table, script include, or script object being requested.
Target Type	The type of request: table, script include, or script object.
Operation	The operation the script performs on the target. The target type determines the available operations. Tables support the read, write, create, and delete operations. Script includes and script objects only support the execute API operation.
Status	The authorization for this record: requested, allowed, or denied

Administrators can manually create cross-scope privilege records for application developers in advance to communicate which cross-scope resources they expect developers to access. For example, administrators could create these cross-scope privilege records to permit application developers access to resources from Incident Management.

Sample cross-scope privilege records

Source Scope	Target Scope	Target Name	Operation	Status
My App	Global	incident	Read	Allowed
My App	Global	incident	Write	Allowed
My App	Global	ScopedGlideRecord	Execute API	Allowed

During testing, application developers should run all of their application scripting logic to ensure the system creates any necessary cross-scope privilege records. After application publication, the system only allows runtime requests to run that have a valid cross-scope privilege record.

Note: Table privilege granting is limited to, at most, the permissions set on the table object (sys_db_object) record. For example, granting a scope privilege to delete for table incident would not be allowed if the table object for incident did not allow Can delete scopes.

Application design access record

Administrators use application design access records to specify which other applications are available to developers during application creation.

When administrators restrict an application's table choices, they can then create application design access records to grant developers access to the tables of selected applications.

Each application design access record contains the following information.

Application design access fields

Field	Description
Source Scope	The application requesting design access to another application's tables.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Application design access fields (continued)

Field	Description
Target Package	The application whose tables will be available for design time access.
Application	The application to which this record belongs.

Application design access records allow administrators to have oversight of the resources available to application developers. When creating configuration records, application developers can only select tables from another application if there is an application design access record granting them access to the application. For example, administrators could create the following application design access records to grant developers access to tables from Incident Management and Problem Management.

Sample application design access records

Source Scope	Target Package
My App	Incident
My App	Problem Management

After developers create configuration records to other applications, the system displays these applications as dependencies.

Table design and runtime settings

The Application access fields determine whether a table is accessible to other applications during design-time or run-time operations.

Application access fields

Field	Description
Can read	Select the check box to allow script objects from other application scopes to read records stored in this table. This option offers runtime protection. For example, a script in another application can query data on this table. You must first select read access to grant any other API record operation.
Can write	Select the check box to allow script objects from other application scopes to modify records stored in this table. This option offers runtime protection. For example, a script in another application can modify a field value on this table. This option is available only when the **Can read** check box is selected. Clear the check box to prevent script objects from other application scopes from modifying data stored in this table.
Can create	Select the check box to allow script objects from other application scopes to create records in this table. This option offers runtime protection. For example, a script in another application can insert a new record in this table. This option is available only when the **Can read** check box is selected. Clear the check box to prevent script objects from other application scopes from creating records in this table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Application access fields (continued)

Field	Description
Can delete	Select the check box to allow script objects from other application scopes to delete records from this table. This option offers runtime protection. For example, a script in another application can remove a record from this table. This option is available only when the **Can read** check box is selected. Clear the check box to prevent script objects from other application scopes from deleting records from this table.
Allow access to this table via web services	Select the check box to allow users to make inbound web service queries to this table. This option offers both design-time and runtime protection. The user performing the query must have the correct permissions to access this table, even when this check box is selected. Clear the check box to prevent users from making web service queries to this table.

Runtime access to applications tables

Runtime access determines if an API or web service call can run against an application table.

Access permissions can be set for the following access points.

Runtime access points

Access points	Description
Script API	Any supported object or method call from the scoped system API such as a GlideRecord call.
Web services	Any supported web service call such as a REST, JSON, or SOAP web service.

The system does not prevent you from creating API or web service calls to the application tables, rather it determines if the API or web service call is allowed to run against the application table. API or web service calls that violate the access permissions for an application table produce an error. For example, making a web service call to a protected application table produces a 403 Forbidden HTTP error.

Default runtime access permissions

The default runtime access permissions apply to new application data tables.

By default, new application tables only allow read access from other application scopes.

Default runtime access

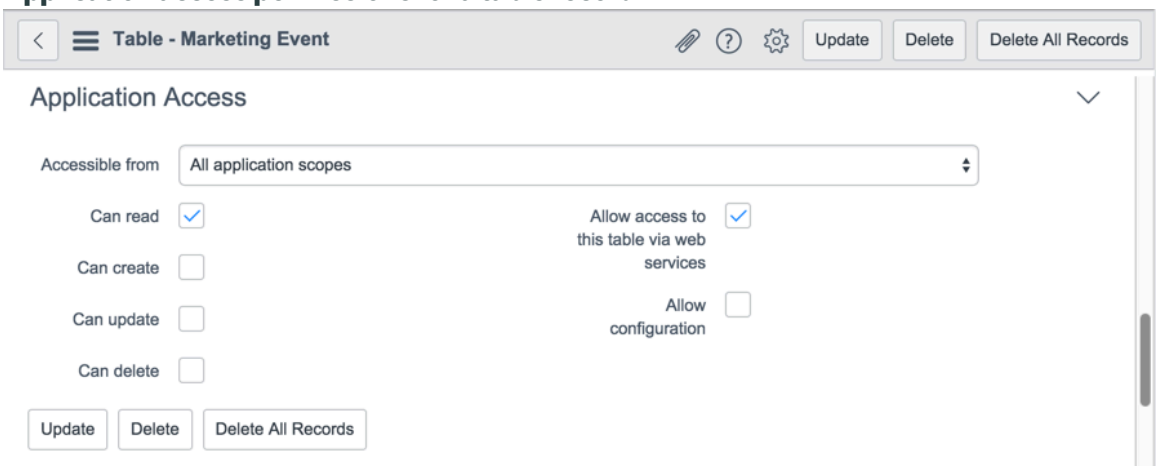
Field	Value
Accessible from	All application scopes
Can read	Enabled
Can create	Disabled

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Default runtime access (continued)

Field	Value
Can update	Disabled
Can delete	Disabled
Allow access to this table via web services	Enabled

Application access permissions for a table record



Set runtime access to application tables

Set these access permissions to protect application tables at runtime.

Before you begin

Role required: admin

Procedure

1. Navigate to **All** > **System Applications** > **Applications**.
2. Click the button for the application type you want to edit.

Option	Description
Developed	displays applications created on this in stance.
Downloaded	displays applications downloaded on this in stance.

3. Click the application name or the **Edit** button for the application you want to work on.
4. From the **Tables** related list, select the table whose access permission you want to set.
5. From the **Application Access** section, set the runtime access permissions.
6. Click **Update**.

Example denying all runtime access to a table

You can prevent script API and web service calls from other application scopes.

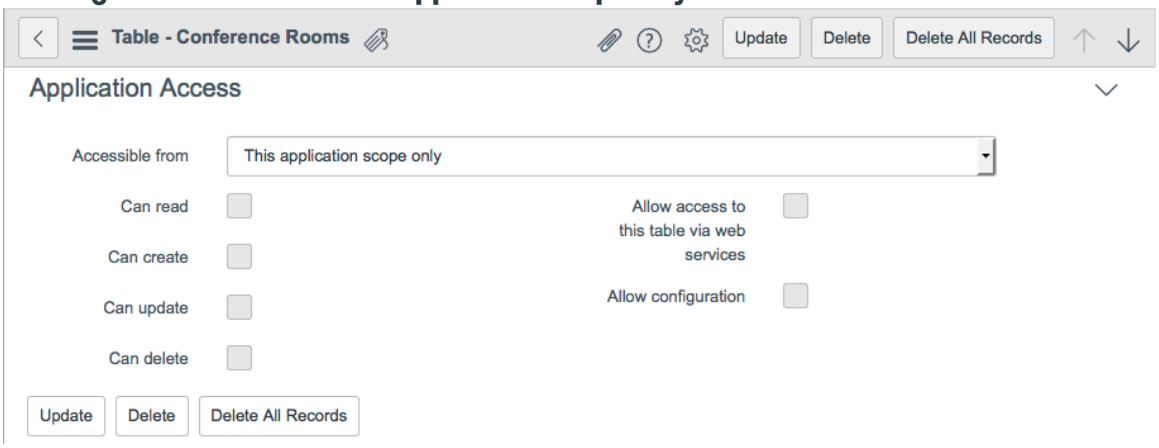
Typically, this is to prevent any other application from creating or modifying data in the table. Denying access requires setting the following value in the table record.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Denying all runtime access

Field	Value
Accessible from	This application scope only
Can read	Disabled
Can create	Disabled
Can update	Disabled
Can delete	Disabled
Allow access to this table via web services	Disabled

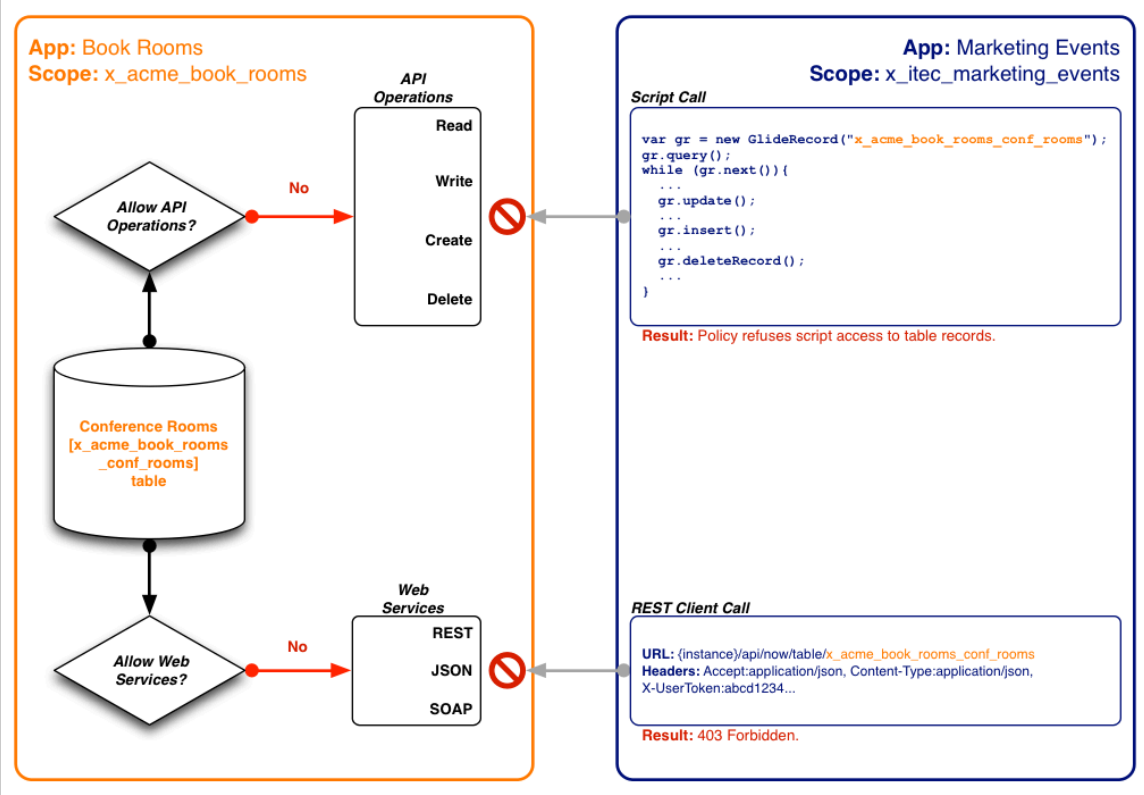
Limiting runtime access to this application scope only



The following diagram illustrates the effect of denying other application scopes access to application tables from script API and web service calls.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Deny all runtime access permissions to application tables



Example granting all runtime access to a table

You can permit some or all runtime script API and web service calls from other application scopes.

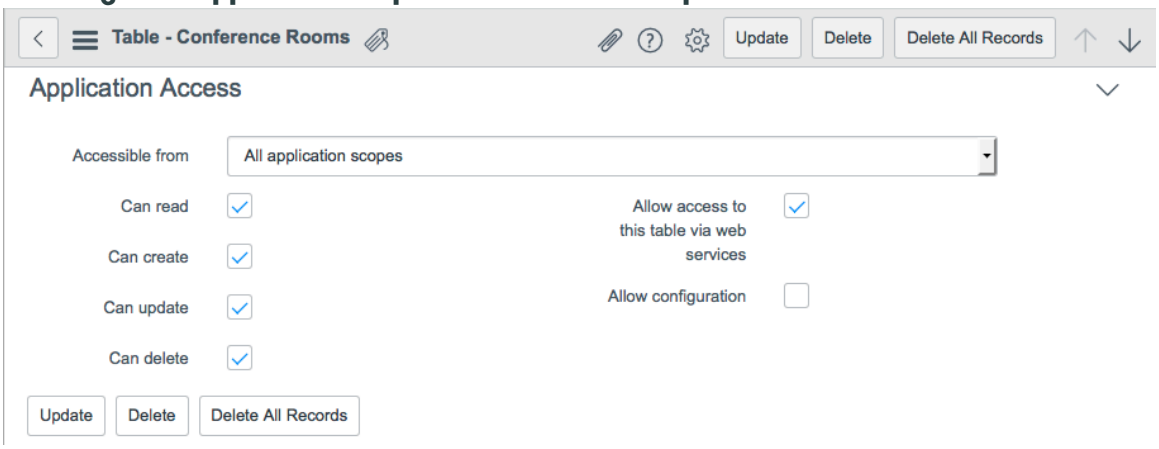
Granting access requires setting the following values in the table record.

Granting all runtime access

Field	Value
Accessible from	All application scopes
Can read	Enabled
Can create	Enabled
Can update	Enabled
Can delete	Enabled
Allow access to this table via web services	Enabled

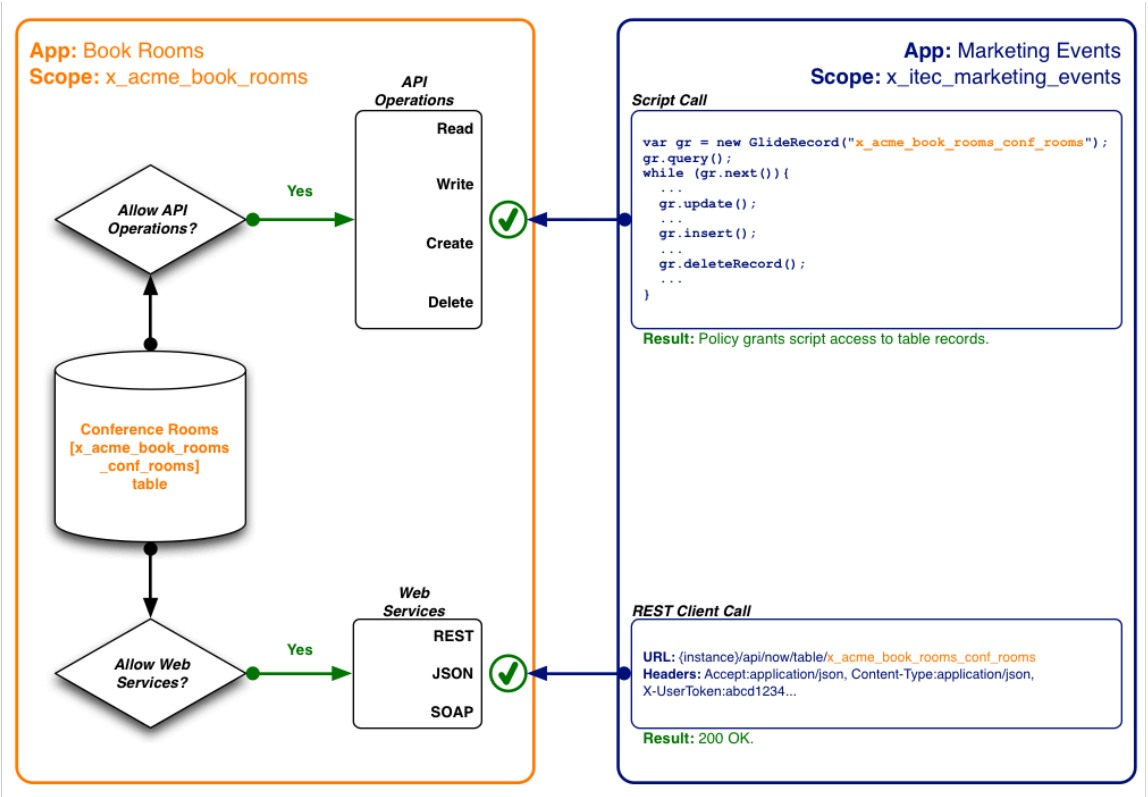
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Granting other application scopes all runtime access permissions



The following diagram illustrates the effect of granting access to application tables from API calls and web services in other application scopes.

Granted access to application tables



Design-time access to application tables

As the application developer, you can grant or deny other applications the permission to create configuration records, also known as application files, that extend the functionality of an application.

The permission applies to any platform feature that extends the functionality of an application data table such as:

- Business rules
- UI actions
- Client scripts

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

These access permissions protect the application data table at design-time. The system prevents you from creating configuration records by hiding the application data table as an option in the **Table** field. For example, a protected application table does not appear as an option when you create configuration records such as UI actions and client scripts.

Even when permission is granted to create configuration records, some configuration records have additional restrictions to protect application data from unwanted changes from other application scopes.

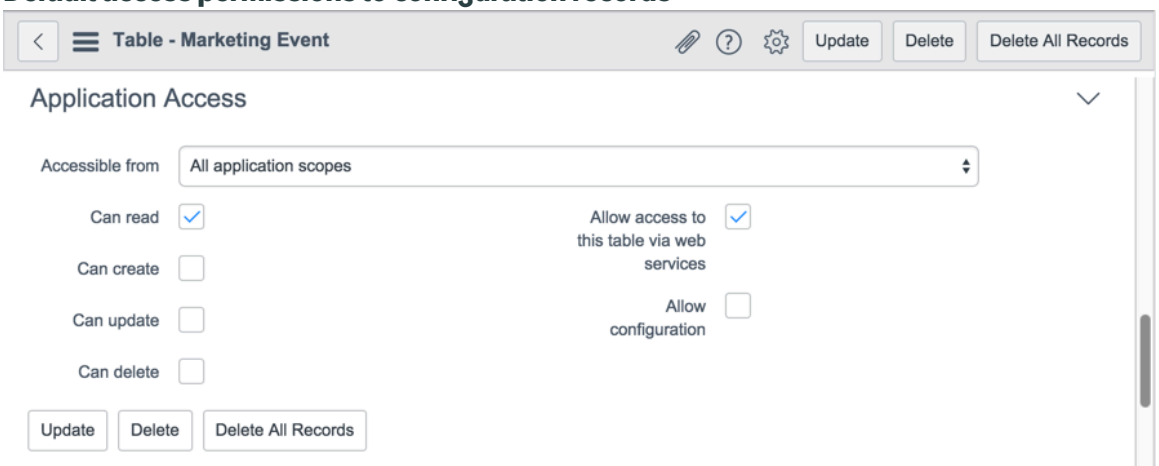
Default design access permissions

By default, new application tables prevent other application scopes from creating configuration records on application data tables. This prevents any other applications from changing the functionality of a table.

Default design-time access

Field	Value
Accessible from	All application scopes
Can read	Enabled
Can create	Disabled
Can update	Disabled
Can delete	Disabled
Allow access to this table via web services	Enabled
Allow configuration	Disabled

Default access permissions to configuration records



Set design-time access to application tables

Set these access permissions to protect application tables at design-time.

Before you begin

Role required: admin

About this task

To set runtime access permissions:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **System Applications** > **My Company Applications**.
2. Click the button for the application type you want to edit.

Option	Description
Developed	displays applications created on this in stance.
Downloaded	displays applications downloaded on this in stance.

3. Click the application name or the **Edit** button for the application you want to work on.
4. From the Tables related list, select the table whose access permission you want to set.
5. From the Application Access section, set the design-time access permissions.
6. Click **Update**.

Related topics

Example denying all design access to a table

Example allowing configuration records for a table

Example denying all design access to a table

You can prevent other application scopes from creating configuration records on application data tables.

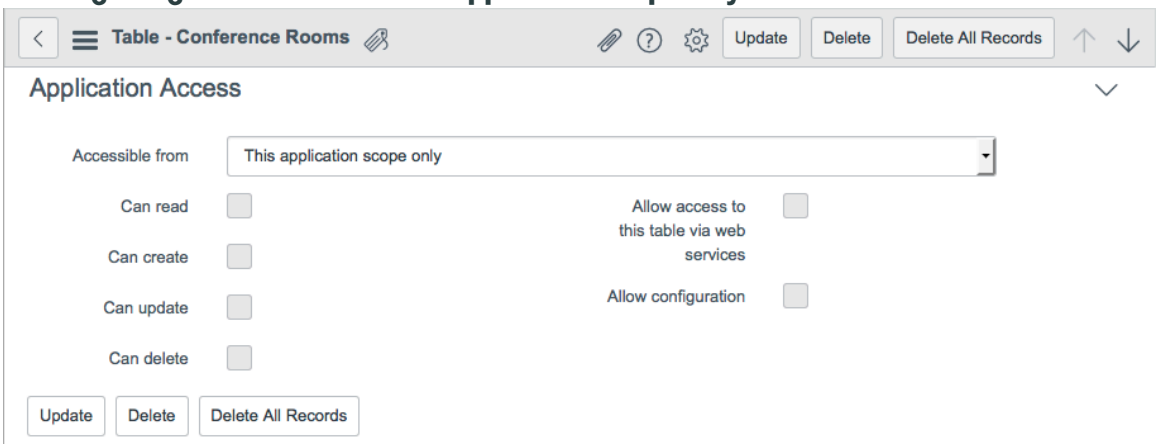
Typically, this is to prevent any other applications from changing the functionality of a table. Denying access requires setting the following value in the table record.

Denying all design-time access

Field	Value
Accessible from	This application scope only
Can read	Disabled
Can create	Disabled
Can update	Disabled
Can delete	Disabled
Allow access to this table via web services	Disabled
Allow configuration	Disabled

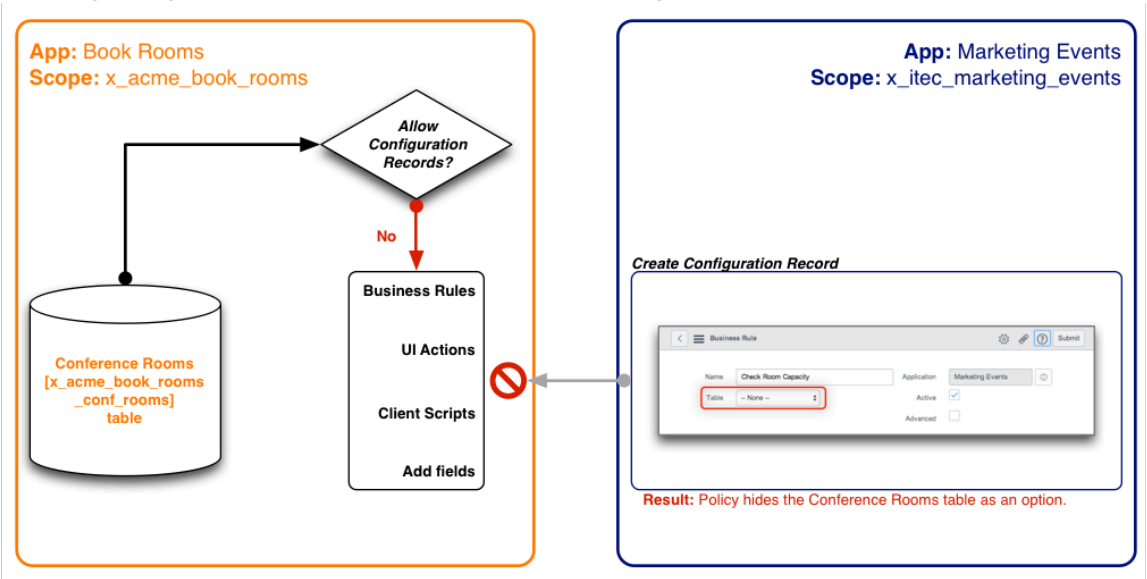
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Limiting design-time access to this application scope only



The following diagram illustrates the effect of denying other application scopes the ability to create configuration records.

Limiting design access to this application scope only



Example allowing configuration records for a table

You can permit other application scopes to create configuration records on application data tables.

You can grant access to the following configuration records with these settings.

Granting access to configuration records

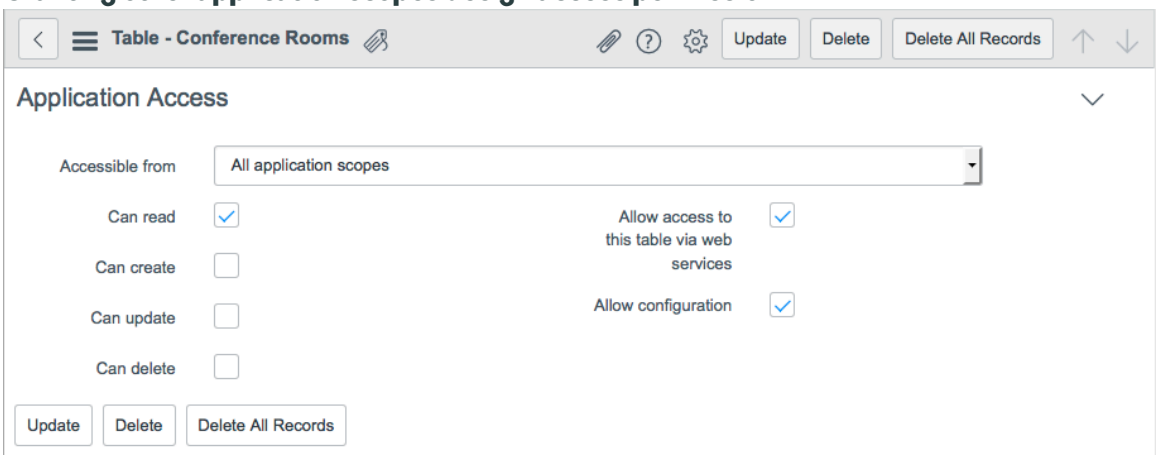
Configuration record	Setting required to grant access
Access controls	• **Accessible from** set to **All application scopes** • **Can read** is selected
Business rules	

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Granting access to configuration records (continued)

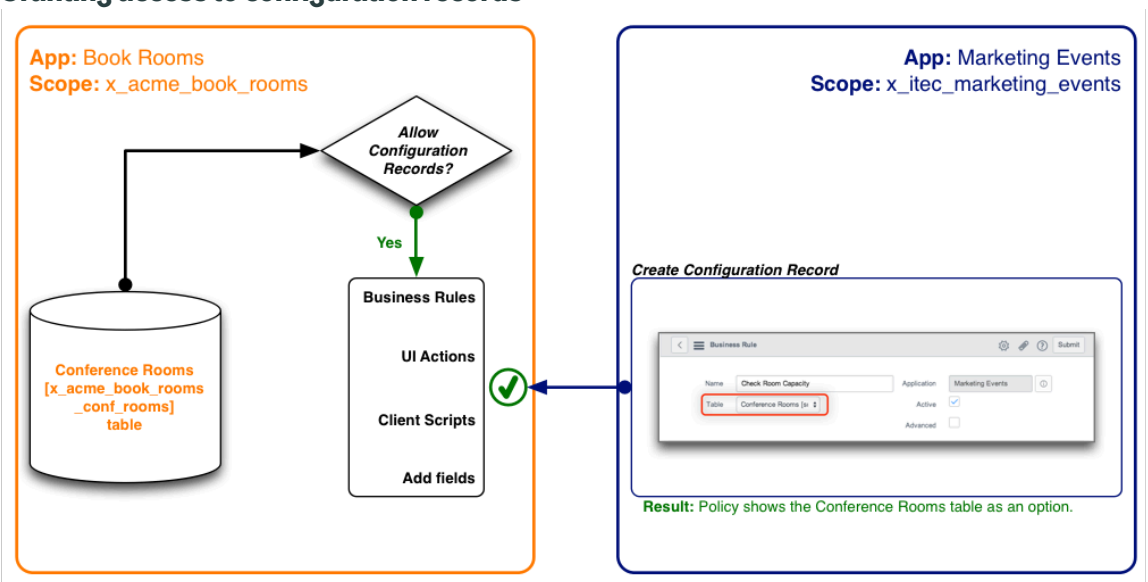
Configuration record	Setting required to grant access
Client scripts	• **Accessible from** set to **All application scopes** • **Can read** is selected • **Allow configuration** is selected
Dictionary entry (new field only)	
UI actions	

Granting other application scopes design access permission



The following diagram illustrates the effect of granting other application scopes the ability to create configuration records.

Granting access to configuration records



Restricted caller access privilege settings

Define cross-scope access to an application, application resource (such as an access control role, a business rule, a UI action, or a script include), or event. You can even use these settings to allow or deny requests for access.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Restricted caller access privilege settings overview

Restricted caller access [sys_restricted_caller_access] records track cross-scope applications or scripts that request access to an application, application resource, or event in the ServiceNow AI Platform. The ServiceNow AI Platform creates sys_restricted_caller_access records when one of these actions occurs:

- Caller access is set to **Caller Restriction** or **Caller Tracking**.
- A cross-scope script attempts to access an application resource or event.

Note: A system scope to target scope is an example of a cross-scope.

You can use these records to do these tasks:

- Track cross-scope requests for access to an application resource. You can use access requests to determine which applications need access to resources and data from other application scopes.
- Approve or deny any cross-scope requests for access to application resources or events. For example, you can create a Restricted Caller Access record to allow access for all scope-to-scope requests.

For more information, see Requested restricted caller access (RCA).

Restricted caller access privilege setting combinations

As a target application owner, you can define various combinations of privilege settings for restricted caller access and specify whether access is allowed or denied for each relationship. RCA records must be created in the target application scope to control access to your application's resources. You can define various combinations of privilege settings for restricted caller access and specify whether access is allowed or restricted for each relationship.

You can define various combinations of the following settings:

Scope

All application resources in a selected source or target scope. To learn more about application scopes, see Application scope.

Source

A specific application resource (such as a business rule, script include, or table) in a selected source scope.

Target

A specific application resource in a selected target scope.

These restricted caller access privilege settings combinations include, but are not limited to, the following combinations:

- Scope-to-scope: Control access from all resources in a source application to all resources in your target application
- Scope-to-target: Control access from all resources in a source application to a specific resource in your target application
- Source-to-scope: Control access from a specific source application resource to all resources in your target application
- Source-to-target: Control access from a specific source application resource to a specific resource in your target application

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For more information about these access setting combinations and to learn how to create each combination, see Set the application scope, application resource, and event access.

Note:

Source application developers who need to request access to resources in another application should coordinate with the target application owner. You can package Requested RCA records in your application, which will then be reviewed and approved or denied by the target application administrator upon installation.

Activating application restricted caller access

You can activate application restricted caller access through one of the following methods:

- Activate the Scoped Application Restricted Caller Access plugin (com.glide.scope.access.restricted_caller).
- Request the HR Service Delivery or Security Incident Response applications. By default, restricted caller access is active in these applications.
- Enable the Restricted Caller Access system property for Workflow Studio.

For more information, see: Activate application restricted caller access.

Activate application restricted caller access

You can activate the Scoped Application Restricted Caller Access plugin (com.glide.scope.access.restricted_caller) if you have the admin role.

Before you begin
Role required: admin

Procedure

1. Navigate to **All** > **System Applications** > **All Available Applications** > **All**.
2. Find the plugin using the filter criteria and search bar.

   You can search for the plugin by its name or ID. If you cannot find a plugin, you might have to request it from ServiceNow personnel.
3. Select **Install** to start the installation process.

   Note: When domain separation and delegated admin are enabled in an instance, the administrative user must be in the **global** domain. Otherwise, the following error appears: `Application installation is unavailable because another operation is running: Plugin Activation for <plugin name>.`

   You will see a message after installation is completed. For information about the components installed with a plugin, see Find components installed with an application.

Related topics

List of plugins (madrid)

Define cross-scope access to an application resource

Track cross-scope requests for access to an application resource and approve or deny requests.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

If you enable application administration for the target application, only application administrators of the target application can set access to an application. If application administration is not enabled, an admin user can set access to an application.

Role required: admin or application admin

Note: To learn about application-specific administrator roles and delegated development, see Access control rules in application administration apps and Delegated development and deployment.

Procedure

1. To define access to an application resource, navigate to the application resource record. Available application resources include these options.
   - Table
   - Script Include
2. Set the **Accessible from** field to **All application scopes**.
   If set to **This application scope only**, no other application scopes can access the resource.
3. Select the appropriate access level in the **Caller Access** field.

Option	Description
None	Cross-scope calls to the resource are approved or denied based on the value of the **Accessible from** field.
Caller Restriction	Calls to the resource must be manually approved. Access requests are tracked in the Restricted Caller Access table with a status of Requested.
Caller Tracking	Calls to the resource are automatically approved. Calls are tracked in the Restricted Caller Access table with a status of Allowed.

4. Allow or deny an access request from a calling application.
   When a cross-scope application attempts to access a resource set to Caller Restriction, the system denies access to the resource and creates a record in the Restricted Caller Access table with a status of Requested. An admin user or application administrator must allow or deny the request. When access is allowed, all future access attempts gain access to the restricted resource.

   If a calling resource changes (such as when a business rule's script changes), the restricted caller access record status changes to Invalidated. An admin user or application administrator must update the status to Allowed or Denied.

   a. In the application record, navigate to the **Restricted Caller Access Privileges** tab.

   b. Review records that have a Status of Requested.
   Each Restricted Caller Access Privilege record lists the operation performed, information about the calling source, and information about the target resource requested.

   c. In the **Status** column, set the value from **Requested** to **Allowed** or **Denied**.

   Once a calling source is allowed, all subsequent calls are allowed.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Set the application scope, application resource, and event access

Create a record in the Restricted Caller Access Privileges [sys_restricted_caller_access] table to set cross-scope resource access requests. Approve or deny requests from a source scope or source scope application resources to a target scope or to target scope application resources.

Before you begin

If you enable application administration for the target application, only application administrators of the target application can set access to an application. If application administration is not enabled, an admin user can set access to an application.

Role required: application admin or admin

Note: To learn about application-specific administrator roles and delegated development, see Access control rules in application administration apps and Delegated development and deployment.

About this task

You can set the following restricted caller access privilege settings combinations:

- Scope-to-Scope
- Scope-to-Target
- Source-to-Scope
- Source-to-Target

Note: An RCA privilege record must be present in the target application to grant access to a resource. This means that the target scope must match the application scope.

Procedure

1. Navigate to **All** > **System Applications** > **Application Restricted Caller Access**.
2. On the form, fill in the fields.

Restricted Caller Access fields

Field	Description
Operation	Operation performed on the target resource. ○ Read ○ Write ○ Create ○ Delete ○ Execute API
Source	Cross-scope record that is accessing a restricted application resource.
Source Scope	Scope of the calling application.
Source Table	Table that contains the Source record.
Source Type	Type of record that is calling the application resource: ○ ACL ○ Business Rule ○ Document Title

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

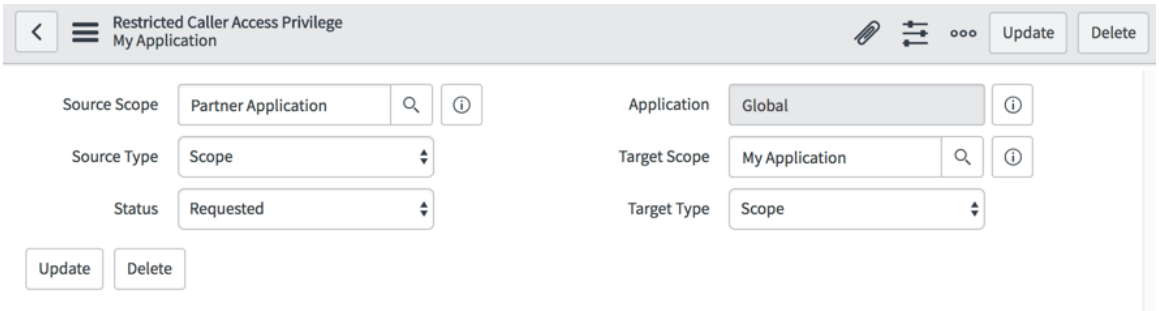
Field	Description
	○ Flow ○ Flow Action ○ GlideScopedEvaluator ○ Inbound Email Script ○ Orchestration RunScript Activity ○ Record Producer Script ○ Service Portal Widget ○ Scheduled Script ○ Scope ○ Script Include ○ UI Action ○ UI Macro ○ UI Page ○ Workflow Activity For example, to allow access from an entire application, select **Scope**.
Status	Status of the access request: ○ Requested ○ Denied ○ Allowed ○ Invalidated **Note:** If a calling resource changes, the restricted caller access record status changes to Invalidated. If you enable application administration, only application administrators of the target application can update the status of a request.
Target	Record of the requested resource.
Target Scope	Scope of the requested resource.
Target Table	Table that contains the Target record.
Target Type	Type of requested resource. ○ Event **Note:** An event is a special type of target for restricted caller access. By selecting an event in a target scope, you give a source application permission to queue an event that is registered as part of a target application. However, if you set the caller access on the event registry to None, it prevents cross-scope access calls to an event. This setting combination is a one-to-one relationship. To learn more about events, and their function, see Events. If you set caller access to None on the event registry, the cross-scope access calls to an event are denied.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
	○ Scope ○ Table ○ Script Include For example, to allow access to an entire application, select **Scope**.

Scope-to-scope settings

Allow or deny access of all application resources in a source scope to all application resources in a target scope. This setting combination is a many-to-many relationship.



Enter the following field settings for Scope-to-Scope restricted caller access.

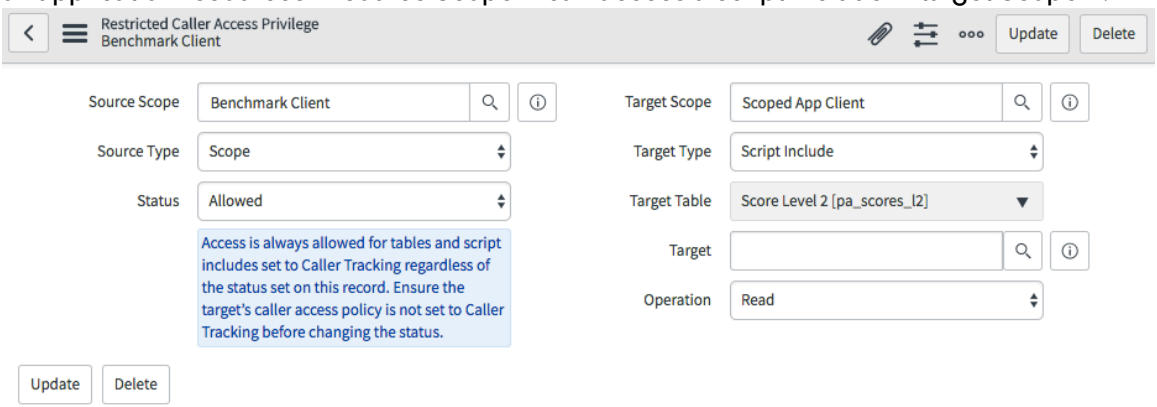
Field	Entries
Source Scope	Scope of the calling application that contains the source application resources.
Source Type	Type of record calling the application resource. To allow access from an entire application, select **Scope**.
Status	Status of the access request. Select **Allowed** to allow access, or select **Denied** to restrict access for this source-target resource relationship.
Target Scope	Scope of the requested resource that contains the target application resources that the source application resource requests access to.
Target Type	Type of requested resource. Select **Scope** to include all application resources in the target scope.

Scope-to-target settings

Allow or deny access of all application resources in a source scope to a specific application resource (business rule, table, script include, or event) in a target scope.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

This setting combination is a many-to-one relationship. For example, you can specify that all application resources in source Scope A can access a script include in target Scope B.



Enter the following field settings for Scope-to-Target restricted caller access.

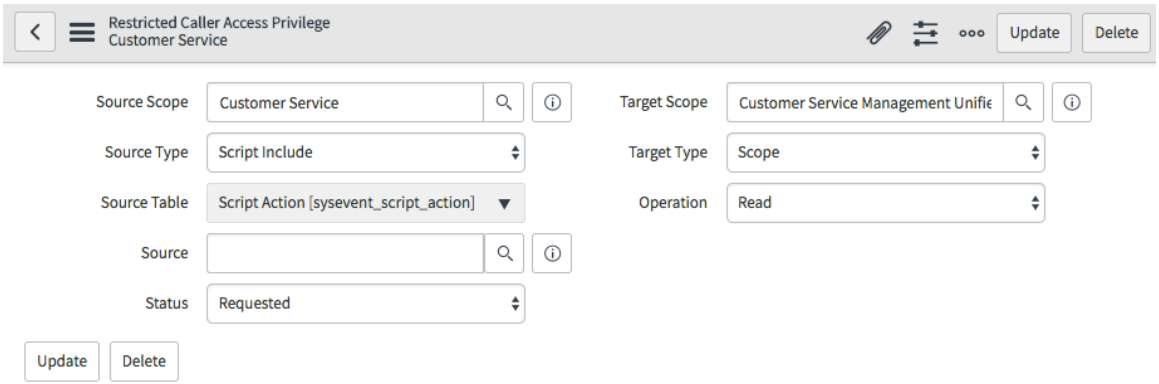
Field	Entries
Source Scope	Scope of the calling application that contains the source application resources.
Source Type	Type of record calling the application resource. Select **Scope** to include all application resources in the source scope.
Status	Status of the access request. Select **Allowed** to allow access, or select **Denied** to restrict access for this source-target resource relationship.
Target Scope	Scope of the requested resource that contains the target application resources that the source application resource requests access to.
Target Type	Type of requested resource. Select the specific application resource (for example, business rule, script include, UI page, event) the source application resource requests access to.
Operation	Type of operation (for example, Read, Write) in the target application resource the source application resource requests access to.

Source-to-scope settings

Allow or deny access of a specific application resource in a source scope to all application resources in a target scope.

This setting combination is a one-to-many relationship. For example, you can specify that a particular business rule in source Scope A can access all application resources in target Scope B.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Enter the following field settings for Source-to-Scope restricted caller access.

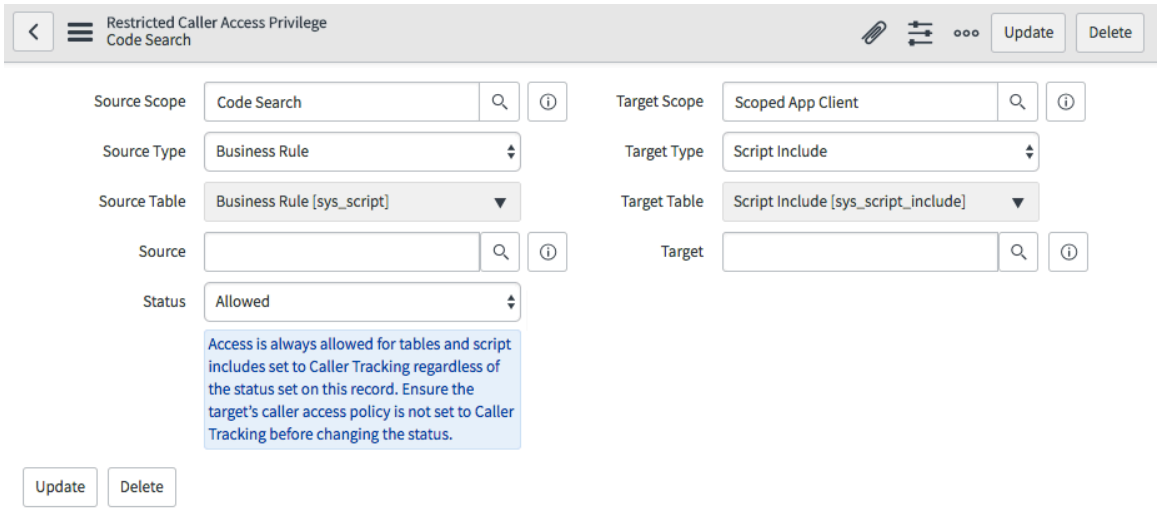
Field	Entries
Source Scope	Scope of the calling application that contains the source application resource requesting access to the target scope application resource.
Source Type	Type of record calling the application resource. Select the specific application resource (for example, business rule or script include) requesting access to the specified target scope application resource.
Status	Status of the access request. Select **Allowed** to allow access, or select **Denied** to restrict access for this source-target resource relationship.
Target Scope	Scope of the requested resource that contains the target application resources that the source application resource requests access to.
Target Type	Type of requested resource. Select **Target** to include all application resources in the selected target scope.
Operation	Type of operation (for example, Read, Write) in the target application resource the source application resource requests access to.

Source-to-target settings

Allow or deny access of a specific application resource in a source scope to a specific application resource in a target scope.

This setting combination is a one-to-one relationship. For example, you can specify that a specific business rule in source Scope A can access a specific application resource, such as a business rule, table, script include or event, in a target scope.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Enter the following field settings for Source-to-Target restricted caller access.

Field	Entries
Source Scope	Scope of the calling application that contains the source application resources.
Source Type	Type of record calling the application resource. Select the specific application resource (for example, business rule, script include, or UI page) requesting access to the specified target scope application resource.
Status	Status of the access request. Select **Allowed** to allow access, or select **Denied** to restrict access for this source-target resource relationship.
Target Scope	Scope of the requested resource that contains the target application resources that the source application resource requests access to.
Target Type	Type of requested resource. Select the specific application resource (for example, business rule, Script Include, event) the source application resource requests access to.

Requested restricted caller access (RCA)

You can use a requested RCA to grant store apps access to protected resources in the ServiceNow AI Platform without the need to wait for the next family release. If you have the system admin or application admin role, you can review requested RCAs and approve and deny them.

RCAs are classified into two categories:

- Real RCA: sys_scope==target_scope
- Requested RCA: sys_scope!=target_scope

For example: A real RCA record is where the application scope and target scope match. A requested RCA is a record that is still awaiting approval for access to the target scope.

When you install an application, your scheduled jobs generate RCA records with the status of Requested in the target application for each requested RCA record that is packaged in the source application.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: The jobs are generated once Upgrade Summary has run.

Example of how a store app accesses a table

Let's say that a store app called HR Integrations Framework wants to access an HR Core Case table. The table is in the business rule called Find Case in the Integration Service table.

To request access, the HR Integrations Framework app requires that an RCA privilege is packaged in its own scope as follows:

- sys_scope = HR Integrations Framework
- target = HR Core Case
- status = Allowed
- target_scope = Human Resource: Core
- source = Find Case

App development example for developers

When you are developing an application, real RCAs are generated with the status of Requested when the target has a caller restriction. If the target has caller tracking, the status becomes Allowed. The developer can review and finalize all the real RCA records that are required for the application to work. For example, those RCAs with a status of Allowed.

A developer can click the **Generate RCA Privileges in Current App** in the related links to generate requested RCAs that are packaged in the current application. Requested RCAs are synchronized with real RCAs, which means that if a real RCA is updated or deleted, a requested RCA is updated or deleted too.

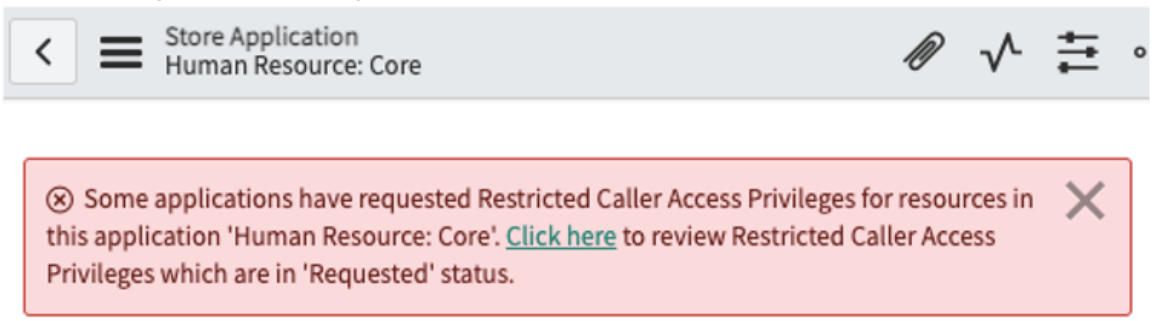
Now, the HR Integration Framework application can be packaged and installed on a customer instance.

App installation example for administrators

When you are installing an app on a customer's instance, real RCAs are generated in the target application. A real RCA would have the Human Resource: Core with a status of Requested. This process is done asynchronously in a scheduled job, where some lag time can occur.

To notify the target app admin about an RCA's pending review, messages have been added to application pages. An example is as follows:

RCA pending review message



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Store App backward compatibility

If a store app is compatible and can be installed on an instance that is pre-Rome, then you must package the RCA records in their own scope with the status of Allowed.

Note: This process ensures that the store app works on all versions.

When upgrading to Rome, you can configure a one-time fix script to move RCAs in the source scope to the target scope. In Rome, if the target app already has the necessary RCA records, no RCA records are generated for the RCAs that are packaged by the source app.

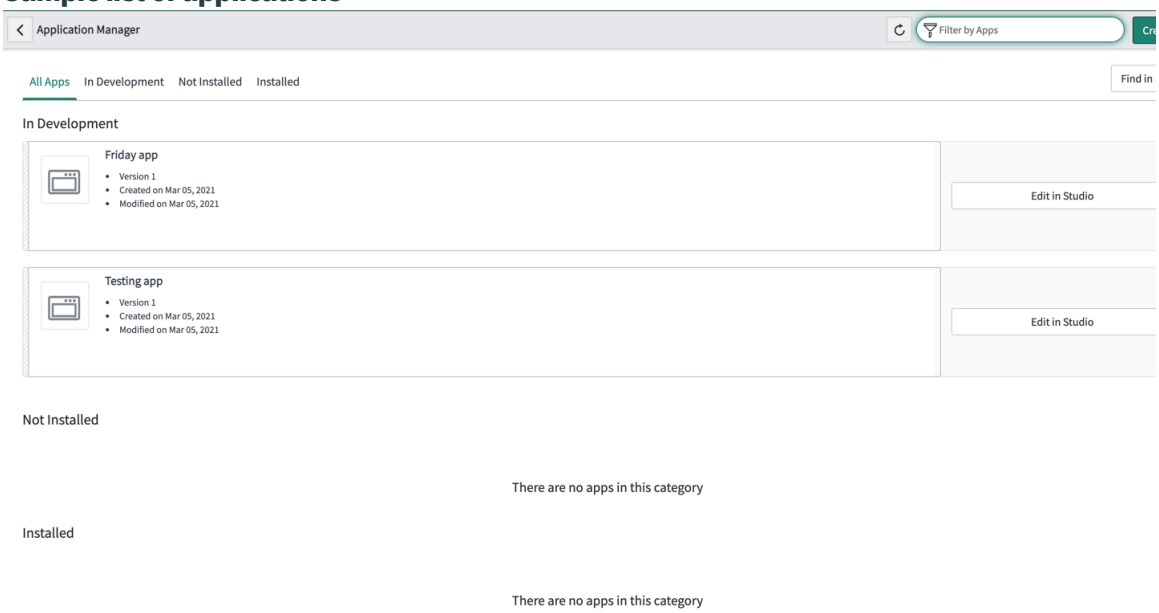
Application list

The applications list allows application developers to view and select applications.

Application developers can use the applications list to open a custom application record. If the contextual development environment detects that you are editing an application artifact in another application scope, it displays a warning message you can use to switch to another application.

Administrators have the following options from the applications list.

Sample list of applications



Application list options

Tab name	Options available	Application source
Develop	• Create new application • Edit existing application • Share application • Delete application	Applications developed on this instance

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Application list options (continued)

Tab name	Options available	Application source
Downloads	• Install • View installed files • Edit installed files • Uninstall	• Applications shared on the company application repository • Applications shared on the ServiceNow Store
Updates	• Update • View version information	• Applications shared on the company application repository • Applications shared on the ServiceNow Store

The application list displays the following information.

Applications list

Type	Description
Developed	Display applications created on this instance.
Downloaded	Display applications downloaded on this instance.
Updates	Display available updates for downloaded applications on this instance.

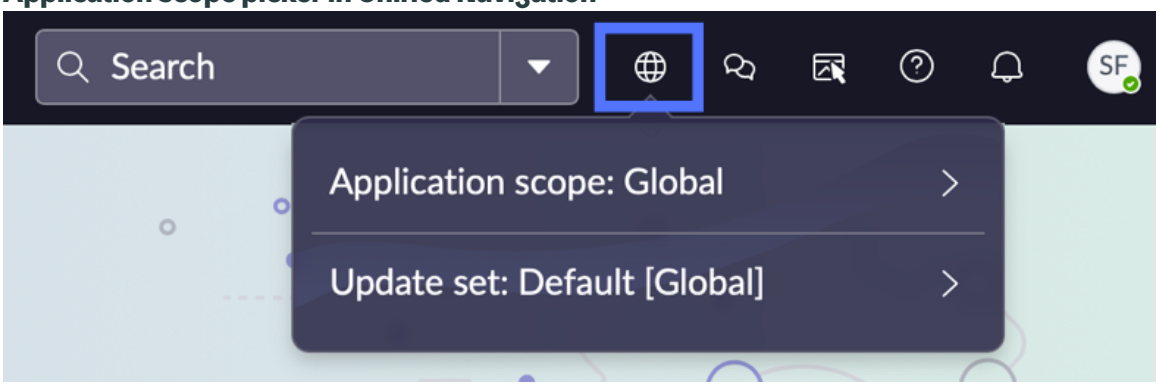
Application picker

The application picker enables application developers to view and select the application scope where their changes apply.

The application picker is available in the Unified Navigation menu. Select the picker icon ( ), then select **Application scope**, to open a list of application scopes that you have access to.

When your application scope is set to a non-Global scope, the picker icon displays with a red ring( ).

Application scope picker in Unified Navigation



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Resources

The application picker is part of the Next Experience picker. For more information, see Exploring Next Experience pickers.

For more information about application scoping, see Application scope.

Lists and forms in scoped applications

The current application context determines what customization and form design options are available when working with lists and forms in scoped applications.

The user interface uses visual indicators to identify a list or form in the same or different application scope.

Available layout and design actions

The system allows the following layout and design actions when working on lists or forms in custom applications.

Available layout and design actions

Action	Access granted	Notes
Create a list view	Always allowed	This action is always available to users with access to customization.
Create a form view	Always allowed	This action is always available to users with access to customization.
Create a form section	Always allowed	This action is always available to users with access to customization.
Select fields to display in a view	Only allowed for sections in the view that match the current scope.	This restriction is independent of a user's role. Administrators cannot bypass this restriction.
Change the order of sections in a view	Only allowed for views that match the current scope.	This restriction is independent of a user's role. Administrators cannot bypass this restriction.
Select fields to display in a section	Only allowed for sections that match the current scope.	This restriction is independent of a user's role. Administrators cannot bypass this restriction.
Add or remove section columns	Only allowed for sections that match the current scope.	This restriction is independent of a user's role. Administrators cannot bypass this restriction.
Delete a form section	Only allowed for sections that match the current scope.	This restriction is independent of a user's role. Administrators cannot bypass this restriction.
Create new fields	Only allowed for sections that match the current scope and when the **Allow configuration** option is enabled.	This restriction is independent of a user's role. Administrators cannot bypass this restriction.

Form design visual indicators

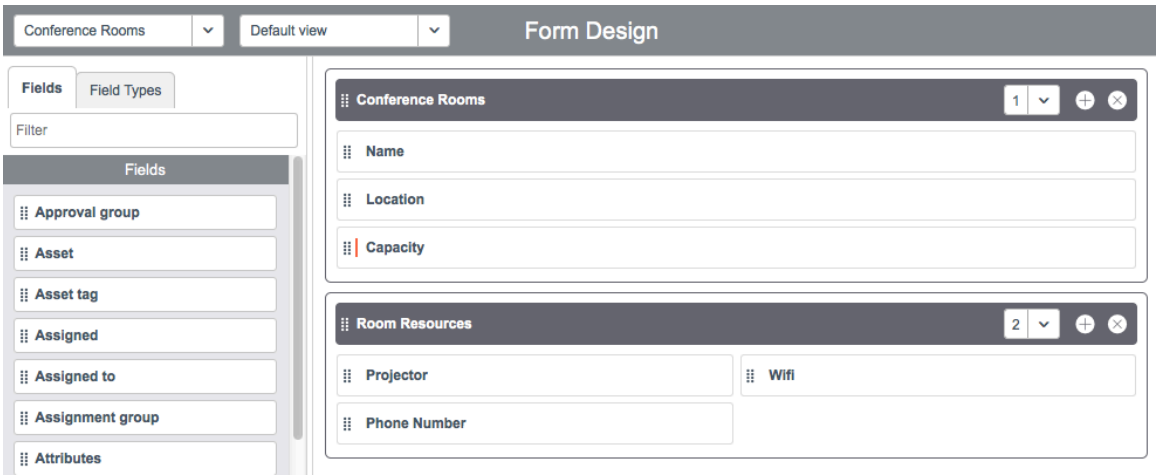
The UI displays the following visual indicators when designing forms in custom applications.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

You can only edit views and sections when you are in the same application scope as the form. Editable sections display:

- Section headings with a solid color background.
- A solid line around the section.
- A control to set the number of columns.
- A **Delete this section** button.
- Grip icons beside section headings.
- Grip icons beside fields.

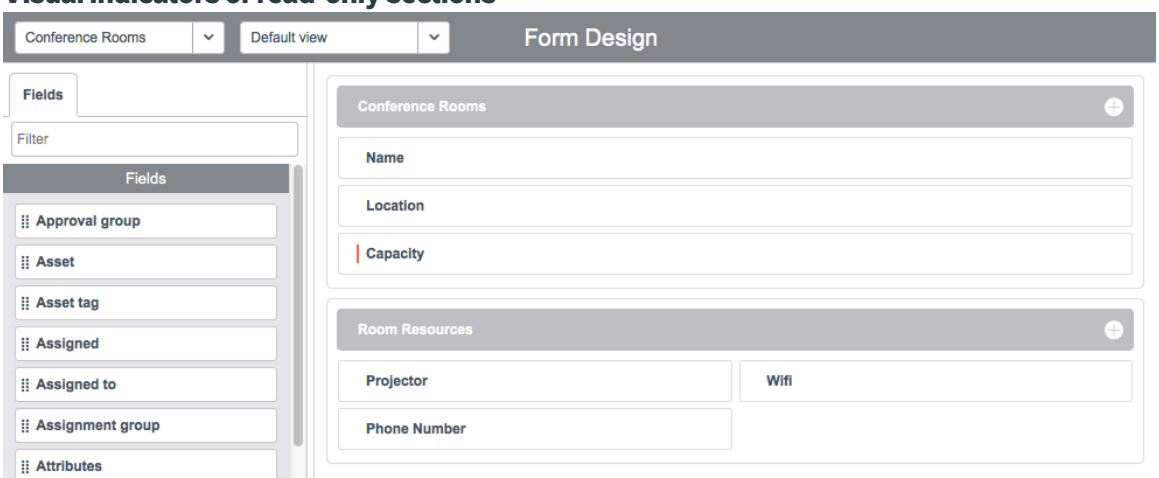
Visual indicators of editable sections



Views and sections in another application scope display as read only. Read-only sections have:

- Section headings with a gray background.
- A gray line around the section.
- No control to set the number of columns.
- No **Delete this section** button.
- No grip icons beside section headings.
- No grip icons beside fields.

Visual indicators of read-only sections



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Default form design permissions

By default, new application data tables have the following form design permissions.

Default form design permissions

Form design action	Permission setting
Create new sections in tables belonging to another application scope	Allowed
Create new fields in sections belonging to the same application scope	By default, denied. Requires application designer to set **Allow configuration** for application table.
Add or remove fields from sections belonging to the same application scope	Allowed
Change the order of fields in sections belonging to the same application scope	Allowed
Change the order of sections belonging to the same application scope	Allowed
Add or remove fields from sections belonging to another application scope	Denied
Change the order of fields in sections belonging to another application scope	Denied
Change the order of sections belonging to another application scope	Denied
Create new fields in sections belonging to another application scope	Denied

Layout visual indicators

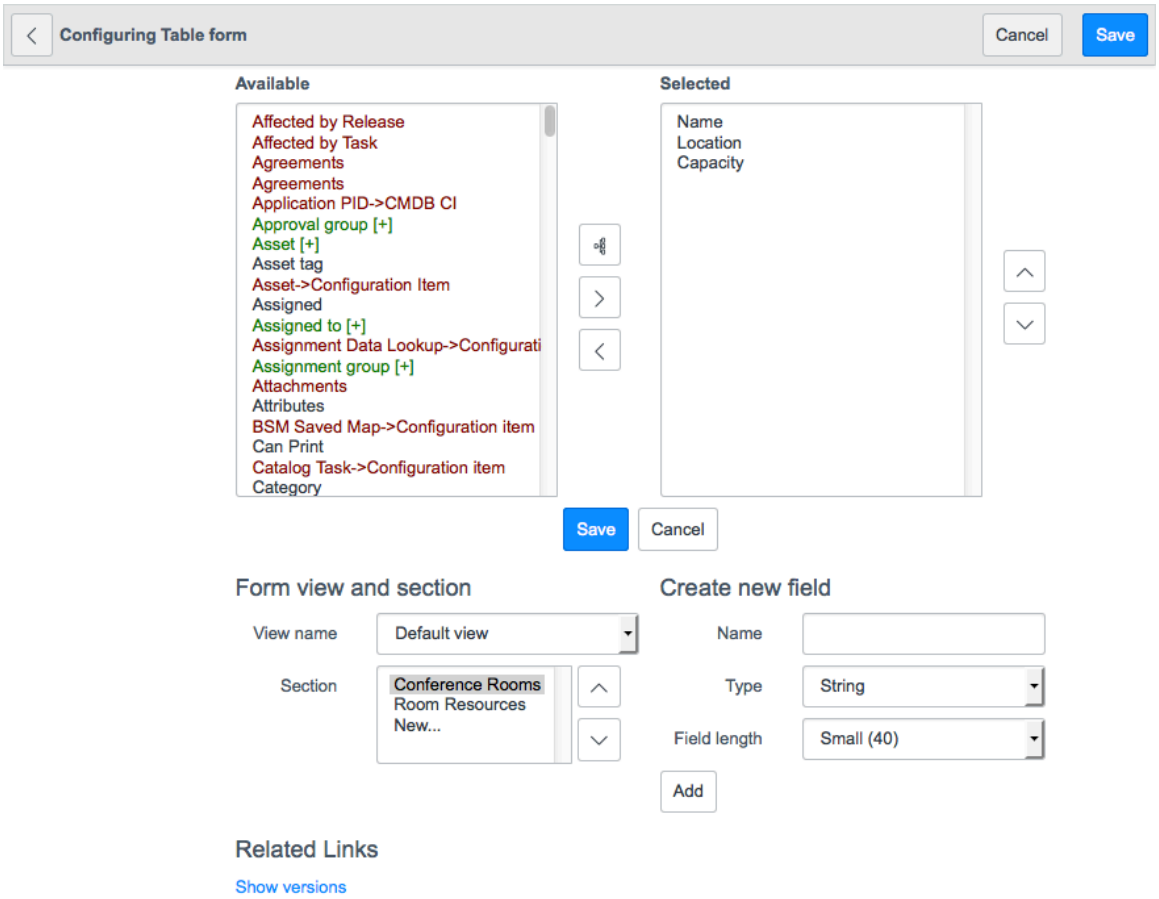
The UI displays the following visual indicators when configuring the layout of a custom application's list or form.

You can only edit fields, views, and sections when you are in the same application scope as the form. Editable sections display:

- Field selections with a white background.
- Buttons to add or remove fields.
- Save button with a blue background.
- Editable fields in the **Create new field** section.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Visual indicators of editable sections

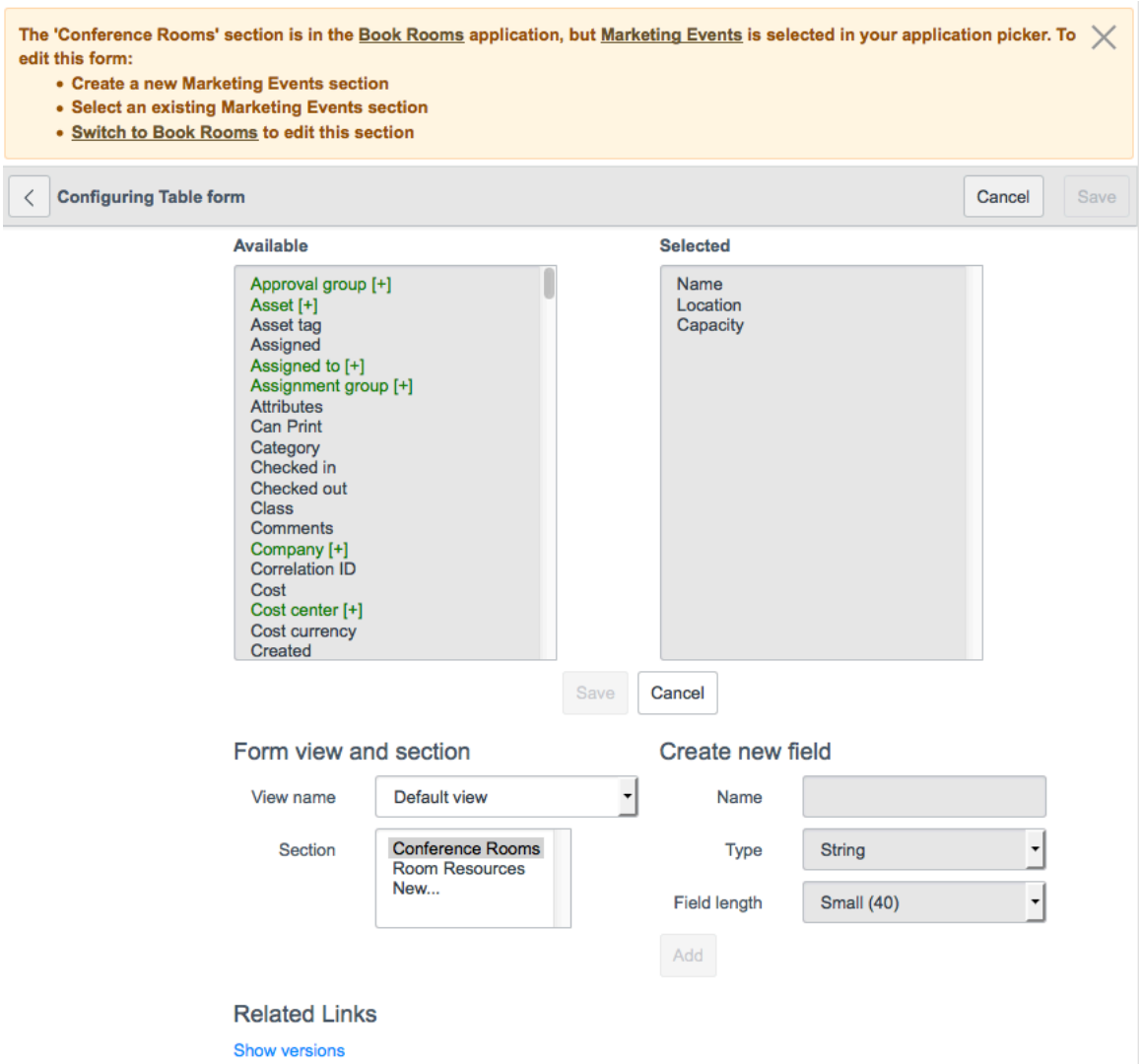


Views and sections in another application scope display as read only. Read-only sections have:

- A warning message about the currently selected application scope and the scope of the form.
- Field selections with a gray background.
- No buttons to add or remove fields.
- Save button with a gray background.
- Read only fields in the **Create new field** section.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Visual indicators of read-only sections



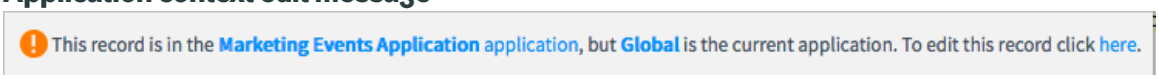
Related topics

- Lists and forms in scoped applications
- Available layout and design actions
- Form design visual indicators

Contextual development edit messages

The platform displays a message if you attempt to edit a Store Application record when you're in a different application scope.

Application context edit message



This message can be used to:

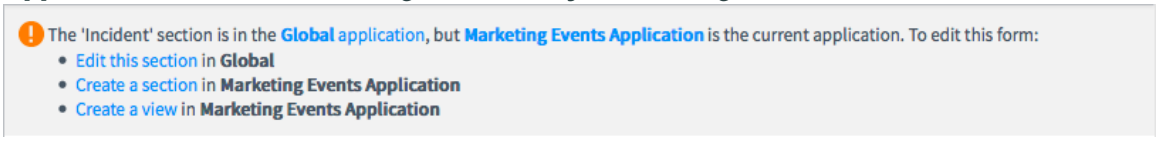
- Open the application to which the current configuration record belongs.
- Open the application of the currently selected application in the application picker.
- Temporarily switch to the application to which the current configuration record belongs and edit it.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: The system returns you to the current application context after you save or cancel out of the record.

The system also displays a message when a user attempts to configure a list or form layout while working from another application scope.

Application context edit message for form layout or design



The message provides a list of valid options:

- Edit the current section by temporarily switching to the application that owns it.
- Create a new section in the current application context.
- Create a new view in the current application context.

Note: The system returns you to the current application context after you save or cancel out of the record.

Script protection policy

Application developers can set a protection policy for script includes published as part of a custom application. The policy determines whether someone can view or edit the script include after the application is installed on their instance.

Application developers have these options to protect their custom application script includes:

Script include protection policy Options

Protection policy	Description
None	Select this option to allow other application developers to customize your script include.
Read-only	Select this option to allow other application developers to see your script logic, but not to change it.
Protected	Select this option to prevent other application developers from changing your intellectual property.

Application administration

Protect sensitive application data by using application administration to restrict how users acquire application-specific roles.

Functions of application administration

Use application administration to:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Help prevent unauthorized users from accessing sensitive data via forms, lists and UI, such as financial records or personally identifiable information.
- Determine who can assign scope-protected roles such as the administrator and designated developers for the application.
- Help prevent users with the system-level admin role from:
  - Assigning themselves a protected application role.
  - Assigning themselves to a group containing a protected application role.
  - Bypassing existing access controls to a protected application by creating access controls.
  - Impersonating a user who has a protected application administration role, unless the developer or administrator also has that role.
  - Inheriting a protected application role.
  - Overriding existing access controls to a protected application.

Roles in application administration

You can make any role an application-specific administrator by selecting the **Application Administrator** check box in Role Configuration. To learn more, see Restrict access to an application. By convention, create the following roles:

Application administration roles

Role name	Description
Application-specific admin	Users with this role can assign other users to an application-specific role for that application. For example, you can create a role named my_application.admin. It should include the name of the restricted application, with a suffix of "admin" to indicate that it is the admin role for the application.
Application-specific developer	Users with this role can access the restricted application. For example, you can create a role named my_application.developer. It should include the name of the restricted application, with a suffix of "developer" to indicate that it is the developer role for the application. The developer role needs both application administrator and delegated development permissions to modify the application files. To learn more, see Delegated development and deployment and Delegate development and deployment permissions to personnel.

Application-specific admin role

The application-specific admin role enables a user to access a specific application, but does not grant the user any other admin rights. Assign the system-level admin role to a user before that user can do these tasks:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Configure form and list layouts.
- Change application tables and fields.
- Assign the application-specific admin role to new users.

If you do not want a user with the application-specific admin role to have the system-level admin role:

- Do not assign the system-level admin role to the user. Assign only the application-specific admin role.
- Have the user assign themselves the application-specific developer role.

As an application-specific developer, the user can perform a subset of administrative tasks without having the system-level admin role.

Note: Assign the application-specific admin role to more than one user. Then if a user with the application-specific admin role leaves the company, you are not prevented from changing the application.

Enabling application administration and assigning application-specific roles

You can enable application administration for an application from the application record and restrict the assignment of application-specific roles from the user role record.

Note: Enable application administration and assign application-specific roles after completing development of the application, but before adding application records. This practice protects sensitive data in the application records from access by unauthorized users.

The target instance must have at least one authorized user with the application-specific admin role.

- If you enable application administration for an application but do not assign the application-specific roles, no user can access the application.
- If you assign only one application-specific role, you cannot delete that role.

A warning appears if you enable application administration for an application, but no users have the application-specific admin role required to assign roles for the application. The warning reminds you to assign the application-specific roles for administrators and developers of the application.

Set the *[scoped_app_name].min_admin_count property* to require that more than one user must have the application-specific admin role. Assigning the application-specific admin role to multiple users reduces the risk of getting locked out of the scoped application. The *[scoped_app_name].min_admin_count* property has the following limitations:

- If you specify an invalid value for the property, the default requirement for assigning at least one application-specific admin is enforced.
- If you specify a valid value for the property, you can't delete any application-specific admins unless you exceed the specified value. For example, if you specify a value of two and you have three application-specific admins, you can delete only one of those roles.
- You can specify a value higher than the actual number of assigned application-specific admins. However, you can't delete any application-specific admins until you exceed the specified value. For example, if you specify a value of six, but have only three application-specific admins, you can't delete any of those roles.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For procedures, see Restrict access to an application.

Deploying applications with application administration

You must have the system-level admin role in both your developer and production instances to deploy an application protected by application administration. The process is outlined in the following steps.

1. Develop the application on a development instance.
2. Create the application-specific admin role.
3. Grant the application-specific admin role to all system-level admin users.
4. Update the application record to enable application administration and restrict access to the application.
5. Publish the application to the application repository.
6. From a production instance, install the application from the application repository.
7. As a system-level admin on the production instance, grant the application-specific admin role to the appropriate users.
8. Remove the application-specific admin role from all users with the system-level admin role.

For procedures to enable application administration and restrict the assignment of application-specific roles, see Restrict access to an application.

Training

The ServiceNow® Developer Site has training for Securing Applications.

Related topics

Delegated development and deployment

Restrict access to an application

Restrict management of an application and access to that application to prevent unauthorized users from assigning administrative rights to the application or accessing sensitive information in the application records.

Before you begin

- Records required:
  - Role record to designate a role as the application-specific admin role
  - User record to assign an application-specific admin role to a user
  - Application record to enable application administration for a specific application
- Role required: admin

If an application-specific admin role does not already exist, create it before beginning this procedure. For example, you can create a role named my_application.admin that includes the name of the restricted application with the suffix "admin" to indicate that it is the admin role for the application.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **System Security** > **Users and Groups** > **Roles** or **User Administration** > **Roles**.
2. Open the role record for the application-specific admin role.
3. Configure the form to add the **Application Administrator** field.

   Note: The **Application Administrator** check box replaces the **Assignable by** field. By default, when you upgrade from a Jakarta or earlier release to a Kingston or later release, any role that was in the **Assignable by** field is defined as the application-specific admin role, and the **Application Administrator** check box is selected.

4. In the role record, select the **Application Administrator** check box, and then click **Update**.
5. Navigate to **System Security** > **Users and Groups** > **Users**.
6. Open the user record for the admin user.
7. On the Roles tab, add the application-specific role.
   Only users with the application-specific admin role can enable application administration for an application.

   Note: Assign the application-specific admin role to more than one user. Then, if a user with the application-specific admin role leaves the company, you are not prevented from performing changes to the application.

8. Click **Update**.
9. Log out and then log in with the application-specific admin role.
10. Navigate to **System Applications** > **My Company Applications**.
11. Select the application for which you want to enable application administration.
12. In the application record, select **Application administration**.
13. Click **Update**.
    The system validates that the following requirements have been met:
    - The application has an application-specific admin role (there is at least one role with **Application Administrator** selected).
    - The current user has the application-specific admin role.

    If the validation passes, the system updates the application record. Otherwise, the system displays this error message and does not update the application record:

```
Application Administration uses the 'Application Administrator'
 role to define what users are application administrators. None
 of the roles defined by this application have 'Application
 Administrator' enabled.
```

14. **Optional:** From the Related Links, you can select one of the following options:

Related Link	Description
Manage Developers	Modal that enables the application-specific admin to manage these tasks: ○ Designate developers for the application. ○ Make themselves a delegated developer. After the application-specific admin becomes a delegated developer,

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Related Link	Description
	the application-specific admin can perform a subset of administration tasks without having the system-level admin role. Learn more: Delegated development and deployment
Grant application administration to all admins	Modal that creates a Contained Role [sys_user_role_contains] record for the system-level admin role. This adds the application-specific admin role as a contained role of the system-level admin role. **Note:** When you publish the application with this record, users with the application-specific admin role can access the application after installing it.

Access control rules in application administration apps

By default, when application administration is enabled for a scoped application, ACL rules for the scoped application are applied. If no ACL rules for the scoped application are found, global ACL rules can apply.

This behavior applies to configuration records created in tables that extend the Application File [sys_metadata] table only. You can also change the default behavior.

When no access control (ACL) rules for an application administration app are defined, global ACL rules can apply to the configuration records of the application administration apps. See Application files for more information.

Note: Not all applications utilize global ACL rules. If you are unable to perform an action as expected, verify that the application allows global ACL rules. If it does not, you may need to create specific ACL rules for some applications.

To allow a table in an application administration app to inherit global ACL rules, check that the system property is true and follow the instructions detailed in Configure a table in an application administration app to inherit global ACL rules.

- `glide.security.scoped_administration.honor_global_acl` system property: If no scoped ACL rules are defined, application administration apps can inherit global ACL rules. By default, this property is enabled for new and upgraded instances.
- Application Administration ACL Inheritances [sys_scoped_admin_acl_inheritance] table: If no ACL rules for the application administration app are found, tables added to this list inherit global ACL rules. Only the administrator for the application administration app can add, remove, read the records owned by the application administration app in this configuration table.

Configure a table in an application administration app to inherit global ACL rules

To avoid duplicating global access control rules (ACLs) in your applications, you can configure application file tables in application administration apps to inherit global ACLs when no ACL rules for the scoped application are found.

Before you begin

Role required: admin role for the application

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Enter `sys_scoped_admin_acl_inheritance.list` in the navigation filter.
   The Application Administration ACL Inheritance table [sys_scoped_admin_acl_inheritance] opens.
2. Click **New**.
3. In the **Table** field, select an application file table (a table that extends the Application File [sys_metadata] table) from the application administration app.
   Only the administrator for the application administration app can add/remove/read records owned by the application administration app in this configuration table.
4. Click **Submit**.
   If no ACL rules from the application administration app are found, tables added to this list inherit global ACL rules.

Access enforcement for ServiceNow Store apps

All production instances monitor and generate reports on usage patterns for ServiceNow Store apps. When subscription enforcement is enabled, users who are not subscribed to the app are blocked from performing fulfiller actions in the app.

Overview of ServiceNow Store access

The following actions are required to enable a production instance to enforce entitled usage of your ServiceNow Store App:

1. The usage admin at your organization uses the Subscription Management application to allocate fulfiller users to the subscription.
2. You decide on the enforcement mode, either:
   - Monitor and report usage with no enforcement (default)
   - In addition to monitoring and reporting usage, enforce that all usage must be by subscribed fulfiller users.
3. To enforce usage only by subscribed users, you configure the tables where only record owners or subscribed fulfiller users can make updates as fulfillment tables.

Licensing

Understand the different types of software licenses and subscriptions available with ServiceNow applications.

Licenses

Note: Discuss licensing with your ServiceNow account representative for information specific to your contract.

You need a license to get entitlement for an application. If you have entitlement to a particular app, you can activate it on the ServiceNow Store.

All software licenses outline exactly how the software may be used without infringing on copyright law. There are many different subcategories of software licenses. These categories are not necessarily exclusive; different kinds of software will often overlap.

For more information on different types of software licenses, see What is a software license?

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Subscriptions

A subscription enables you to upgrade from one family release to the next without dealing with the ServiceNow Store. For example, if you're on Tokyo and have a subscription to an app, you're automatically upgraded to Brazil as soon as it's in general release.

The type of subscription determines the allocation of users, access to applications, and custom application and table entitlements. For more information on types of subscriptions available for ServiceNow applications, see Types of subscriptions in Subscription Management.

Creation restrictions across application scopes

The system restricts the creation of some configuration records when the current application scope does not match the application scope of the configuration record's target table.

Configuration record creation restrictions prevent one application from making unwanted changes to another application's data tables. These restrictions only apply when you create a configuration record whose target table belongs to another application. Configuration records that belong to the same application scope do not have these restrictions.

The system always enforces the following creation restrictions when a developer adds a configuration record belonging to another application scope.

Configuration record creation restrictions

Configuration record type	Creation restrictions when target table is in another application scope
Access controls	• You can only create field-level access controls with a role-based requirement. • You cannot create table-level access controls for a table in another application scope. • You cannot create field-level access controls that apply to all fields. • You cannot create access controls that use conditions. • You cannot create access controls that use a script-based condition.
Business rules	• You can create a rule where **When** is *async* with any of the following options: ○ *Insert*, *Update*, and *Delete* database operations. You cannot select *Query*. ○ **Set field values** actions and scripts (the **Script** field). • You can create a rule where **When** is *before* with any of the following options: ○ *Insert*, *Update*, and *Delete* database operations. You cannot select *Query*. ○ **Set field values** actions only. You cannot write scripts and you cannot abort the database transaction.
Calculated fields	You cannot create calculated fields for tables in another application scope.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Configuration record creation restrictions (continued)

Configuration record type	Creation restrictions when target table is in another application scope
Data policies	• You cannot create data policy rules for fields in another application scope. • You cannot make a field mandatory.
Field styles	You cannot create field styles for fields in another application scope.
Form sections	• You cannot modify existing form sections created in another application scope. • You can create new form sections.
Record producers	You must have create access to the application table to create records from a record producer.
UI policies	• You cannot create UI policy rules for fields in another application scope. • You cannot make a field mandatory.
UI script	You cannot create a global UI script from a scoped application.
Views	• You can create new views. • You cannot modify existing views created in another application scope.

Planning your application

The application development process starts with planning. Consider how the application will work, who will use it, and how it improves user experience.

Planning on the Next Experience

App intake

With your carefully planned idea in hand, you're ready to submit it through Application Intake for approval and development in App Engine Studio (AES).

Define tables and fields for application records

Applications use tables and records to manage data and processes, such as Incident, Problem, and CMDB.

Security: App Engine Studio roles and permissions

Administrators assign roles to give team members permission to configure or use AES.

Planning on UI16

Delegated development

Delegated development enables designated users without a system admin role to develop or deploy applications on the ServiceNow AI Platform.

Schema map for tables

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The schema map displays the details of tables and their relationships in a visual way, enabling administrators to view and easily access different parts of the database schema.

Table Administration

Administrators can use Table Administration to view and modify the database structure.

Team Development

Team Development supports parallel development on multiple, non-production ServiceNow instances.

Related applications and features

Security: Access control

By default, when application administration is enabled for a scoped application, access control (ACL) rules for the scoped application are applied. If no ACL rules for the scoped application are found, global ACL rules can apply.

Security: Edge Encryption

ServiceNow® Edge Encryption encrypts sensitive data on your company premises before sending it over the internet to your ServiceNow instance (encrypted in flight), where it remains encrypted at rest.

Security: Roles

Set up the security and roles for your UI Builder instance. Security and roles in UI Builder are controlled through your application's scope, domain separation, and protection policy settings.

Security: UI policies

Create a UI policy to define custom process flows for tasks.

Security: Users and groups

Create users and groups using the ServiceNow AI Platform user administration feature.

ServiceNow application repository

After you develop and test a custom application, you can make the application available to company instances by publishing it to the ServiceNow application repository.

Submit your idea for app development

With your carefully-planned idea in hand, you are ready to submit it through Application Intake for approval and development in App Engine Studio (AES).

Before you begin

If you have not already activated the Apply for Citizen Development standard catalog item, you must do so before you can perform this procedure.

Role required: none

About this task

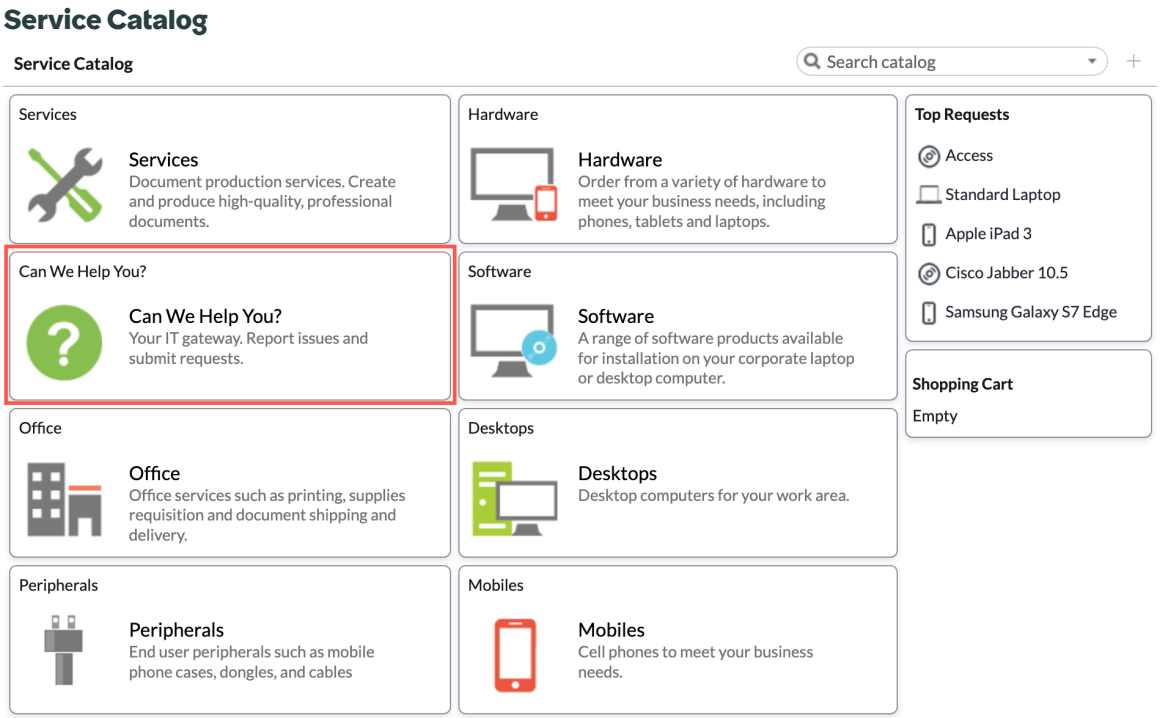
The intake request process streamlines submitting ideas for app development. Using Application Intake, the idea is shared with your App Engine Studio admin for review. The admin either approves or denies the request based on how well you have sold the business value of the idea.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

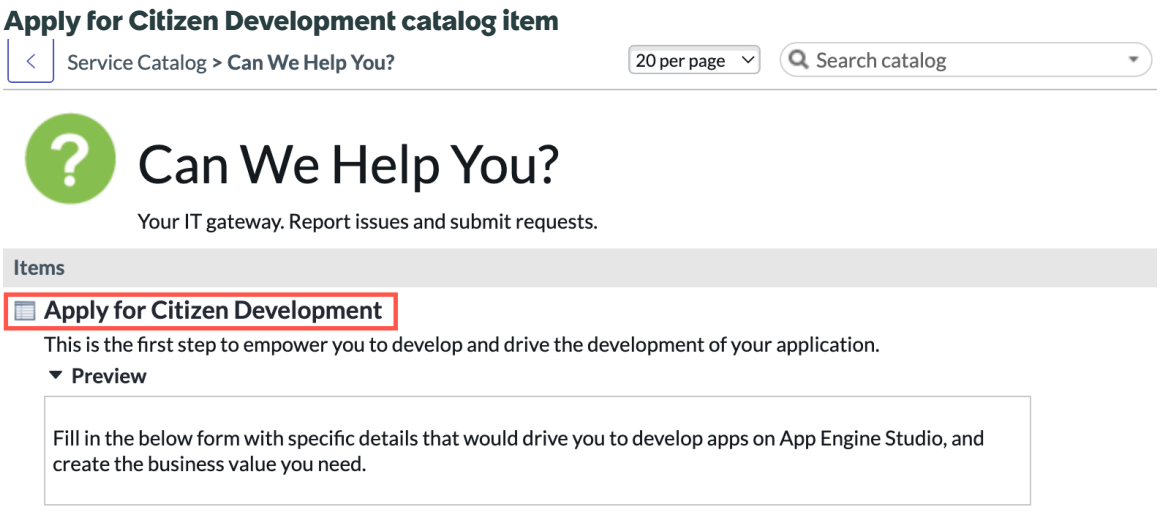
If the idea is approved, developers can build the app with the guided, intuitive app development environment provided in App Engine Studio.

Procedure

1. Navigate to **All** > **Self-Service** > **Service Catalog**.



2. In the Service Catalog, select the **Can We Help You?** category.



3. Select the **Apply for Citizen Development** catalog item to open the Application Intake form.
4. On the form, fill in the fields.

Fields on the Apply for Citizen Development form

Field	Description
Application Name	Unique name for the app.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Describe your idea in a few sentences	Simple description of the application.
Is your process repeatable?	Option to indicate whether the app idea could be used in other areas of the business.
Do you have an email or excel-based process?	Option to indicate whether an existing process is in place that this application would replace.
How many users are involved in this?	Number of users who would use the resulting application.
Does this involve any sensitive/PII data?	Option to indicate if the application would use any of the users' health, financial, or personally identifying information.
Do you need data from other departments?	Option to indicate whether other departments must supply data for the application.
Who are the users that will have access to the data in this application?	Users or groups who need access to the application.

5. Select **Order Now**.

Result

If the admin approves your request, you can start building your app in App Engine Studio. You should receive a confirmation email with access to build the app.

If you are an admin who approves requests, learn more about managing AEMC requests.

Specify data for your application

In its most basic form, building applications on the ServiceNow AI Platform means you're storing and using data in some way. Defining your application's data model is up to you and is based on the type of application you want to build.

You can use any combination of the following three methods to define your application's data model.

- **Reuse existing tables and fields to store record data.** For example, use the existing Task table to work on existing record types such as incidents and problems. Typically, this option is only available for tables in the global scope or tables that make their data available to all application scopes.
- **Extend existing tables to store record data.** For example, extend the Task table to create VIP incidents. The extended table inherits the fields from the parent table, but you can add new fields as needed.
- **Create new tables and fields to store record data.** You can use a variety of available tools to create new tables. If you have access to App Engine Studio, use Table Builder to create new tables and fields as you build applications. Alternatively, you can create tables and fields separately from application development using basic Table administration in the ServiceNow AI Platform.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

More resources

These concepts are explained in further detail in the App Engine Studio, Table Builder, and Table administration documentation.

Create a data model for your application

Add data to apps that you build in App Engine Studio.

Table administration

Table administration currently contains more comprehensive functionality and options for managing tables.

Table Builder

Table Builder offers an intuitive way to visually build tables and forms in the ServiceNow AI Platform.

Delegated Development

Delegated development allows designated users without a system admin role to develop or deploy applications on the ServiceNow AI Platform.

Get started

Explore	Configure	Administer
		
Learn about Delegated Development features and concepts.	Configure permissions and users.	Administer permissions and users.

Troubleshoot and get help

- Ask questions and explore other resources for Delegated Development in the ServiceNow Community
- Search the Known Error Portal for known error articles
- Contact Customer Service and Support

Exploring Delegated Development

Delegated Development allows designated users without a system admin role to develop or deploy applications on the ServiceNow AI Platform.

Delegated Development overview

If you have the application-specific admin role or the system-level admin role, you can delegate application development to designated developers at the application level.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Delegated Development users

Users

User	Description
Admin	Admin user can delegate developers.
Application Administrator	If Application administration is enabled, only an application administrator of the target application can delegate developers to an application. If application administration is not enabled, an admin user can delegate developers.

What to explore next

To learn more about configuring and using Delegated Development, see:

- Configuring Delegated Development
- Administer Delegated Development Delegated Development

Delegated development and deployment

Delegated development allows designated users without a system admin role to develop or deploy applications on the ServiceNow AI Platform.

If you have the application-specific admin role or the system-level admin role, you can delegate application development to designated developers at the application level.

Delegated deployment tasks

You can also delegate deployment tasks (application publishing, first-time installation, or update) to developers or non-admin users, such as Change Management personnel. You delegate deployment tasks to specific users at the application level, or through assignment of specific user roles at the instance level.

Assignment Method	Applies to	Available options
Setting deployment permissions in Manage Developers. See Delegate development and deployment permissions to personnel	Specific applications	Publishing and upgrades of specific applications. Publishing options include the application repository, ServiceNow Store, and update sets.
Assignment of deployment user roles to specific persons. See Instance-specific deployment user roles.	Local non-production instance (for example, Development or QA)	First-time installations and upgrades of all applications that contain the same company as the current instance. For example, applications for ABC Company and XYZ Company display on the Application Client page. A user with this role can only install XYZ Company applications when logged in to a XYZ Company

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Assignment Method	Applies to	Available options
		instance. The user cannot install applications for ABC Company.

Application-specific permissions

Developer and deployment permissions are application-specific. For example, a developer who has permission to access all file types for one application does not necessarily have any developer permissions for another application. Administrators must set developer (and optionally deployment) permissions for each application. Administrators must be familiar with application files and the system table structure to set developer permissions. For example, a developer expected to create advanced business rules needs both the **All File Types** and **Allow Scripting** developer permissions.

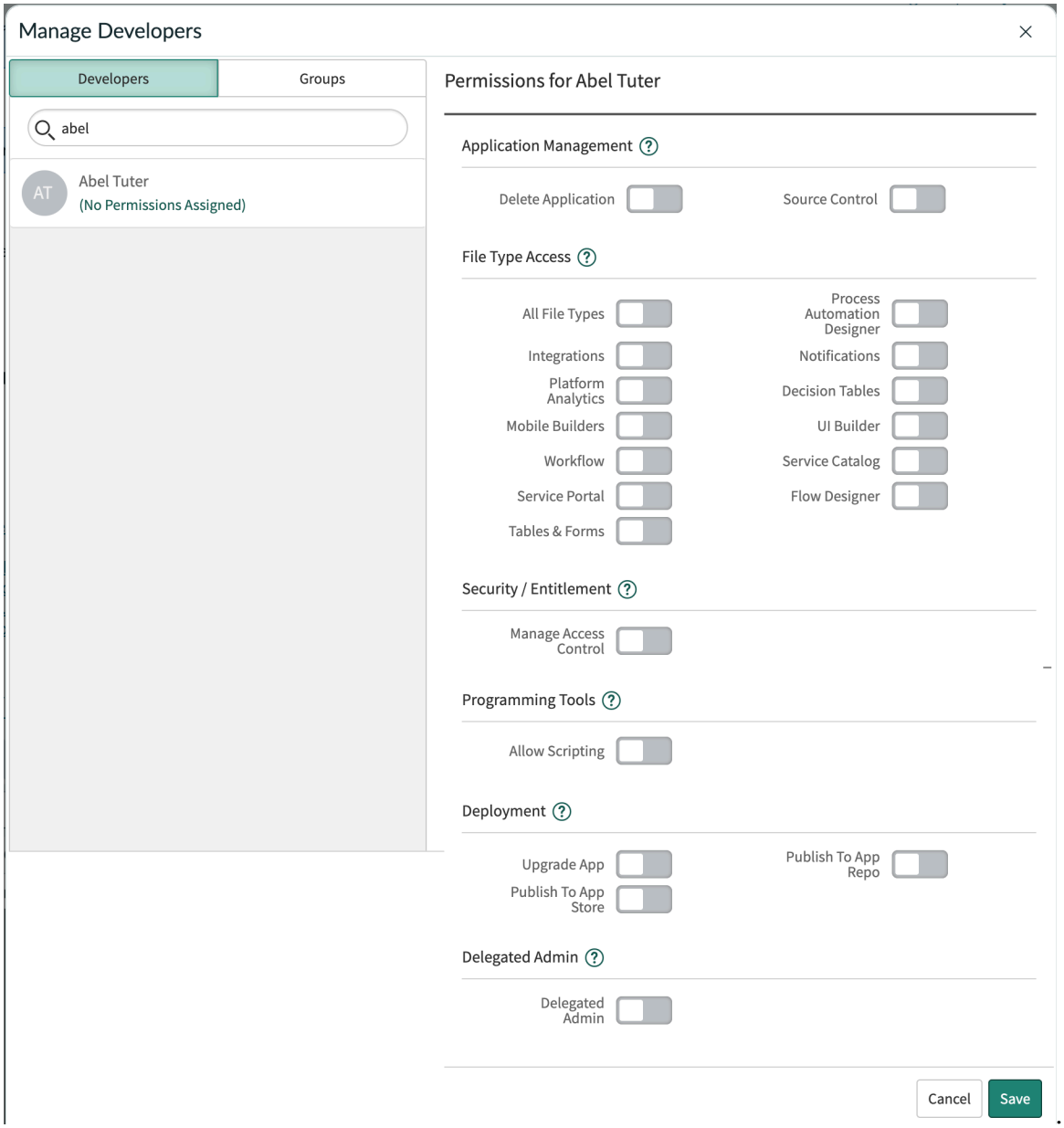
Important: If Application administration is enabled, only an application administrator for the target application can delegate developers for an application. Application administrators do not have system admin privileges. To enable a delegated developer to perform the functions granted in the developer permissions, the delegated developer must also be given the application administrator role.

Setting each permission grants one or more system-managed delegated development roles, allowing system admins to retain control over the system. System admins no longer have to elevate developers (or users who deploy applications) to the system admin role to enable them to develop or deploy applications.

Developer and deployment permissions example

As a system administrator, you want to assign Abel Tuter certain developer and deployment permissions for a specific application in your development instance. For more details on developer and deployment permissions, see Delegate development and deployment permissions to personnel.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Related topics

- Application files
- Install an application or plugin
- Update an application or plugin
- Application sharing

Domain separation and Delegated Development

Domain separation is unsupported in the Delegated Development feature. Domain separation enables you to separate data, processes, and administrative tasks into logical groupings called domains. You can control several aspects of this separation, including which users can see and access data.

Support level: No support

- The domain field may exist on data tables but there is no business logic to manage the data.
- This level is not considered domain-separated.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For more information on support levels, see Application support for domain separation.

Related topics

Domain separation for service providers

Configuring Delegated Development

Configure Delegated Development to assign, delete, display or hide permissions.

Configuration overview

- Assign source control permissions

  Ability to assign full access to source control for a particular scope to a delegated developer. The Source Control menu is only visible if you have the correct permissions for the application that you are working in. Before you begin.

- Assign delete permissions

  Assign the ability to delete an application to a delegated developer.

- Display or hide update set deployment permissions

  Display or hide deployment permissions for update sets from the Manage Developers dialog.

- Instance-specific deployment user roles

  Assign roles that enable non-admin users install or upgrade all applications in specific instances. You delegate these tasks by manually assigning specific user roles per instance.

Assign source control permissions

Ability to assign full access to source control for a particular scope to a delegated developer. The Source Control menu is only visible if you have the correct permissions for the application that you are working in.

Before you begin

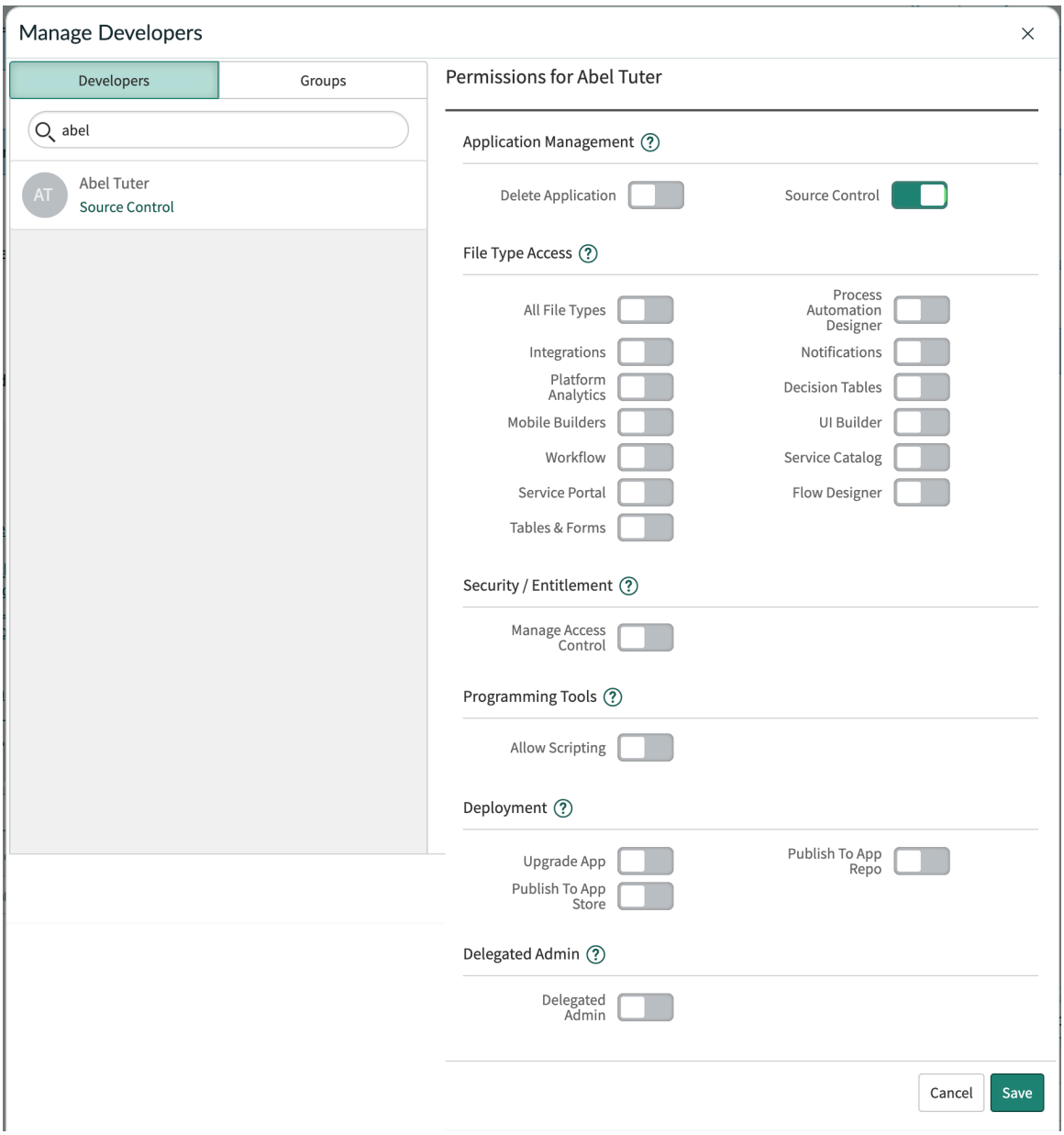
Role required: admin

Procedure

1. Navigate to **All** > **System Applications** > **My Company Applications**.
2. Open an application for which you want to assign delegated developer permissions.
3. As a System Admin, select **File** > **Manager Developers**.
4. Choose a developer to apply delegated developer permissions.
5. Toggle on **Source Control**, and then select **Save**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Source Control



6. The delegated developer can now access the Source Control menu options.

Assign delete permissions

Ability to assign the ability to delete an application to a delegated developer.

Before you begin

Role required: admin

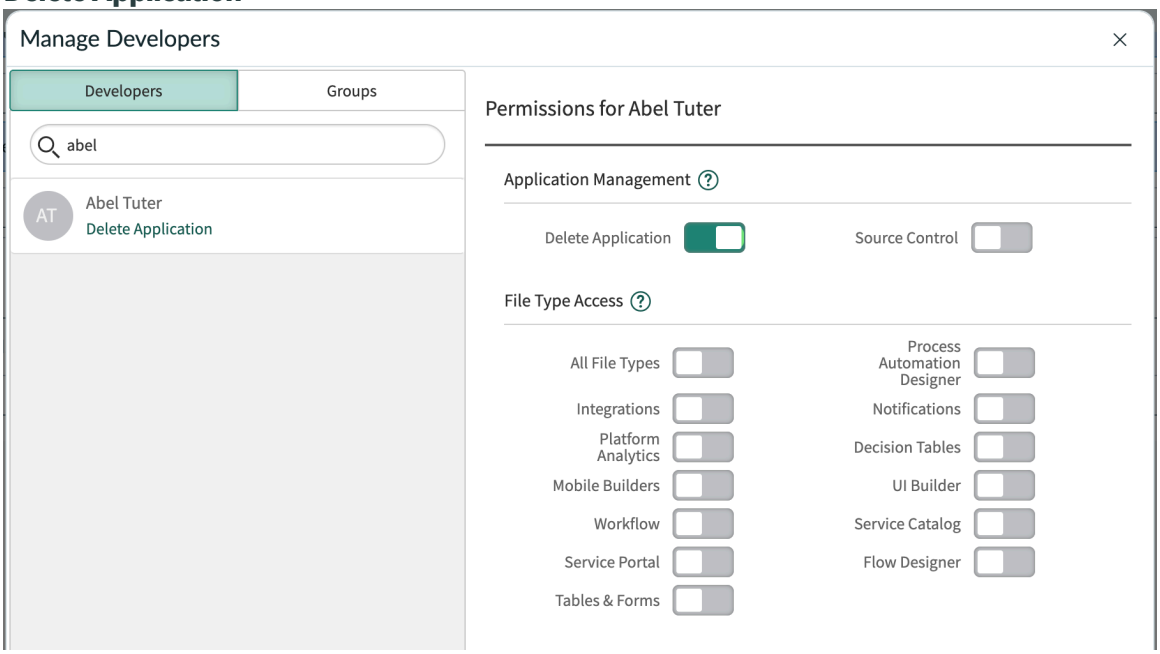
Procedure

1. Navigate to **All** > **System Applications** > **My Company Applications**.
2. Open an application for which you want to assign delegated developer permissions.
3. As a System Admin, select **File** > **Manager Developers**.
4. Choose a developer to apply delegated developer permissions.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

5. Toggle on **Delete Application**, and then select **Save**.

Delete Application



6. The delegated developer can now delete the application.

Display or hide update set deployment permissions

Display or hide deployment permissions for update sets from the Manage Developers dialog.

Before you begin

Role required: admin

About this task

System properties control the visibility of the **Publish To Update Set** and **Manage Update Sets** permissions. By default, both deployment permissions for update sets are hidden.

Procedure

1. In the Navigation filter, enter `sys_properties.list`.
   The entire list of properties in the System Properties [sys_properties] table appears.

   Note: For details on how to add and enable system properties, see Add a system property.

2. Enable or disable specific deployment permissions by adding or updating the following system properties.

Deployment permissions

System property	Description
com.snc.dd.manage_update_set_enabled	Enables or disables display of the **Manage Update Set** permission.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

System property	Description
	The default value of this system property is **false**. To enable the display of this permission, set this value to **true**.
com.snc.dd.publish_to_app_repo_enabled	Enables or disables display of the **Publish To App Repo** permission. The default value for this system property is **true**. To disable display of this permission, set this value to **false**.
com.snc.dd.publish_to_app_store_enabled	Enables or disables display of the **Publish To App Store** permission. The default value for this system property is **true**. To disable display of this permission, set this value to **false**.
com.snc.dd.publish_to_update_set_enabled	Enables or disables display of the **Publish To Update Set** permission. The default value for this system property is **false** by default, which disables display of this permission. To enable the display of this permission, set this value to **true**.
com.snc.dd.upgrade_app_enabled	Enables or disables display of the **Upgrade App** permission. The default value for this system property is **true**. To disable display of this permission, set this value to **false**.

Instance-specific deployment user roles

Assign roles that enable non-admin users install or upgrade all applications in specific instances. You delegate these tasks by manually assigning specific user roles per instance.

For example, you can assign user roles to Change Management personnel that allow them to perform application installations in non-production (development or QA) instances.

Important: Give careful thought to whom, and in what instances you assign installation and upgrade user roles. You may only want to assign these roles to non-system administrators in non-production instances (for example, Developer or QA), but decline to do so in production instances. Given the potential impact of installing and upgrading applications, you may want to leave the responsibility for installations and upgrades in production instances to a system administrator.

To learn more about managing per-user subscriptions, see Managing per-user subscriptions in Subscription Management and contact your account representative.

Add deployment user roles

A system administrator can assign user roles to specific personnel that allow them to perform first application installations only, or install and upgrade applications in a local instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin
Role required: admin

Procedure
1. Navigate to **All** > **User Administration** > **Users** and then open a user record.

   Note: For details on how to assign a role to a user, refer to Assign a user role.

2. In the **Roles** related list, select **Edit**.
3. In the **Collection** list, select the desired deployment roles, and then select **Add**.

Delegated deployment roles

User Role	Description
sn_appclient.app_client_company_installer	Allows a designated person to perform first-time installations of applications displayed on the Application Client page that contain the same company as the current instance. A user with this role cannot install an application for another company. For example, applications for ABC Company and XYZ Company display on the Application Client page. A user with this role can only install XYZ Company applications when logged in to a XYZ Company instance. The user cannot install applications for ABC Company. **Note:** While this role does not grant a user the ability to upgrade applications, the system auto-adds a delegated developer upgrade role after the user does a first-time installation. The user can then perform upgrades for the application in the local instance. For more details, see System-managed developer and deployment roles.
sn_appclient.app_client_user	Allows a designated person to install and upgrade all applications displayed on the Application Client page.

4. Select **Save**.

System-managed developer and deployment roles

Although system admins can still manually assign and remove the user roles, they are encouraged to let the system manage the following delegated developer roles.

To learn more about managing per-user subscriptions, see Managing per-user subscriptions in Subscription Management and contact your account representative.

Role	Description
delegated_developer	User has one or more developer permissions.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Role	Description
Roles that start with an sn_dd prefix (for example, sn_dd_<app_name>_upgrade_app)	User has an application-specific developer permission. The role name indicates the application scope to which i applies. For example, after a user with a sn_appclient.app_client_company_installer role installs a company application, the system automatically grants a sn_dd_<app_name>_upgrade_app delegated deployme role. This role allows the user to upgrade the application when future updates are published to the Application Client page.
glide.security.add_admin_contained_roles_to_system	Default Value: true if the property is true, then all the roles, directly or indirectly contained by the admin role, are added to the system user, including the scoped-admin roles. **Note:** Setting the property to false results in the old behavior, where the system user has the admin role, but not any of the scoped-admin roles contained by the admin role. Example: The admin role contains the *sn_templated_snip.template_snippet_adm* role Old behavior: The system user does not have the *sn_templated_snip.template_snippet_adm* role. New behavior: The system user has the *sn_templated_snip.template_snippet_adm* role and other scoped roles that it contains.

Note: Users with delegated developer roles cannot add or remove the system admin role.

Administer Delegated Development Delegated Development

Perform administrative duties in Delegated Development.

Overview of Delegated Development

- Delegate development and deployment permissions to personnel

  A system administrator can assign a non-administrator user or group as a developer or deployment resource for a specific application.

- Developer and deployment permissions

  Administrators can assign one or more developer and deployment permissions to a group or user for a specific application.

- Remove a developer

  Remove a user as a developer prevents the user from developing, changing, or deploying the application in the current instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Delegate development and deployment permissions to personnel

A system administrator can assign a non-administrator user or group as a developer or deployment resource for a specific application. You can set permissions that designate what specific actions the assigned user can perform in the current instance, or make the user a Delegated Admin to grant access to all permissions at once.

Before you begin

- Role required: admin or application administrator

  If Application administration is enabled, only an application administrator of the target application can delegate developers to an application. If application administration is not enabled, an admin user can delegate developers.

- Records required:
  - Application
  - User
  - Group

Procedure

1. Navigate to **All** > **System Applications** > **My Company Applications**.
2. Select the name of the application to which you want to add developers.
   The system opens the application record.
3. Select **Manage Developers**.

   Note: If the App Collaboration Component is installed, the link appears as **Manage Collaborators**.

   The system displays the Developer Permissions window.

   Note: Developer permissions are available only for scoped apps, not global apps.

4. Select the type of developer (or user without system admin roles) you want to assign.

Option	Description
Developers	To add individual users as developers or de ployers.
Groups	To add all members of a group as developers or deployers.

5. In **Developer Name** or **Group Name**, enter the name of the user or group you want to grant developer or deployment permissions.
6. Select specific developer and deployment permissions for the application, or select **Delegated Admin** to quickly grant access to all permissions for the application.
   For details, see Developer and deployment permissions.
7. Select **Save**.
   The system assigns the designated developer and deployment permissions for the application.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Developer and deployment permissions

Using Manage Developers, administrators can assign one or more developer and deployment permissions to a group or user for a specific application. These permissions designate the specific actions the assigned user can perform for the application.

For example, you might grant permissions that enable a user to upgrade the application, publish to the application repository and ServiceNow Store, but prevent publishing to an update set.

Developer permissions

Note: If the App Collaboration Component is installed, the link appears as **Manage Collaborators**.

Permission	Description
Delete Application	Grants the assigned developer within a scoped app rights to delete the application.
Source Control	Grants the assigned developer full access to source control.
All File Types	Grants the assigned developer access to all application file types, including some not granted by the other options. This permission is equivalent to granting the user the admin role but with some limitations. Specifically, it provides access to all file types that are configured in your application per the Manage Developers task in the Application Creator. For an example of such file types, see the permissions example in Delegated development and deployment.
Playbooks	Grants the assigned developer access to the Playbooks design environment to create processes. Editing activity subflows or actions requires the **Flow Designer** permission.
Integrations	Grants the assigned developer access to web service APIs, REST APIs, data sources, and Integration Hub - Import.
Reporting	Grants the assigned developer access to reports and scheduled reports.
Notifications	Grants the assigned developer access to work with automatic email notifications.
Decision Tables	Grants the assigned developer access to Decision Tables to create decision logic based on multiple if-then rules.
Mobile Builders	Grants the assigned developer access to mobile builders, such as Mobile App Builder.
UI Builder	Grants the assigned developer access to UI Builder to create pages for experiences.
Workflow	Grants the assigned developer access to the Workflow Editor and Activity Creator.
Service Catalog	Grants the assigned developer access to catalog-related file types such as catalog items, record producers, and variables.
Service Portal	Grants the assigned developer access to Service Portal editors and tools.
Workflow Studio	Grants the assigned developer access to the Workflow Studio design environment to create flows and actions. Script action steps require the **Allow Scripting** permission.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Permission	Description
Tables & Forms	Grants the assigned developer access to model and layout related file types such as table columns, form layout, and list layout.
Manage ACLs & Roles	Grants the assigned developer access to security-related file types such as access controls and user roles.
Allow Scripting	Grants the assigned developer write access to script fields such as those in business rules, client scripts, and Workflow Studio script action steps. **Note:** The SNC scripting role is also added.
Manage Collaborators	Grants the assigned developer the ability to invite and manage users and groups. This permission allows the delegated developer to further invite and manage other developers to the application.
Invite Collaborators	Grants the assigned developer the ability to invite users and groups. This permission allows the delegated developer to further invite other developers to the application.
Delegated Admin	Grants access to all Delegated Development permissions for a specific scoped application so that permissions do not need to be granted individually. **Note:** The SNC scripting role is also added.

Deployment permissions

The update set deployment permissions are hidden by default and require a system administrator to enable them with system properties. See Display or hide update set deployment permissions for more information.

The Submit for Deployment, Manage Collaborators, and Invite Collaborators delegated development permission sets are only available with the Developer Collaborator feature. They will not be shown in Manage Developers.

Permission	Description
Upgrade App	Grants a user with an assigned delegated developer role permission to upgrade the associated application after it has been installed in the current instance.
Publish To Update Set	Grants a user with an assigned delegated developer role permission to publish the associated application to an update set in the current instance. **Note:** Users with this permission cannot also have the **Manage Update Set** permission.
Publish To App Store	Grants a user with an assigned delegated developer role permission to publish associated application to the ServiceNow Store in the current instance. **Note:** The **Upgrade App**, **Publish To App Repo**, and **Publish To App Store** permissions display by default. The **Publish To Update Set** permission only displays if manually enabled by a system administrator. For more details, see Display or hide update set deployment permissions.
Manage Update Set	Grants a user with an assigned delegated developer role permission to manage local and retrieved update sets. This permission allows users to create, update,

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Permission	Description
	and delete local update sets as well as preview, resolve conflicts, and commit retrieved update sets. **Note:** Users with this permission cannot also have the **Publish To Update Set** permission.
Publish To App Repo	Grants a user with an assigned delegated developer role permission to publish the associated application to the application repository in the current instance.
Submit for Deployment	Grants a user with assigned delegated developer role permission to submit the associated application for review and deployment.

Remove a developer

Removing a user as a developer prevents the user from developing, changing, or deploying the application in the current instance.

Before you begin

Role required: admin or application administrator

If Application administration is enabled, only an application administrator of the target application can delegate developers to an application. If application administration is not enabled, an admin user can delegate developers.

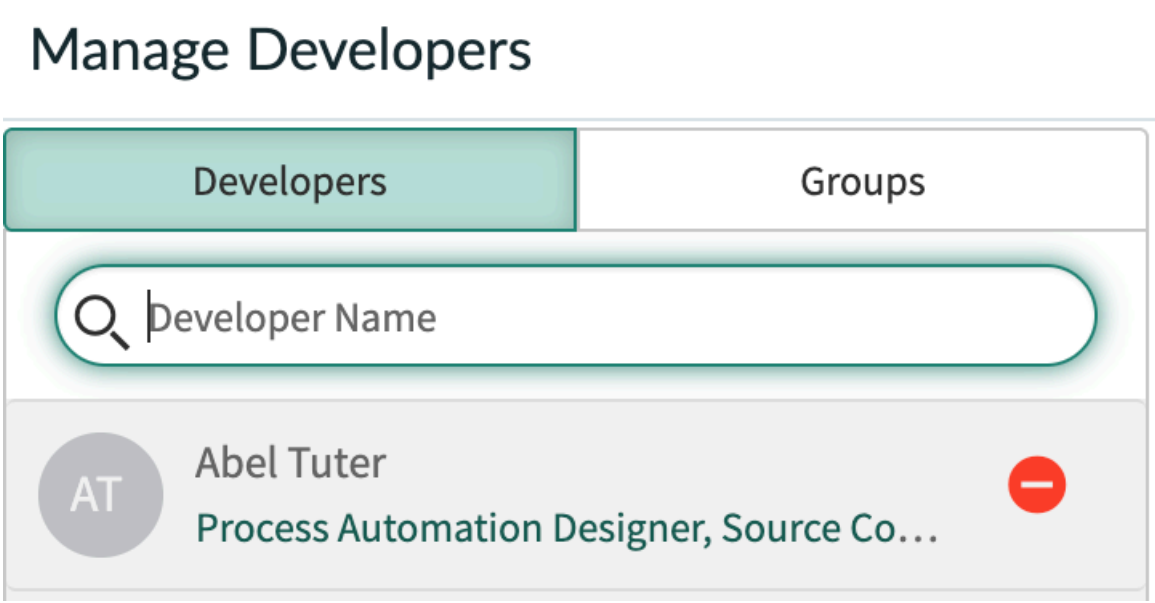
Procedure

1. Navigate to **All** > **System Applications** > **Applications**.
2. Select the application name of the application from which you are removing developers.
   The system opens the application record.
3. Select **Manage Developers**.

   Note: If the App Collaboration Component is installed, the link appears as **Manage Collaborators**.

   The system displays the Developer Permissions window.
4. Point to the developer you want to remove.
   The system displays a minus icon next to the developer name.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



5. Select the minus icon next to the developer name.
   The system removes the developer and any associated application roles.

Team Development

Team Development supports parallel development on multiple, non-production ServiceNow instances.

Team Development provides the following features:

- Branching operations, including pushing and pulling record versions between instances.
- The ability to compare a development instance to other development instances.
- A central dashboard for all Team Development activities.

Get started

Explore	Configure	Administer
Learn about Team Development features and concepts.	Set up and configure Team Development.	Perform administrative duties in Team Development to support parallel development in your instances.

Exploring Team Development

Team Development supports parallel development on multiple, non-production ServiceNow instances.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Team Development overview

Team Development allows developers to work on separate development instances while sharing code and resolving collisions throughout the development process.

After setting up the instance hierarchy, you can develop changes on your local development instance. Use the team dashboard to manage Team Development activities, such as:

- Tracking local changes and determining which changes to promote to the parent development instance.
- Pulling changes from the parent instance and resolving any collisions with local changes.
- Comparing your instance with other development instances and resolving any collisions with other development projects.
- Pushing changes when a feature is tested and ready to promote to the parent development instance.

Developers with admin access to their development instance and the parent instance can use Team Development. For alternative access settings, see Access rights for developers.

Team Development users

Users

User	Description
Admin	Configures Team Development.
Developer	Uses Team Development.

What to explore next

To learn more about configuring and using Team Development, see:

- Configuring Team Development
- Administer Team Development

When to use Team Development

Team Development allows multiple developers to work on applications.

Local changes

The Local Changes table tracks which customized records have current versions that exist on the development instance but not on the parent instance.

Use local changes to collect changes in preparation for a push.

You queue local changes that are ready to push. Each development instance maintains a single queue, regardless of who develops or queues the changes. You ignore local changes that you do not want to push. For example, you may want to ignore changes to the color scheme that visually distinguish a development instance from the production instance. You can remove a change from the queue or stop ignoring a change.

Changing the parent instance or reconciling recreates the list of local changes that have not been queued or ignored. If you had previously queued or ignored a local change, that designation is maintained.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Local change lists

On the team dashboard, the *Local Changes* list shows the local changes that have not been queued for the next push or ignored for all pushes.

The *Ready to Push* list shows the changes that are queued, and the *Ignored* list shows the changes that are ignored. Use any of these methods to navigate a list of local changes.

local changes list

Action	Option
Click the reference icon beside the row	To open the local change record itself.
Click the link in the first column	To open the customized record.
Right-click the row and select **Show Changes Since Last Pull**. An error message appears if a previous version does not exist (for example, in the case of a newly created record). If a previous version is available, you can revert to that version from the comparison window	To view a comparison between the current local version and the version most recently pulled from or pushed to the parent.
Right-click the row and select **Show Application File**	To open the application file for the customized record.
Right-click the row and select **Show Version**	To open the current version record.

Pull exceptions

Pulling ignores versions when certain conditions occur.

Pull exceptions table

Issue	Description
Matched an exclusion policy	An exclusion policy prevents pulling changes for records matching the policy conditions. The pull identifies the changes but does not include versions for these records.
Private properties	A private property is excluded from all Update Sets and pulls.
Collisions	A collision is detected when the pulled version and the current local version both include modifications to the same record. You must resolve all collisions before you can pull.
Previously resolved collisions	When you resolved a collision by accepting either the pulled version or local version of a record, the pull remembers your decision and accepts the version that you indicated as a "previously resolved collision".
Skipped	Pulls skip versions where there is a problem with the version record such as a corrupt or missing version.

Team dashboard

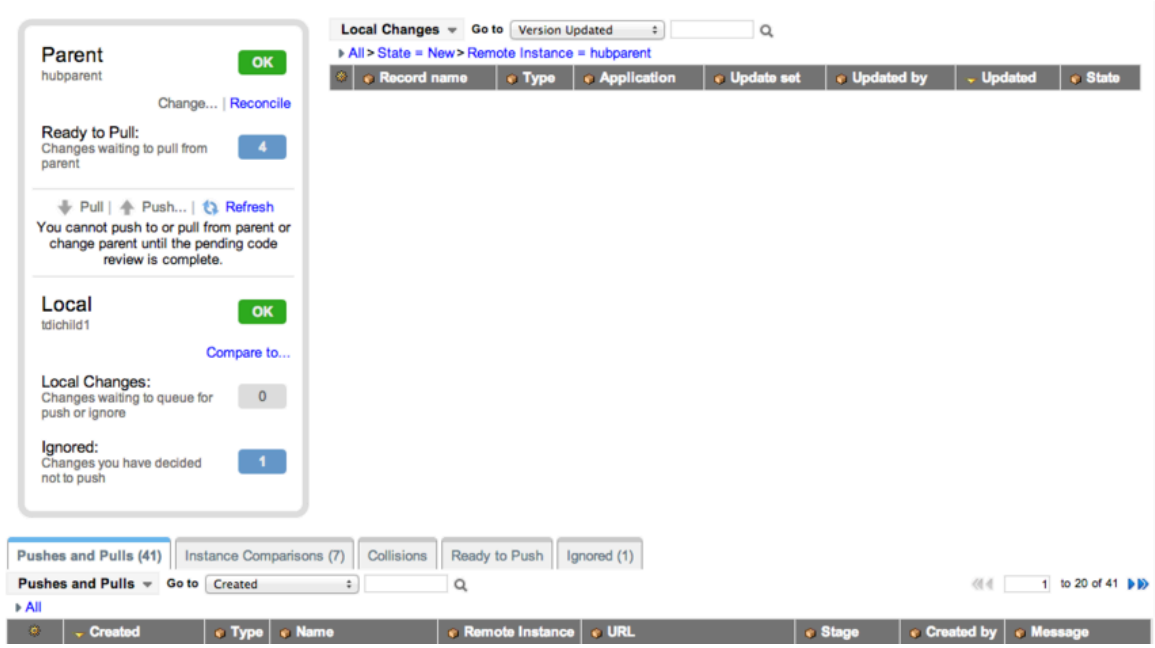
The team dashboard provides a central place to manage all Team Development activities on your development instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

You can track local changes, pull and push changes between the local and parent instances, compare the local instance to other development instances, and resolve any collisions. You can also reconcile with the current parent instance or change the parent instance.

To access the dashboard, navigate to **Team Development** > **Team Dashboard**.

Team dashboard



The control panel in the top left provides status indicators and Team Development actions.

- Parent: indicates the status of the connection to the parent instance. If a problem or warning is detected, point to the indicator to view the error messages, or click the indicator to open the remote instance record.
- **Change**: changes the parent instance. See Changing the Parent Instance.
- **Reconcile**: compares the development instance to the parent instance. See Reconciling.
- Ready to Pull: indicates the number of changes on the parent that have not been pulled to the local instance.
- **Pull**: initiates a pull. See Pulling Versions.
- **Push**: opens a page that allows you to review the changes before a push. See Pushing Versions.
- **Refresh**: updates the status indicators on the control panel. The dashboard updates only when you reload or refresh the page.
- Local: indicates the status of the most recent comparison with another instance. If collisions are detected, click the indicator to open the list and resolve the collisions. See Resolve a collision in Team Development.
- Collisions: appears only if any local changes collide with versions pulled from the parent and indicates the number of collisions. Click the indicator to open the list and resolve the collisions. See Resolve a collision in Team Development.
- **Compare to**: allows you to select another development instance to compare with the local instance. See Comparing to Peer Instances.
- Ready to Push: indicates the number of local changes that are queued for the next push. See Queuing and Ignoring Local Changes.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Local changes: indicates the number of local changes that have not been queued or ignored. Click the indicator to open a list of these changes.
- Ignored: appears only if any local changes are ignored and indicates the number of ignored changes. Click the indicator to open a list of these changes.

The team dashboard includes lists for tracking local changes and viewing the history of Team Development activities.

- Local changes: lists the local changes that have not been queued or ignored.
- Pushes and Pulls: provides a history of pushes and pulls. Expand a row to see the customized records for which versions were transferred as part of the push or pull.
- Instance Comparisons: provides a history of comparisons with other development instances.
- Collisions: lists the collisions that must be resolved before the next pull or push. You can right-click a row and select **Resolve Collision**. See Resolving Collisions.
- Ready to Push: lists the local changes that have been queued for the next push.
- Ignored: lists the local changes that are ignored for all pushes.

Related topics

Administer Team Development

Pull a version

Pull exceptions

Resolve a collision in Team Development

Resolve multiple collisions

Local changes

Local change lists

Queue a local change for a push

Ignore a local change

Back out a local change

Push a version

Approve or reject a push

Check the review status of a pushed change

Cancel a code review request

Compare to peer instances

Change the parent instance

Reconcile changes

Approve or reject a push

Code reviewers must approve or reject a push from the Team Development application.

Before you begin

Role required: none.

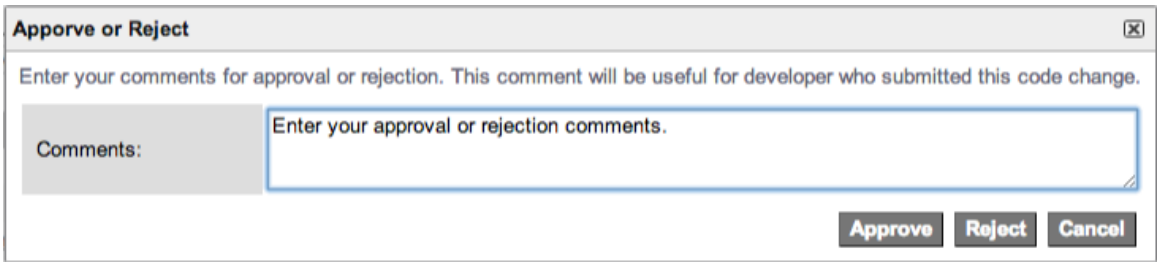
About this task

Although reviewers can see the individual versions within a push, they must approve or reject the push as a whole.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Log in to the parent instance that requires code review.
2. Navigate to **Team Development** > **Code Review Requests**.
3. Select a change in the *Awaiting Code Review* stage.
4. Review the changes in the *Push or Pull Versions* related list.
5. Click **Approve or Reject**.
6. **Optional:** Enter review comments in *Comments*.
   These comments are visible to anyone who can see the Pushes and Pulls history.
7. Click either **Approve** or **Reject**, as appropriate.



Note: The *URL* and *Remote Instance* fields list the address and name of the instance where the change originated.

Back out a local change

Back out all local changes and restore the last version reconciled with the parent instance.

Before you begin

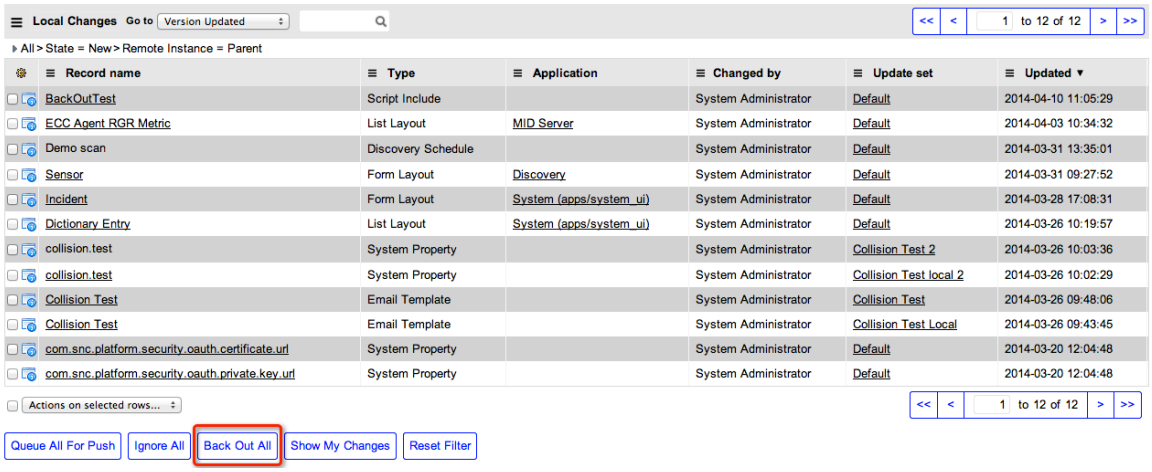
Role required: none.

Procedure

1. Define a parent instance.
2. Pull changes from the parent instance.
3. Navigate to **Team Development** > **Team Dashboard**.
4. Filter the *Local Changes* list to show only the changes that you want to back out.
5. Do one of the following actions:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Click **Back Out All**.
- Right-click the local change you want to back out, and then click **Back Out**.



Cancel a code review request

Developers can cancel any push they submitted that is in the *Awaiting Code Review* stage.

Before you begin

Role required: none.

About this task

Canceling a request sets the push to the **Code Review Request Cancelled** stage on the submitting instance. The submitting instance retains a version history of the push but the parent instance does not.

Procedure

1. Log in to the instance that pushed the changes.
2. Navigate to **Team Development** > **Pushes and Pulls**.
3. Filter for the push you want to cancel.

> **Note:** You cannot cancel a push that has been approved or rejected.

4. Select the Push or Pull record.
5. Click **Cancel Code Review**.

Change the parent instance

If it becomes necessary to modify the instance hierarchy, you can change the parent for a development instance.

Before you begin

Role required: none.

About this task

Changing the parent initiates a complete comparison between the development instance and the new parent instance. To optimize comparison speed and reduce the number of collisions and local changes that need review afterwards, ensure that the new parent instance was cloned recently from an appropriate instance (for example, the production instance). Before you change

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

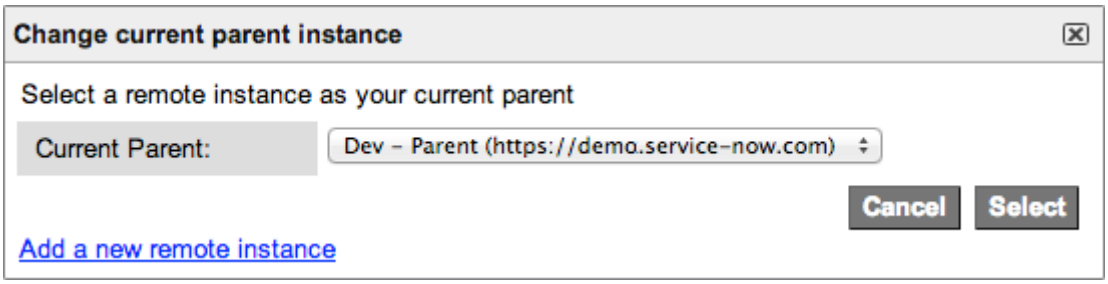
the parent instance, ensure that the change does not conflict with your change management process or other development efforts.

To change the parent for a development instance:

Procedure

1. On the development instance, navigate to **Team Development** > **Team Dashboard**.
2. In the control panel, click **Change**.
3. Select the remote instance you want to use as the parent and click **Select**.

   Alternatively, click the link to define a new remote instance. Then, repeat steps 1–3 and select the remote instance you defined.



   The system initiates a reconcile, which compares the local instance to the parent, and then generates the list of local changes and calculates the number of changes that are ready to pull from the parent.

4. On the completion page, click **Team Dashboard**.
5. Pull versions from the parent instance and resolve any collisions.
6. Review the local changes list and queue or ignore changes, as appropriate.

Check the review status of a pushed change

If the parent instance requires pushed changes to undergo code review, changes are placed in the *Awaiting Code Review* stage.

Before you begin
Role required: none.

About this task
If you configure the parent instance to send notifications, it sends the submitting developer a notification when the pushed changes are approved or rejected. Developers can also manually check the status of their pushed changes from the *Pushes and Pulls* module on the submitting instance.

Procedure

1. Log in to the instance that submitted code for review.
2. Navigate to **Team Development** > **Pushes and Pulls**.
3. Filter for the push you want to review.

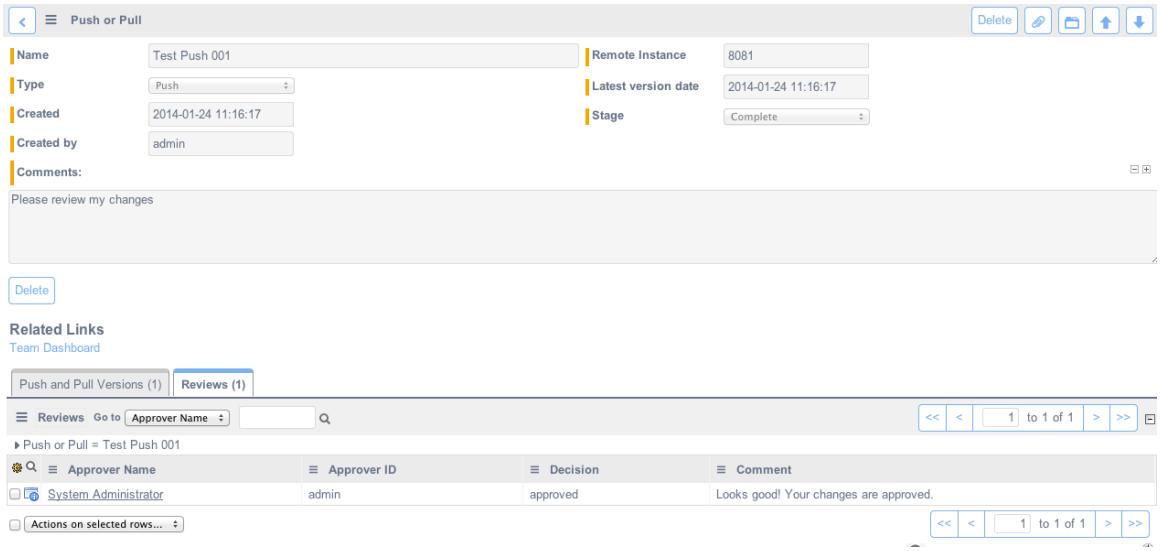
   - Pushes in the *Complete* stage are approved and applied to the parent instance.
   - Pushes in the *Collided* stage are rejected because of a collision.
   - Pushes in the *Awaiting Code Review* stage are awaiting review.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Pushes in the *Code Changes Rejected* stage are rejected by a reviewer.
- Pushes in the *Code Review Request Canceled* stage are canceled by the submitting developer.

4. Click the **Reviews** related list to see the following information.

   - Who submitted a review decision.
   - What the decision was: either approved or rejected
   - What comments if any the reviewer provided.



Compare a pushed version to a local version

Code reviewers can compare the pushed versions to the local versions to see the potential effect of incoming changes.

Before you begin

Role required: none

Procedure

1. Log in to the instance requiring code review.
2. Navigate to **Team Development** > **Code Review Requests**.
3. Select a change in the *Awaiting Code Review* stage.
4. Review the changes in the *Push or Pull Versions* related list.
5. Right-click a row in the list and click **Compare to Current**.
   A comparison of the differences between the pushed and local versions appears.

Related topics

Merge tool

Compare to the current version

Compare two versions of an article

Resolve conflicts for an individual record

Resolve a collision in Team Development

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Revert a change

View customizations and compare with current version

Compare to peer instances

You can compare the local instance to any other remote instance and commit any current versions from the remote instance on your development instance.

Before you begin

Role required: none.

About this task

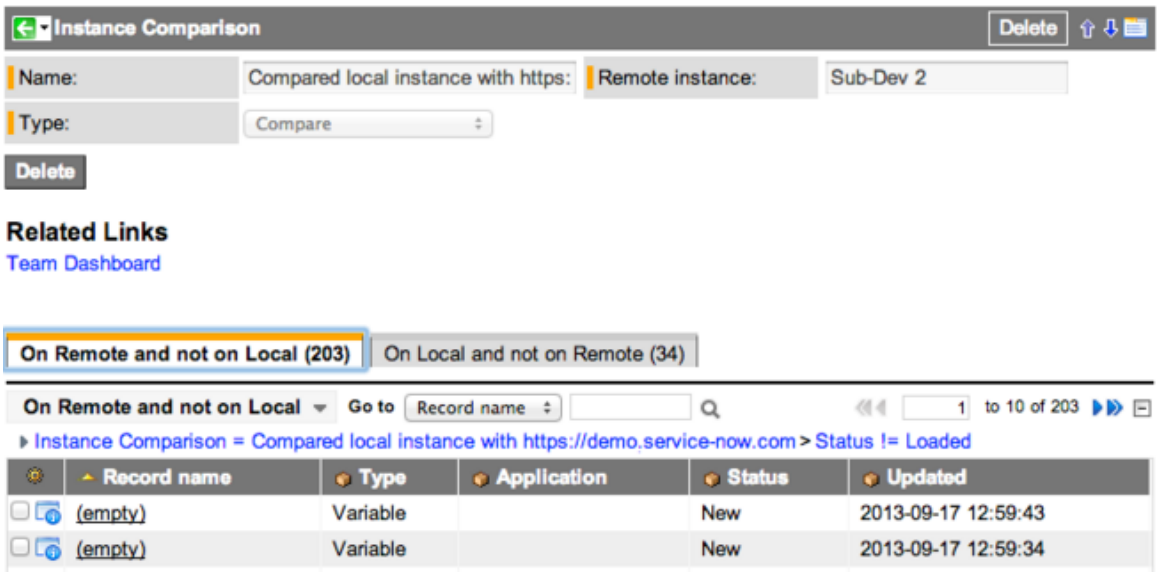
Comparing allows you to share code between instances without pushing to a common parent.

Comparing instances does not automatically commit any versions on the local instance. It initiates a full comparison of all changes on the remote instance and all changes on the local instance, and then reports which customized records have different current versions. You can selectively commit a version from the remote instance or compare it with the version on your local instance. You can delete the instance comparison record when you finish evaluating the differences.

To compare the local instance to a peer instance:

Procedure

1. Ensure that the peer instance is defined as a remote instance.
2. Navigate to **Team Development** > **Team Dashboard**.
3. In the control panel, click **Compare to**.
4. Select the peer instance you want to compare to the local instance and click **Compare**.
5. On the completion page, click **Show Results**.
   The instance comparison record opens.



6. Review the *On Remote and not Local* related list, which shows the customized records where the current version on the peer instance is not on the local instance.

   For each customized record, you can:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Compare the current remote version to the current local version by right-clicking a row and selecting **Compare to Current**.
- Load the current remote version as the current local version by right-clicking a row and selecting **Load This Change**.

Ignore a local change

Ignoring a local change prevents updates to a record from generating new versions in the Local Changes list.

Before you begin

Role required: none.

About this task

An ignored local change always points to the current version for the record. You cannot push ignored records to another instance.

Local change action list

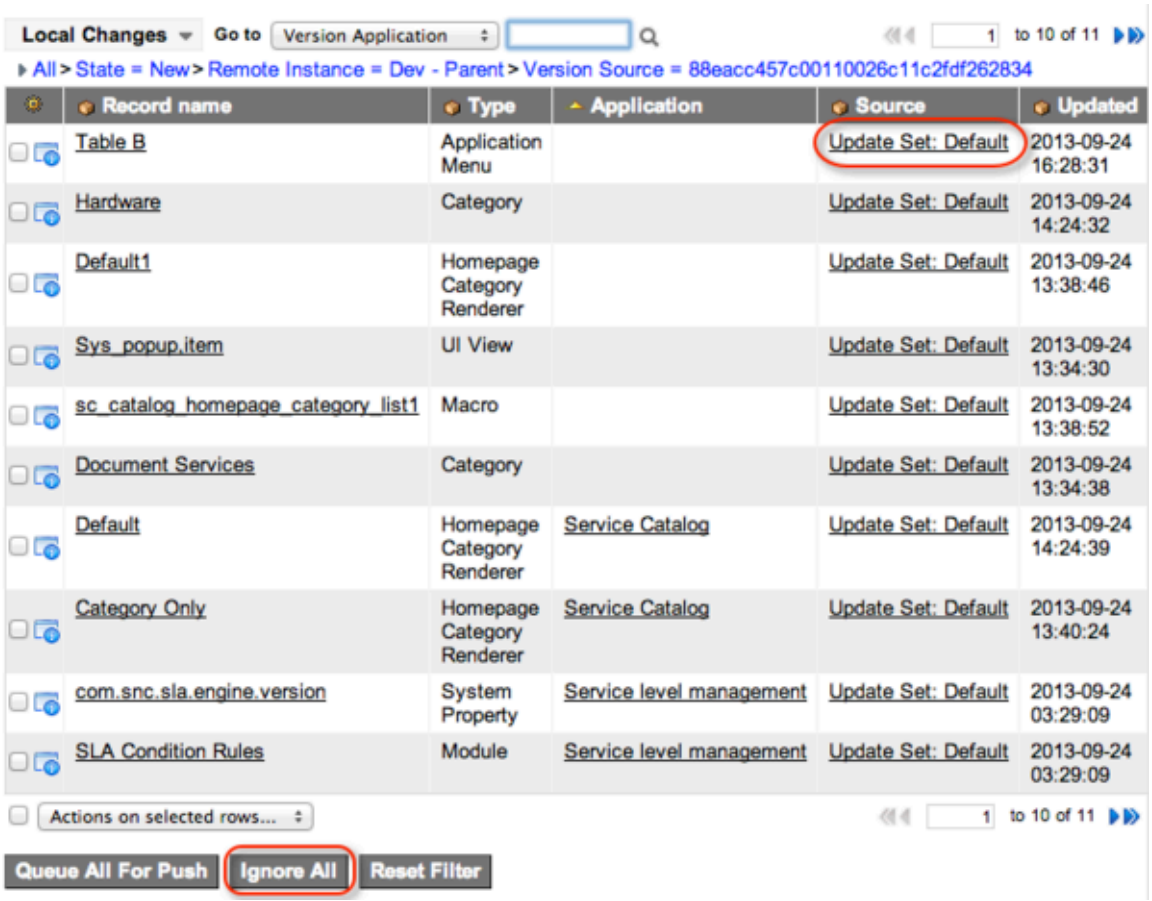
Action	Result
Ignore a record that has a version queued for push	The queued change is deleted
Ignore a record that has a version queued for code review	The queued change is deleted
Pull changes for an ignored record	Collision
Resolve a collision by taking the parent version	There is no longer a local change to ignore
Resolve a collision by keeping the local version	The ignored change remains on the local instance

Procedure

1. Navigate to **All** > **Team Development** > **Team Dashboard**.
2. Filter the *Local Changes* list to show only the changes that you want to ignore.

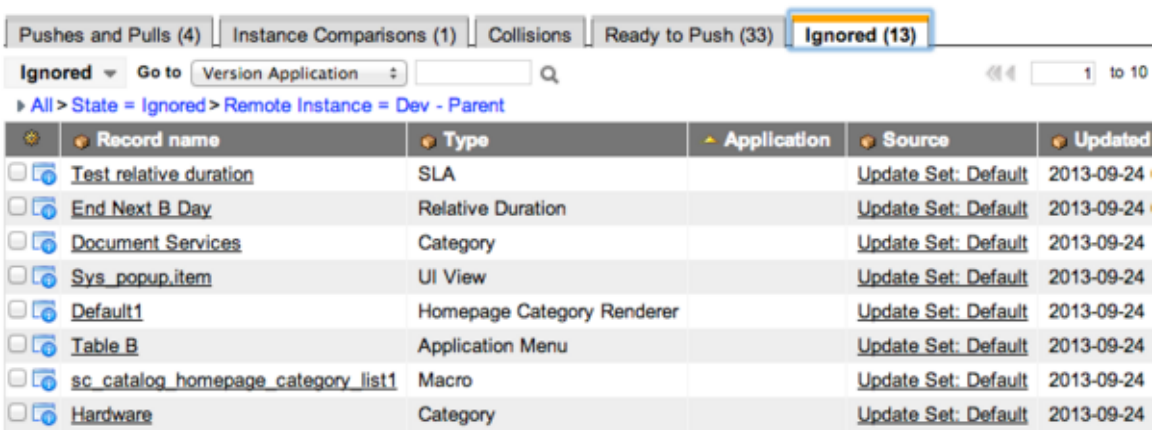
   For example, filter the list to show all changes in the *Default* Update Set.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



3. Click **Ignore All**.
4. Review the *Ignored* list to ensure that the correct changes are ignored.
   This step is a recommended best practice.

Option	Description
To stop ignoring changes	Select the check boxes beside the rows and select **Do Not Ignore** from the *Actions* choice list.
To stop ignoring changes and add them to the queue instead	Select the check boxes beside the rows and select **Queue for Push** from the *Actions* choice list.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Pull a version

Pulling retrieves versions of customized records from the parent instance and adds them on the development instance. Pulling does not retrieve any versions for changes made by system upgrades, but it retrieves all versions for changes made by users, not just the current version.

Before you begin

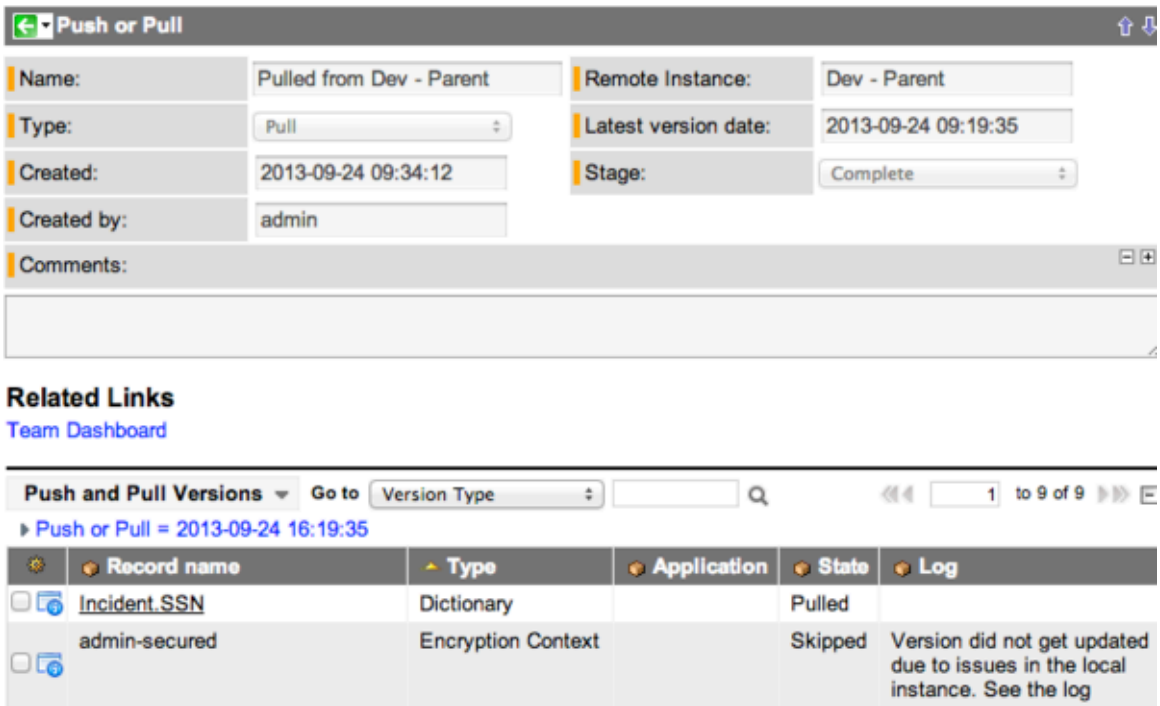
Role required: none

About this task

Pulling retrieves all versions for changes made by users that have not already be pulled onto the development instance, and you cannot choose which versions to pull. The first time you pull from a parent instance, the pull retrieves all versions for changes made by users. Subsequent pulls retrieve the new versions since your last pull. Each pull is recorded in the Push or Pull `[sys_sync_history]` table on the development instance. Historical versions are saved with a state of *History*.

Procedure

1. Navigate to **All** > **Team Development** > **Team Dashboard**.
2. In the control panel, click **Pull**.
3. On the completion page, click **Show Results**.
   The *Push and Pull Versions* related list for the Push or Pull form shows the customized records for which versions were retrieved and indicates if any pull exceptions exist.



4. Resolve any collisions.

Push a version

Pushing promotes changes from the development instance to the parent instance and commits the current version of a customized record on the development instance as the current version on the parent instance.

Before you begin

Role required: none

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

About this task

Pushing adds only the current development version to the parent, not all the development versions.

> **Note:** Updates to records from different applications cannot be pushed/pulled in the same push/pull. To resolve the error in the case that updates to other applications are mixed in: De-queue the updates to other applications. Push for one application. Re-queue the updates to one application. Push and then repeat as needed.

Pushing creates a local Update Set on the parent that is marked as complete. Pushed changes are also tracked as local changes on the parent. Therefore, you can promote changes through your development and test hierarchy by transferring the Update Set or by pushing the local changes. Each push is recorded in the Push or Pull table on the development instance.

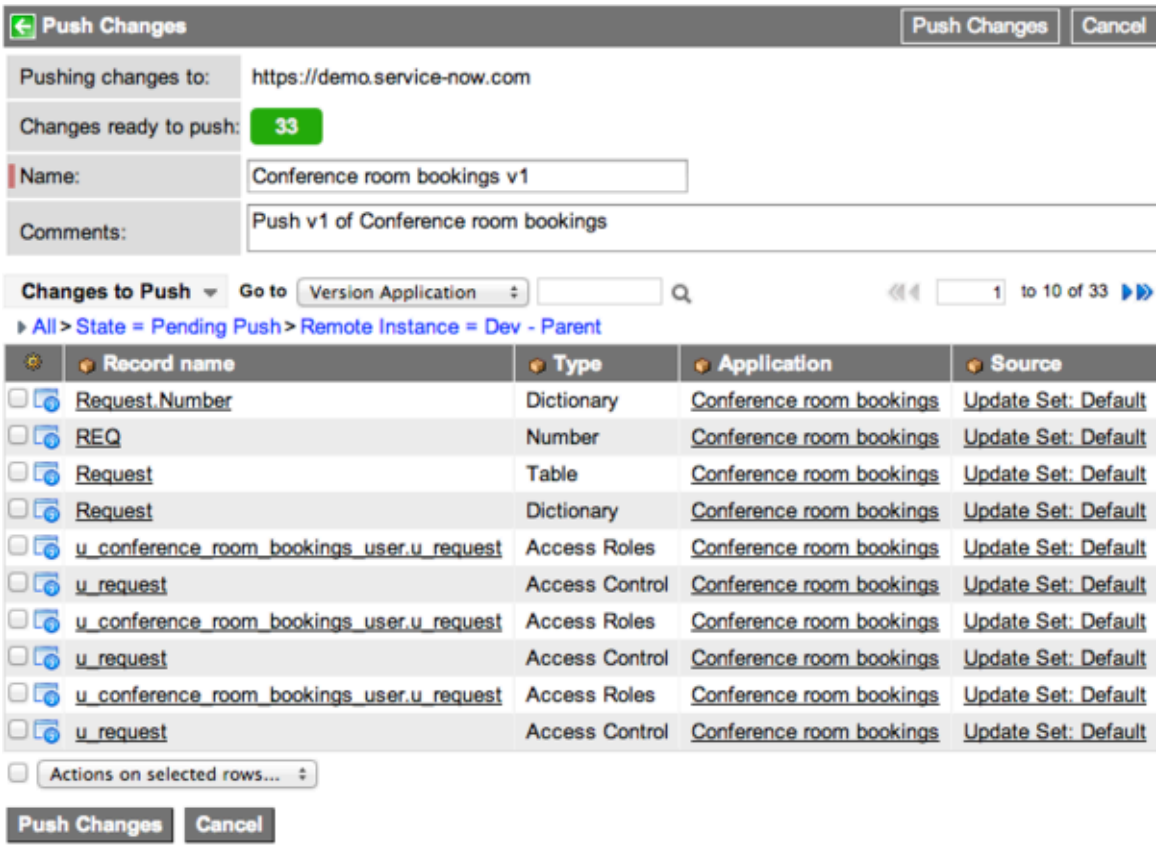
Procedure

1. Navigate to **All** > **Team Development** > **Team Dashboard**.
2. Queue the local changes that are ready to push.
3. Pull versions from the parent instance and resolve any collisions.

   You cannot push changes to the parent instance if collisions are detected.
4. In the control panel, click **Push**.
   The Push Changes page opens.
5. Provide a *Name* for the changes.
6. Review the list of changes to ensure that the correct changes are included.

Option	Description
To remove changes that you do not want to push	Select the check boxes beside the rows and select **Do Not Push** from the *Actions* choice list
To add changes	Click **Cancel** and repeat the procedure from step 2

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



7. **Optional:** Edit the name.
   The name identifies the push record on the development instance and the local Update Set record on the parent instance.
8. **Optional:** Enter comments.
   The comments are added to the push record on the development instance and the local Update Set record on the parent instance.
9. Click **Push Changes**.

   The system initiates a pull to ensure that there are no collisions before the push proceeds.

   - If collisions are detected, the push is automatically canceled and you must repeat the procedure from step 3.
   - If no collisions are detected, the changes are staged on the parent instance. On the parent, each version is validated and then committed in the correct order to maintain dependencies between records. For example, a new table is committed before a field on that table to ensure the field is properly created.

   Note: You cannot push if there is a version conflict between instances or the pushing instance has changes in the *Awaiting Code Review* stage.

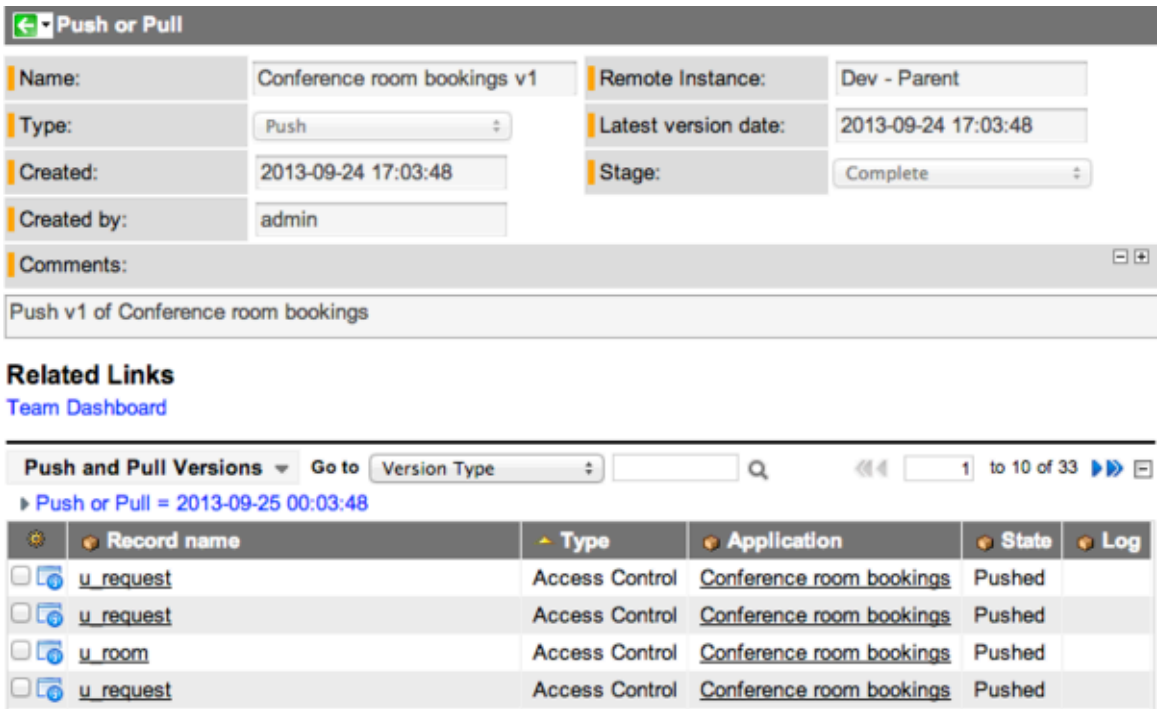
10. On the completion page, click **Show Results**.
11. Review the push record for any errors or skipped changes.

    - Changes with a state of *Pushed* were committed on the parent instance.
    - Changes with a state of *Skipped* were not committed on the parent instance and remain queued as local changes on the development instance.
12. For each skipped change, review the log message to determine why the change was skipped.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Develop any changes that are necessary to commit the desired version on the parent instance, and then push them. Some examples of why a change may be skipped include:

- A table does not exist on the parent because it was created when you activated a plugin on the development instance. Ensure the plugin is activated on the parent and push the change again.
- An error occurred during the push. Try to push again.
- The current version is invalid. Revert to a previous version and make the change again to ensure the version is valid
- An error occurred on the parent during the push. The *Log* field on the push record contains the exception message. Review the system logs on the parent instance and troubleshoot any problems with the instance.



Back out a push

Application developers can back out a push to remove unwanted changes.

Before you begin

Role required: none.

Procedure

1. Navigate to **Team Development** > **Pushes and Pulls**.
2. Select the push to back out.
3. Click **Back Out**.
4. Click **OK** when the confirmation message appears.

Queue a local change for a push

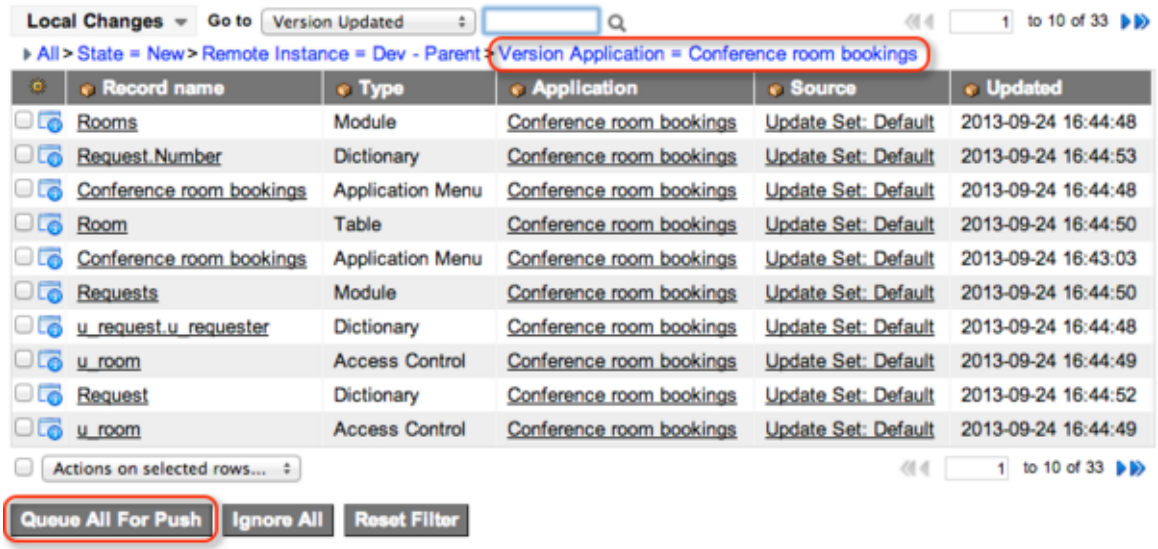
Application developers can queue a local change for a push to ensure the changes are available to other developers.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin
Role required: none

Procedure
1. Navigate to **All** > **Team Development** > **Team Dashboard**.
2. Filter the *Local Changes* list to show only the changes that are ready to push.

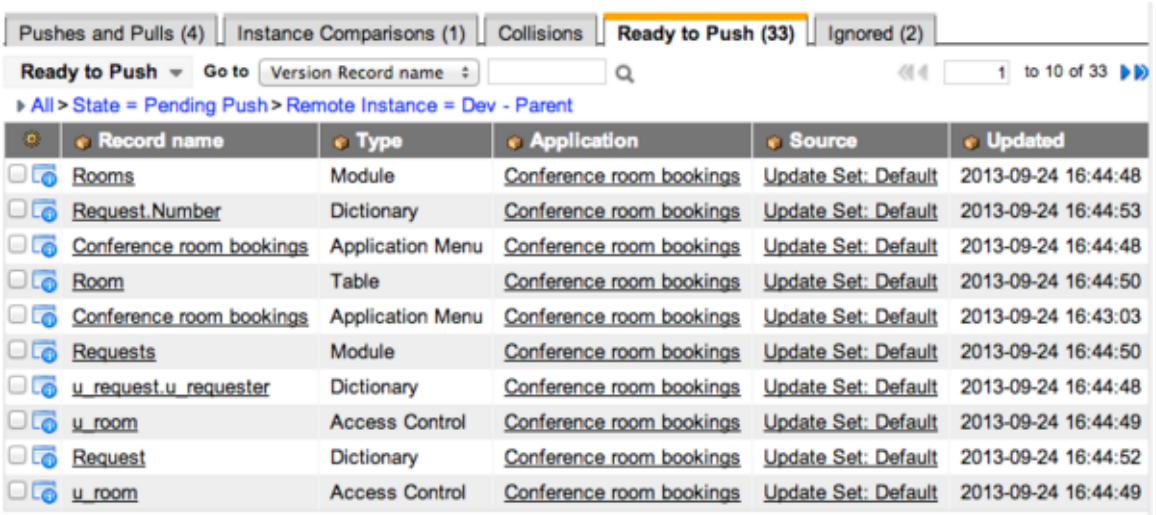
   For example, filter the list to show only the changes associated with a particular application.



3. Click **Queue All For Push**.
4. Review the *Ready to Push* list to ensure that the correct changes are in the queue.
   This step is a recommended best practice.

Option	Description
To remove changes from the queue	Select the check boxes beside the rows and select **Do Not Push** from the *Actions* choice list.
To remove changes from the queue and choose to ignore them instead	Select the check boxes beside the rows and select **Ignore This Change** from the *Actions* choice list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Record name	Type	Application	Source	Updated
Rooms	Module	Conference room bookings	Update Set: Default	2013-09-24 16:44:48
Request.Number	Dictionary	Conference room bookings	Update Set: Default	2013-09-24 16:44:53
Conference room bookings	Application Menu	Conference room bookings	Update Set: Default	2013-09-24 16:44:48
Room	Table	Conference room bookings	Update Set: Default	2013-09-24 16:44:50
Conference room bookings	Application Menu	Conference room bookings	Update Set: Default	2013-09-24 16:43:03
Requests	Module	Conference room bookings	Update Set: Default	2013-09-24 16:44:50
u_request.u_requester	Dictionary	Conference room bookings	Update Set: Default	2013-09-24 16:44:48
u_room	Access Control	Conference room bookings	Update Set: Default	2013-09-24 16:44:49
Request	Dictionary	Conference room bookings	Update Set: Default	2013-09-24 16:44:52
u_room	Access Control	Conference room bookings	Update Set: Default	2013-09-24 16:44:49

Note: For the *Local Changes* list, click **Reset Filter** to remove any filter conditions you added and see all the local changes that have not been queued or ignored.

Reconcile changes

Reconciling first compares the local instance to the parent, and then generates the list of local changes and calculates the number of changes that are ready to pull from the parent.

Before you begin

Role required: none.

About this task

A reconcile occurs automatically whenever you select a parent instance. You may need to manually reconcile after an external disruptive event on the parent instance, such as a clone or failover.

Note: This process may take a while to complete depending on the size and age of the instance.

Procedure

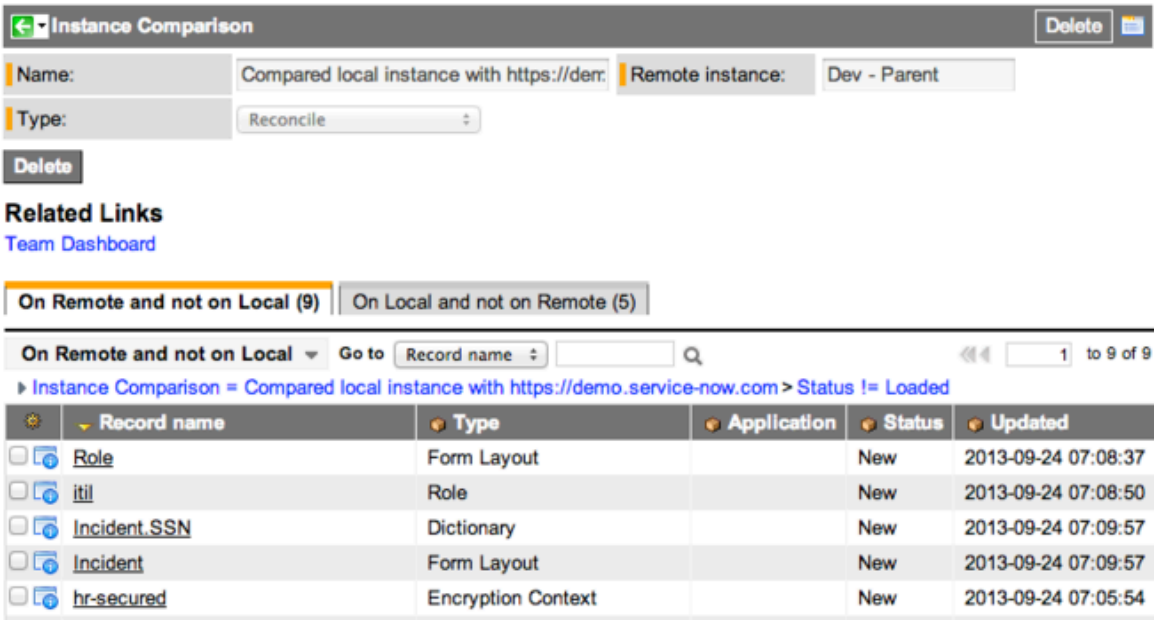
1. Navigate to **All** > **Team Development** > **Team Dashboard**.
2. In the control panel, click **Reconcile**.
3. In the confirmation dialog box, click **OK**.

   The list of local changes that have not been queued or ignored is recreated. If you had previously queued or ignored a local change, that designation is maintained.
4. **Optional:** On the completion page, click **Show Results**.

   Review the instance comparison record.

   - The *On Remote and not Local* related list shows the versions that are ready to pull from the parent.
   - The *On Local and not on Remote* related list shows the local versions that are ready to queue or ignore.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



5. Click **Team Dashboard**.
6. Pull versions from the parent instance and then resolve any collisions.
7. Review the local changes list and queue or ignore changes, as appropriate.

Resolve a collision in Team Development

A collision is detected when the pulled version and the current local version are modifications of a different version, indicating that someone else has modified the same record that you have modified. The team dashboard displays the number of collisions between the local and the parent instance.

Before you begin

Role required: none

About this task

To ensure that your changes do not conflict with other development efforts, you should resolve collisions as soon as they are identified. You must resolve all collisions before you can pull or push.

Procedure

1. Navigate to **All** > **Team Development** > **Team Dashboard**.
2. In the control panel, click **Collisions** or click the count of collisions.
   A list of collisions opens.
3. Right-click a row and select **Resolve Collision**.
   Alternatively, open the record and click the **Resolve Collision** related link.
   The Resolve Collision page displays a comparison between the version that was pulled from the parent and your local record. The page highlights the differences.
4. Review the differences and perform an action.

Option	Action
To maintain the local record as the current version	Click **Use Local Version**. The pulled version is added to the version history for the record.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Action
To load the version pulled from the parent as the current version	Click **Use Pulled Version**.
To move a setting from the selected version to update the current version **Note: Some types of record do not support this method. See Limitations on updating records for more information.**	To move a change, click the **>** (Move Right) button for the field. To work with scripts and text fields, click in the field and modify the text as needed. When the records meet your needs, click **Save Merge and Resolve Collision**.

The system performs that action and also clears the collision for future push/pulls.

5. Repeat the process for every remaining collision.

Result

The system saves the merged changes and resolves the collision.

Related topics

- Compare to the current version
- Compare a pushed version to a local version
- Compare two versions of an article
- Resolve conflicts for an individual record
- Revert a change
- View customizations and compare with current version

Limitations on updating records

There are some types of records that you cannot merge while resolving differences on the Compare to Current and Resolve Collision pages.

Record types that allow a choice only between reverting or accepting the pulled or current record

The following record types do not allow you to merge individual values. Instead, differences involving the following record types display a read-only comparison and allow a choice between updating and reverting:

- sys_choice [Choice]
- sys_choice_set [Choice Set]
- sys_ui_form [Form]
- sys_ui_list [List]
- sys_ui_related_list [Related List]
- sys_ui_section [Form Section]
- wf_workflow [Workflow]
- wf_workflow_version [Workflow Version]

Team Development, Resolve Collision page: **Use Pulled Version** and **Use Local Version** options.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Upgrade History, Compare to Current page: Comparing non-current update versions to current update version. Allows only **Revert to Base System** option.

Field types that do not support merging

The following field types do not support individual merging between versions or updates:

- auto_increment [Auto Increment]
- auto_number [Auto Number]
- breakdown_element [Breakdown Element]
- catalog_preview [Catalog Preview]
- collection [Collection]
- color_display [Color Display]
- composite_field [Composite Field]
- compressed [Compressed]
- counter [Counter]
- currency [Counter]
- data_array [Data Array]
- data_object [Data Object]
- data_structure [Data Structure]
- date [Other Date]
- datetime [Basic Date/Time]
- days_of_week [Days of Week]
- document_id [Document ID]
- due_date [Due Date]
- Email [Email]
- external_names [External Names]
- field_list [Field List]
- float [Floating Point Number]
- glide_action_list [UI Action List]
- glide_precise_time [Precise Time]
- glide_var [Glide Var]
- image [Basic Image]
- index_name [Index Name]
- int [Integer String]
- integer_time [Integer Time]
- ip_address [IP Address]
- journal [Journal]
- journal_input [Journal Input]
- journal_list [Journal List]
- long [Long Integer String]

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- mask_code [Mask Code]
- metric_absolute [Metric Absolute]
- metric_counter [Metric Counter]
- metric_derive [Metric Derive]
- metric_gauge [Metric Gauge]
- mid_config [MID Server Configuration]
- month_of_year [Month of Year]
- multi_small [Multiple Line Small Text Area]
- name_values [Name/Values]
- nl_task_int1 [NL Task Integer 1]
- order_index [Order Index]
- password [Password (1 Way Encrypted)]
- percent_complete [Percent Complete]
- ph_number [Phone Number]
- phone_number [Phone Number (Unused)]
- phone_number_e164 [Phone Number (E164)]
- price [Price]
- reference_name [Reference Name]
- related_tags [Related Tags]
- reminder_field_name [Reminder Field Name]
- repeat_count [Repeat Count]
- repeat_type [Repeat Type]
- replication_payload [Replication Payload]
- schedule_date_time [Schedule Date/Time]
- short_field_name [Short Field Name]
- short_table_name [Short Table Name]
- Slushbucket [slushbucket]
- source_id [Source ID]
- source_name [Source Name]
- source_table [Source Table]
- string_boolean [<none>]
- sys_class_name [System Class Name]
- sysrule_field_name [System Rule Field Name]
- table []
- text []
- time []
- timer [Timer]
- translated [Translated]
- tree_code [Tree Code]

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- tree_path [Tree Path]
- user_image [Image]
- user_input [User Input]
- variables [Variables]
- version [Version]
- video [Video]
- week_of_month [Week of Month]
- wide_text [Wide Text]
- wms_job [WMS Job]
- workflow [Workflow]

Related topics

Resolve a collision in Team Development

Resolve multiple collisions

You can resolve multiple collisions without reviewing the differences between the local and pulled versions.

Before you begin
Role required: none.

Procedure

1. Navigate to **All** > **Team Development** > **Team Dashboard**.
2. In the control panel, click the number of collisions.
   A list of collisions opens.
3. Select the check boxes beside the rows you want to resolve.
4. In the *Actions* choice list, use one of the following methods to resolve the collision.

Option	Description
To load the version pulled from the parent as the current version for all selected collisions	Select **Use Pulled Version**
To maintain the local version (local record) as the current version for all selected collisions. **The pulled versions are added to the version history for the records.**	Select **Use Local Version**

Configuring Team Development

Configure Team Development for your instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Configuration overview

1. Access rights for developers

   To enable parallel development on multiple non-production instances, administrators can set up the Team Development instance hierachy and grant access rights for developers.

2. Create an exclusion policy

   Application developers can create an exclusion policy to prevent pushes or pulls to particular instances in the team development hierarchy.

3. Define a remote instance

   Define other instances in the hierarchy as remote instances.

4. Enable a code review

   You can require a code review of all changes pushed to an instance.

5. Select the parent instance

   An instance can have multiple peer instances but only one parent instance.

6. Set up an instance hierarchy

   Set up an instance hierarchy that best supports your development life cycle.

Access rights for developers

To use Team Development, application developers must have a set of credentials for each instance in the Team Development hierarchy.

An instance's placement in the team development hierarchy determines the credentials it requires.

Credentials for Team Development access

Desired Access	Credential Requirements
Access to the Team Development application	A user with the admin role on the instance you are accessing
Right to register a remote instance	One of following: • A user with the admin role on the instance you are registering • A user with the teamdev_user role on the instance you are registering
Right to push changes to the parent instance from a development instance	One of following: • A user with the admin role on the parent instance • A user with the teamdev_user role on the parent instance
Right to compare to a registered remote instance	One of following:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Credentials for Team Development access (continued)

Desired Access	Credential Requirements
	• A user with the admin role on the registered development instance • A user with the teamdev_user role on the registered development instance
Access to the Code Review Requests module	One of following: • A user with the admin role on the parent instance • A user with the teamdev_code_reviewer role on the parent instance

Note: The teamdev_user role does not grant access to the Team Development application and is not intended for developers to work on local development instances. It is intended to grant developers non-admin access to remote instances such as the parent instance or a peer development instance.

Create an exclusion policy

Application developers can create an exclusion policy to prevent pushes or pulls to particular instances in the team development hierarchy.

Before you begin

Role required: none

Procedure

1. Navigate to **All** > **Team Development** > **Exclusion Policy**.
2. Click **New**.
3. Complete the Exclusion Policy form (see table).
4. Click **Submit**.

Exclusion policy form

Field	Description
Name	Enter a unique description of the policy.
Policy	Select when the policy applies. Options include: ○ Push only ○ Push and Pull ○ Pull only
Remote Instance	[Optional] Select a specific remote instance to ignore changes from during pull operations. Leaving this field blank ignores changes from all remote instances.
Table	Select which table to ignore changes for.
Conditions	Select any additional criteria a change must meet to be ignored other than the table name. This field is only visible when the *Policy* is Push only.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Define a remote instance

For each instance, define other instances in the hierarchy as remote instances.

Before you begin

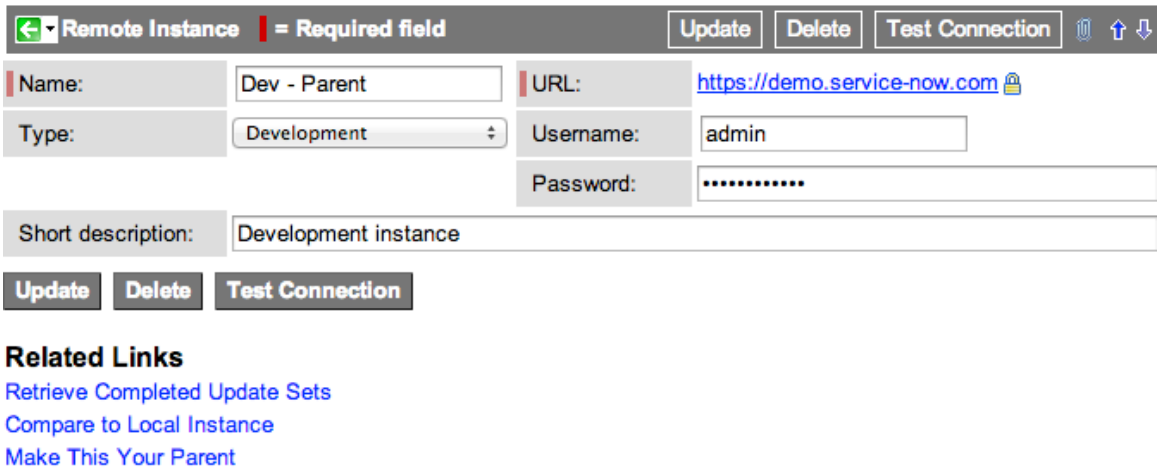
Role required: none

About this task

For example, to set up remote instances for Sub-Dev 1:

Procedure

1. If IP address access control is enabled, log in to the remote instance and add Sub-Dev 1 as an exception.
2. On Sub-Dev 1, navigate to **Team Development** > **Remote Instances**.
3. Click **New**.
4. Define the remote instance, such as Dev-Parent, by completing the form (see table).



5. Click **Submit**.
6. Repeat step 1 through step 5 for each instance in the hierarchy that this instance needs to push and pull with (for example, Sub-Dev 2 and Sub-Dev 3).

Remote instance form

Field	Description
Name	Enter a unique name describing the instance.
Type	Specify whether the remote instance is a development, test, or UAT instance.
Active	Specify whether the local instance communicates with the remote instance as a member of Team Development. Team Development operations such as comparing changes between instances or selecting a parent instance are only available for active remote instances.
URL	Specify the URL of the remote instance using the appropriate transfer protocol. Each remote instance record should have a unique URL. Creating duplicate records with the same URL can cause errors.
Username	Enter the user on the remote instance who authorizes Team Development operations on the instance. This user account must have an appropriate role on the remote instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Password	Enter the password of the authorizing user.
Short description	[Optional] Enter any other relevant information about the remote instance.

Enable a code review

You can require a code review of all changes pushed to an instance.

Before you begin
Role required: none

Procedure

1. Navigate to **All** > **Team Development** > **Properties**.
2. Select the **Yes** check box for *If this property is set to Yes, code review is required before pushing to this instance* (com.snc.teamdev.requires_codereview).
3. Click **Save**.

   Setting this property adds the Code Review Requests module to the application menu and requires all changes pushed to this instance to remain in the *Awaiting Code Review* stage until someone in the Team Development Code Reviewers group approves them.

Select the parent instance

An instance can have multiple peer instances but only one parent instance.

Before you begin
Role required: none

About this task
The parent instance is the only instance you can pull changes from and push changes to.

The parent instance must be on the same release family as the local instance. For example, a development instance on the Geneva release family must have a parent instance also on the Geneva release family. If you select a parent from a different release family, the Team Development dashboard displays an error message and prevents you from pulling changes and reconciling. If you select a parent from a different patch release, the dashboard displays a warning message but allows you to pull changes and reconcile.

Do not use Team Development with production or test instances.

- Do not use a test or production instance as the parent instance in Team Development.
- Do not make any instance the parent of a production instance.
- Production instances should never have a parent.

When you back out a change on a Team Development instance, it backs out the change all the way back down the chain, including undoing the work on the source instance. This behavior can cause major problems on test and production instances.

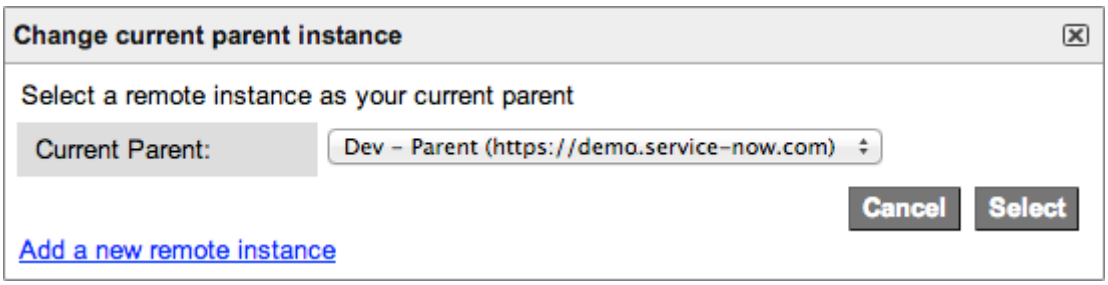
Procedure

1. Navigate to **All** > **Team Development** > **Team Dashboard**.
2. In the control panel, click the appropriate link:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Team dashboard control panel options

Option	Description
Use <instance name and URL>	Selects the most recently defined remote instance as the parent instance.
Select a different instance	Opens a dialog box where you can select another remote instance or define a new remote instance.
Register a new instance or List all remote instances	Opens the remote instance form or list, where you can define a new remote instance. These options are available when no remote instances are defined.



3. If you defined a new remote instance in step 2, repeat step 1 through step 2 and select the remote instance you defined.

   The system initiates a reconcile, which compares the local instance to the parent. It then generates the list of local changes and calculates the number of changes that are ready to pull from the parent. The reconcile also validates the instance versions.

4. Pull all changes from the parent instance if both instances are in the same release family.

   Note: The parent instance is saved in the `glide.apps.hub.current` system property.

Set up an instance hierarchy

Set up an instance hierarchy that best supports your development life cycle.

Before you begin

Role required: none

About this task

This example demonstrates how to set up an instance hierarchy where several peer sub-development instances have the same parent development instance, but a more complex configuration may be required to handle multiple project teams or other customer requirements.

Do not use Team Development with production or test instances.

- Do not use a test or production instance as the parent instance in Team Development.
- Do not make any instance the parent of a production instance.
- Production instances should never have a parent.

When you back out a change on a Team Development instance, it backs out the change all the way back down the chain, including undoing the work on the source instance. This behavior can cause major problems on test and production instances.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

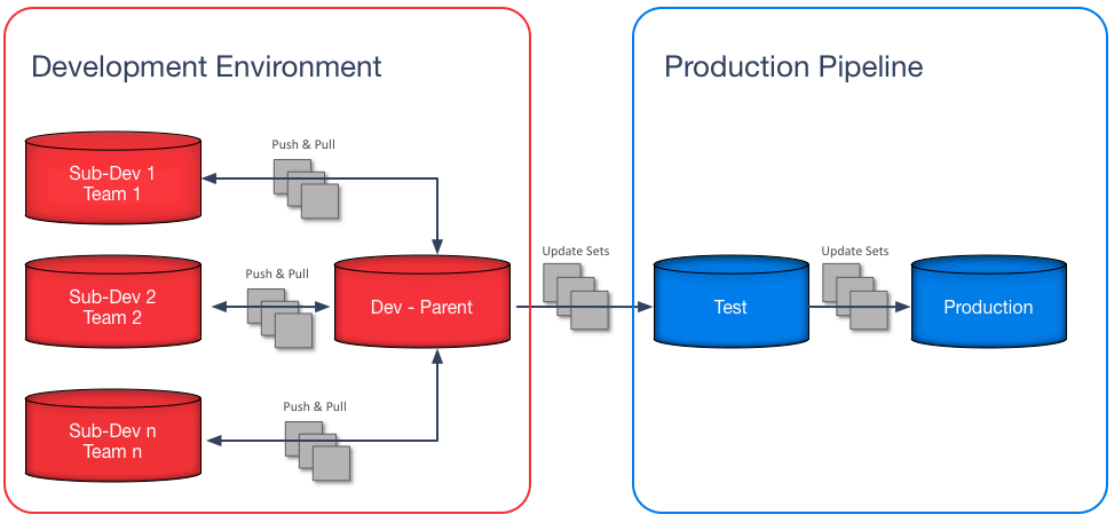
1. Provision a parent development instance on the same software version, such as Dublin, as the target instance, such as production.
2. Clone the production instance to the parent development instance.
   This step is a recommended best practice.
3. Provision sub-development instances on the same software version as the parent development instance.
4. **Optional:** Log in to the parent development instance and clone it to the sub-development instances.
5. On each sub-development instance:

   a. Define remote instance connections to other instances in the hierarchy that this instance needs to push and pull with.

   b. Select the parent instance.

   c. Pull all changes from the parent instance.

   d. Grant access rights to appropriate developers.



Administer Team Development

Administer Team Development within your instance to compare instances, push or pull record versions, or view the team development dashboard.

Overview of Team Development

Team Development provides the following features:

- Branching operations, including pushing and pulling record versions between instances.
- The ability to compare a development instance to other development instances.
- A central dashboard for all Team Development activities.

- Code reviews
- Code review notifications

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Code review workflow
- Exclusion policies
- Instance hierarchies
- Pulls and pushes
- Team Development process

Code reviews

Team Development administrators can require that pushes undergo code review before accepting pushes.

When code review is enabled, pushing a change to the parent instance triggers the code review workflow. By default, users with the teamdev_code_reviewer role receive notifications to review changes and can approve or reject changes. The Team Development Code Reviewers has the teamdev_code_reviewer role.

For each change, reviewers can see the following information.

- Which remote instance the pushed change comes from.
- Who pushed the change to the parent.
- What the change is called.
- When the change was created.
- Which versions the change includes.

Reviewers must approve or reject a push from the Team Development application.

While changes are being reviewed on the parent instance, a child instance cannot do the following activities involving the parent instance:

- Push changes to the parent instance.
- Pull changes from the parent instance.
- Reconcile changes with the parent instance.
- Change the parent instance to another instance.
- Delete the remote instance record for the parent instance.

Code review notifications

You must enable email notifications on the instance requiring code review for that instance to send code review notifications.

The Team Development Code Review workflow sends notifications to members of the Team Development Code Reviewers group when:

- A push requires code review.
- A user cancels a push.

If the user who pushed the changes has a user record with an email address on the instance where code review was required, the user receives a notification when the approval stage is set to *Complete* (approved) or *Code Changes Rejected*.

The code review notifications contain the following information:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Code review notification table

Notification name	Table	Contents
Code review update for developer	Push or Pull [sys_sync_history]	• The push name • The approval stage of the push (approved or rejected) • A link to the instance where the code review request was made
Notify code reviewer of canceled review	Push or Pull [sys_sync_history]	• The user who canceled review • The push that was canceled

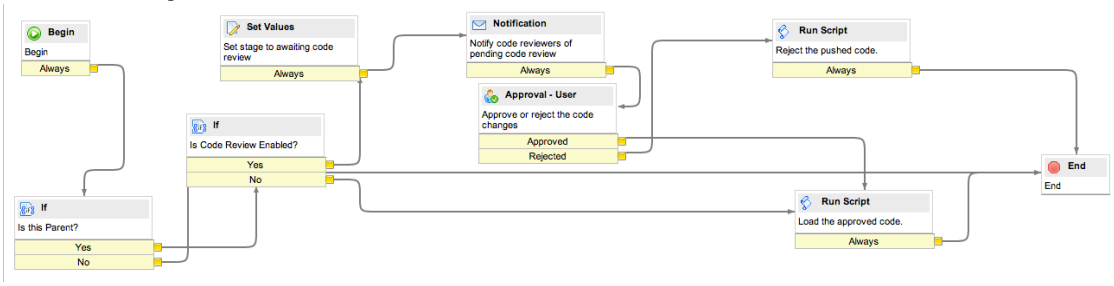
Code review workflow

The Team Development Code Review workflow manages how changes are pushed to the parent.

By default this workflow:

- Starts when changes are pushed to the parent instance.
- Verifies that the code review property is active on the parent instance.
- Sets the stage of changes requiring approval to *Awaiting Code Review*.
- Notifies the Team Development Code Reviewers group to review pushed changes, if configured.
- Loads approved changes or sets the stage to *Code Changes Rejected*.

Team Development code review workflow



Warning: Use caution when modifying this workflow, as the code review feature may not function properly.

Exclusion policies

You can exclude certain files from change tracking by creating an exclusion policy.

When a change matches an exclusion policy, the change does not generate records in the local changes list. The change still generates local version records and Update Set records as normal. See Creating an Exclusion Policy.

Note: The exclusion policy applies to changes identified during a reconciliation operation. If you create an exclusion policy after a reconciliation, Team Development still tracks the changes until the next reconciliation.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Instance hierarchies

Team Development allows you to set up a distributed version control system between multiple ServiceNow instances where each instance acts as a source repository, or branch.

Developers use separate instances to work on different features, applications, or product releases at the same time. With Team Development, developers can share code between these instances and resolve collisions throughout the development process.

Team Development allows you to establish hierarchical relationships between instances and provides a mechanism for transferring changes between instances that integrates with the Update Set process where necessary. In a Team Development instance hierarchy, each non-production instance has a parent instance. Instances that have the same parent instance are peer instances. The shared parent instance becomes the central hub, or repository, and all peer instances synchronize to it.

Pulls and pushes

Developers synchronize their instances to the parent instance by pulling and pushing versions of customized records and resolving collisions between versions on the parent instance and the development instance.

Developers can compare peer instances to one another and share code or resolve collisions before pushing versions to the parent instance.

Pulling from the parent retrieves versions of records that have customer updates. Pulling retrieves all versions that have not already been pulled onto the development instance, including historical versions, and you cannot choose which versions to pull. You must resolve any collisions before proceeding with further pulls or pushes.

Pushing to the parent adds only the current development version to the parent, not all the development versions. You can choose which changes to push to the parent. Pushing creates a local Update Set on the parent that is marked as complete. Pushed versions are also tracked as local changes on the parent. Therefore, you can promote changes through your development and test hierarchy by transferring the Update Set or by pushing the local changes.

Comparing reports the differences between two peer instances. You can choose which versions to pull from a peer instance.

The *Pushes and Pulls* related list on the team dashboard displays the user who created a change and the remote instance where the change was created.

Team Development process

The basic Team Development process sets up the instance hierarchy, grants developer access rights, manages the movement of development changes from development instances to test instances, and promotes applications to the production instance.

Before you begin

Role required: admin

Procedure

1. Set up the development instance hierarchy as described in Set up an instance hierarchy.

   a. Provision development instances on the same software version as the target instance. For example, use the software version that is running on your production instance.

   b. [Recommended] Clone the target to the development instances.

   c. For each instance, define the parent instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

   d. [Optional] For each instance, define the peer instances.
   e. For each instance, pull all changes from the parent instance.
2. For sub-development instances, grant access rights to appropriate developers.
3. Develop customizations on sub-development instances.

   Use the team dashboard to track development activities.

   - Pull versions from the parent instance, such as versions that were pushed from other sub-development instances. Reconcile any conflicts with the current local version, as necessary.
   - Track local changes. Queue changes that are ready to push to the parent development instance.
   - Compare versions on peer instances. Reconcile any conflicts.
4. When a feature is ready to promote to the parent development instance, push the current version of the customized records.
5. **Optional:** Have code reviewers approve or reject the pushed version.
6. Test and promote the feature into production according to your testing and release management process.

Related topics

Team Development

Access rights for developers

Push a version

Team Development roles

To use Team Development, developers must have admin access to their development instance.

To allow pushes to the parent instance, a remote instance connection must be defined with a user account that has admin access to the parent instance.

To limit developer access to the parent instance, see Granting Access Rights to Developers.

To use code review features, users must have the teamdev_code_reviewer role. See Code Review.

Versions

Version records track changes to a customized record over time so that administrators can compare or revert to specific versions later.

Administrators can also transfer versions between instances with Update Sets or team development.

Version record navigation

There are a variety of methods for viewing a list of versions for an object.

- For forms, right-click the header and select **Configure** > **Form Layout**.
- For lists, perform the appropriate action for the list version.
  - List v2: Right-click the header and select **Configure** > **List Layout**.
  - List v3: Open the list title menu and select **List Layout**.
- Click the **Show Versions** related link.

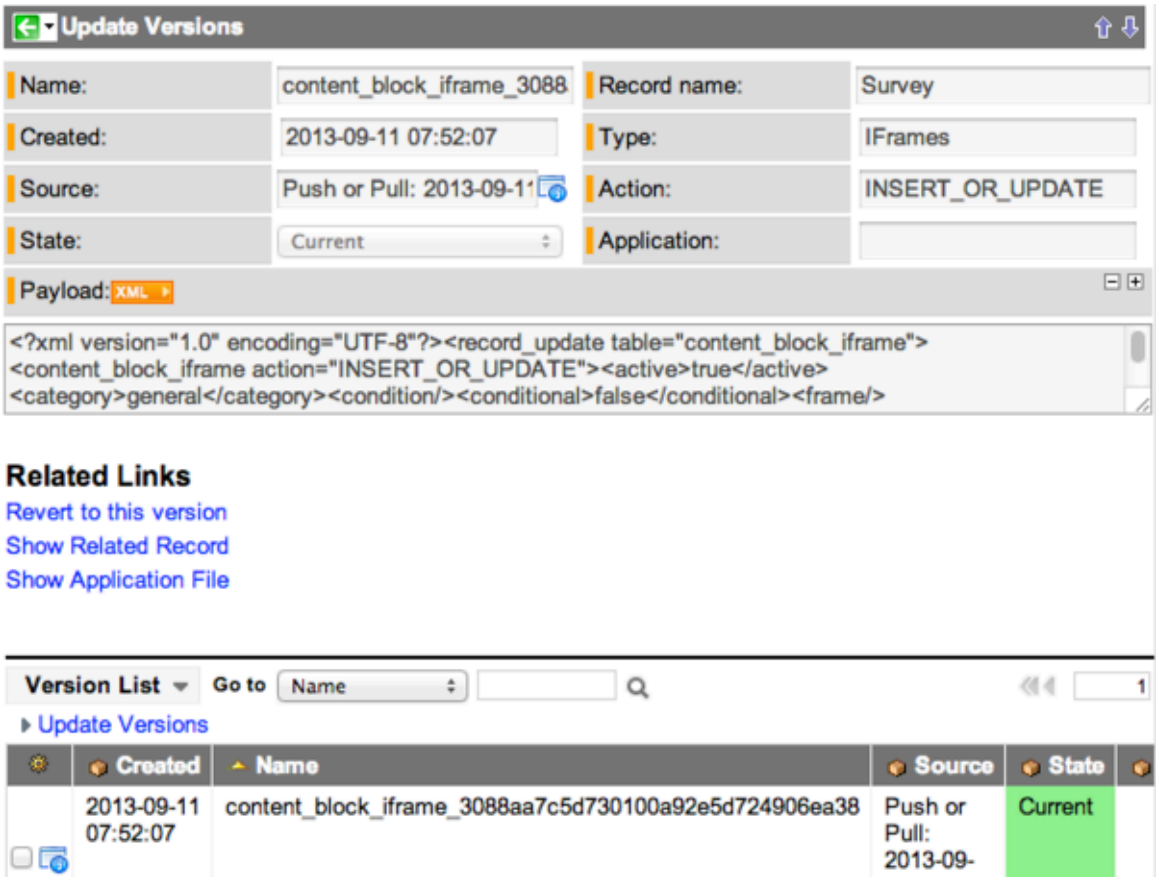
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- For tables that use the update_synch attribute, add the **Versions** related list to the form. This list is on several forms by default, including, business rules, UI actions, and client scripts.
- For any customizable object, right-click the form header and select **Show Application File**, then scroll down to the *Related Record Versions* related list.

You can navigate from a version record to:

- The customized object: Click the **Show Related Record** related link.
- The application file record for the object: Click the **Show Application File** related link.

Version record



Versions transferring

Administrators transfer version records between instances by moving customizations with Update Sets or the Team Development application.

- Update sets: committing an Update Set adds versions. For each update in the Update Set, the version that corresponds to the update is added on the local instance.
- Team Development:
  - Pulling retrieves from the parent instance all versions of customized records that have not already been pulled and then adds them on the local instance.
  - Pushing adds to the parent instance only the current local version, not all the local versions.
  - Loading changes from peer instances adds selected versions to the local instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Version records

The Update Versions [sys_update_version] table contains records that represent the state of a customizable object at a particular time.

A customizable record is any object that is tracked by Update Sets, such as business rules or script includes. A new version record is created automatically whenever a user changes a customizable record or changes the application file for the customizable record.

A record represents the version of a base system object as it was delivered in the most recent upgrade. Baseline versions are created only for objects that have been modified by a user, and they are updated each time the system is upgraded.

Update versions table

Field	Description
Name	A unique identifier for coalescing versions of the same customized record.
Record name	Name of the customized record.
Source	Indicator of how the version was added on the instance. • System Upgrade: from a software upgrade (the baseline version). • Update Set: from an update set that was created or committed on the instance. • Pull History: from a pull in Team Development.
State	Indicator of whether the version is or has ever been loaded on the instance. • Current: the version is currently loaded. • Previous: the version has previously been loaded on the instance. When a current version is replaced by a new version, it becomes a previous version. • History: the version was never loaded on the instance and was only inserted for historical purposes, such as when pulling versions from the parent in Team Development.
Application	The application for the customized record, if it is assigned to an application.
Payload	The data for this version of the customized record.
Additional fields on the list view	
Reverted from	A reference to the older version record, if this version was created by reverting to an older version.
Fields that can be added by configuring the form	
Instance Name	The name of the remote instance where the version was originally created.
Instance ID	The URL of the remote instance where the version was originally created.
Related lists on the form view	
Version List	All versions of the customized record that are available on the instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

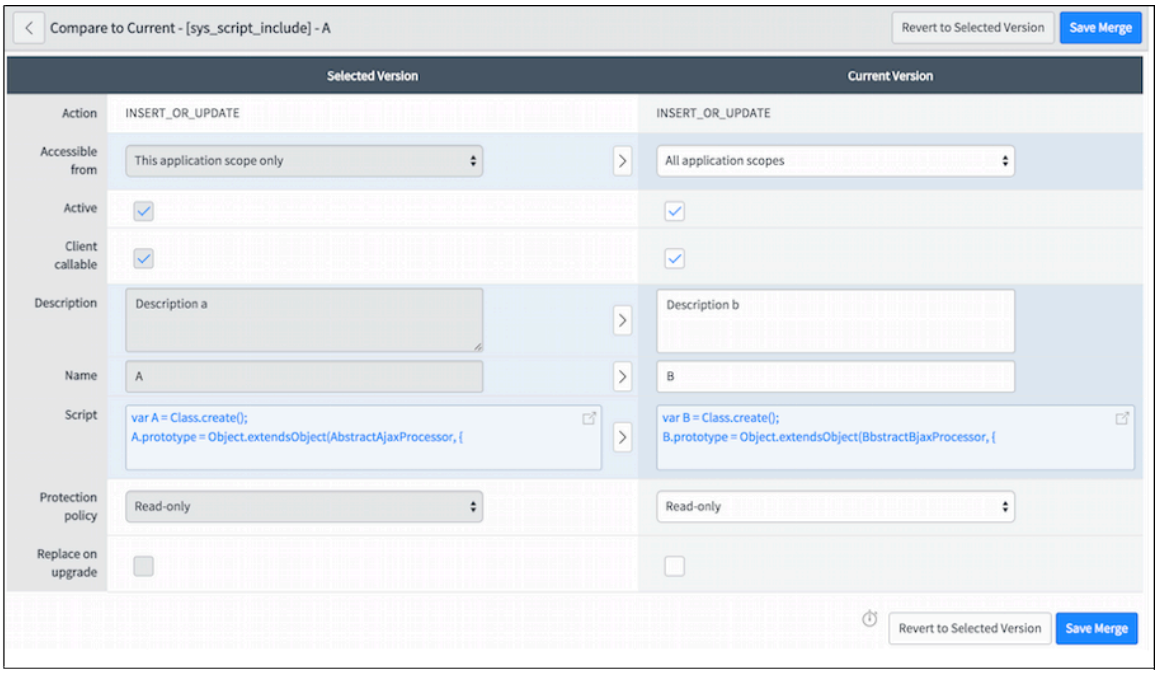
Related topics

Team Development

Merge tool

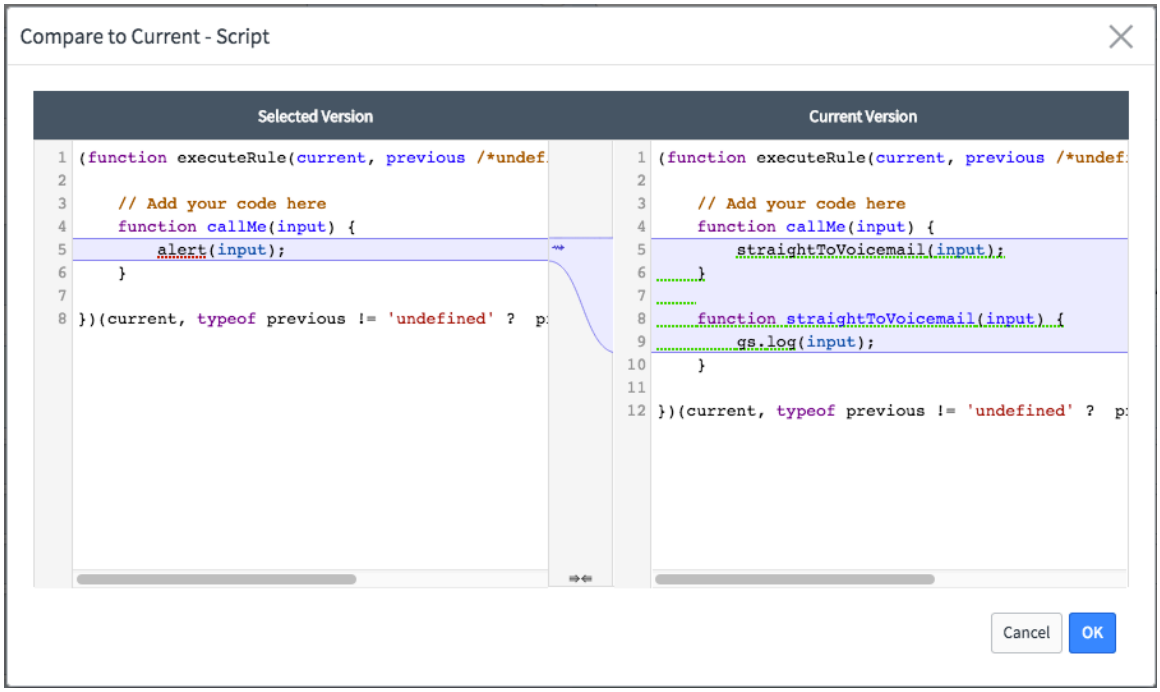
The Diff Merge tool enables administrative users to compare differences between two versions of a record.

Administrators can compare field-level changes between two versions, apply changes using Move Right field-level copy functionality and then merge results, or choose to revert to the non-current version. You can access the Diff Merge tool by comparing versions, resolving conflicts, or resolving collisions, during development or after upgrades.





© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Accessibility Functions

The platform includes accessibility features that support Web Content Accessibility Guidelines (WCAG) 2.0 level A and make the interface accessible to users with disabilities. These features improve the user experience when accessing platform functions with Using accessibility features.

In general, you can use the following set of standard keyboard navigation functions:

- Press **Tab** to navigate major groupings in a pre-defined sequence, including moving between standard interface controls (fields and lists) in a module, or between records within a tab.
- Press **Shift Tab** to move backwards in a pre-defined sequence.

Visually impaired users can navigate the Diff Merge tool. Screen readers can read all critical page content. All links and buttons can be reached when a section that is critical must be read. VoiceOver audible cues describe the content of the section that is necessary to read.

To enable accessibility functions, administrators should set these sys_properties:

1. Setting **glide.ui.javascript_editor** to false makes the following functions accessible:
   - Script fields (such as Script Include).
   - Side-by-side script comparison.
2. Setting sys_properties color settings enables high contrast visibility, which makes the left and right columns more accessible and easier to read by visually impaired users.
   - **mergetool.bg.left.highlight** - Left column cell color when values differ between versions.
   - **mergetool.bg.right.highlight** - Right column cell color when values differ between versions.
   - **mergetool.bg.left** - Left column cell color when version values are the same.
   - **mergetool.bg.right** - Right column cell color when version values are the same.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Related topics

- Compare to the current version
- Compare a pushed version to a local version
- Compare two versions of an article
- Resolve conflicts for an individual record
- Resolve a collision in Team Development
- Revert a change
- View customizations and compare with current version

Compare to the current version

You can compare a version to the current version for any customizable object that a user has modified, such as a form layout or business rule. You can also compare the local and current pulled version of an object in Team Development. Administrators can suppress versions for specific tables.

Before you begin

Role required: none

About this task

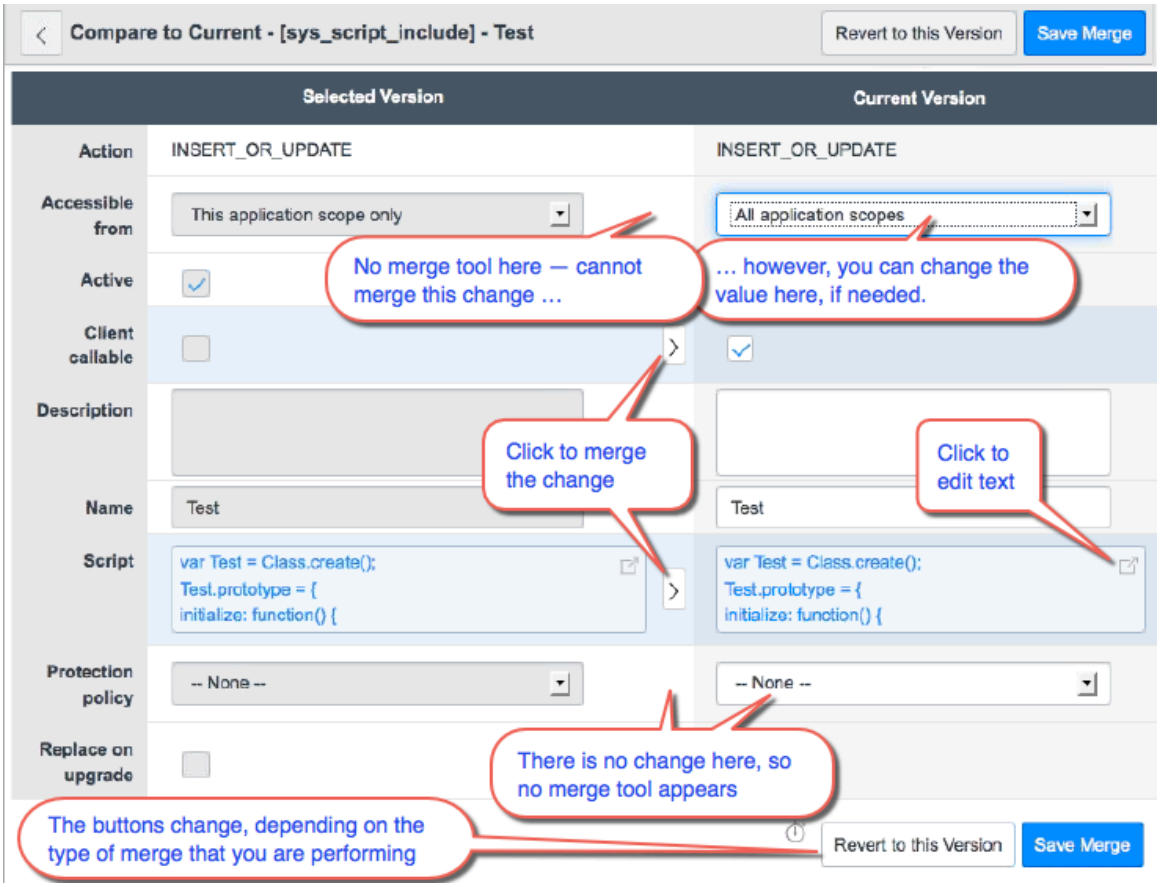
To compare a version to the current version of an object:

Procedure

1. Open the Compare to Current page using one of the methods given in the below table.

Option	Action
From a Versions list	Right-click the version and select **Compare to Current**.
From the Update Versions form	Click the **Compare to Current** related link.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



2. On the Compare to Current page, review the fields that differ.

Option	Action
To resolve the differences by choosing the previous version	○ Team Development: Click **Revert to Selected Version**. ○ For a version of an object: Click **Use Local Version** to maintain the local record as the current version. The pulled version is added to the version history for the record.
To resolve the differences by modifying the current version and saving the merged changes **Note: Some types of record do not support this method. See Limitations on updating records for more information.**	You can either update the setting in the current record or move a setting from the selected version to the current version. To move a change, click the **>** (Move Right) button for the field in the diff/merge tool. To work with scripts and text fields, click in the field and modify the text as needed. When the records meet your needs, click:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Action
	○ Team Development: Click **Save Merge** to save the changes to the current version. ○ Team Development: Click **Use Pulled Version** or **Use Local Version** option to accept or reject all changes, as appropriate. ○ On the upgrade history Compare to Current form, the only option is **Revert to Base System**

Related topics

Merge tool

Compare to the current version

Compare a pushed version to a local version

Compare two versions of an article

Resolve conflicts for an individual record

Resolve a collision in Team Development

Revert a change

View customizations and compare with current version

Revert a change

You can undo changes to a customized record by reverting to an older version.

Before you begin

Role required: none

Procedure

1. View a list of version records for an object.
2. **Optional:** Compare the current version to the older version to ensure that you are reverting the desired changes.
3. Right-click the older version and select **Revert to this version**.
   A confirmation dialog box appears.

   If reverting to this version results in data loss due to a database schema change, a warning message appears in the dialog box.
4. Click **OK** to confirm the action.
   - The current version is marked as a previous version.
   - A new version record is added that duplicates the version that you selected in the preceding step. This new version is marked as the current version.

Note: You can revert to the most recent baseline version. You cannot revert to an older baseline version.

Related topics

Merge tool

Compare to the current version

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Compare two versions of an article
- Compare a pushed version to a local version
- Resolve a collision in Team Development
- Resolve conflicts for an individual record
- View customizations and compare with current version

Suppress versions

Administrators can configure a table so that it does not track customizations in the Versions `[sys_update_version]` table.

Before you begin

Role required: none.

About this task

Warning: If you suppress versions for tables, Team Development may work incorrectly, and you may be unable to compare and revert versions of records on the tables.

Procedure

1. Navigate to **sys_properties.list**.
2. Create a new property:
   - Name: glide.update.suppress_update_version
   - Type: string
   - Value: a comma-separated list of tables. The default value is sys_user,sys_import_set_row.

Versions and local changes

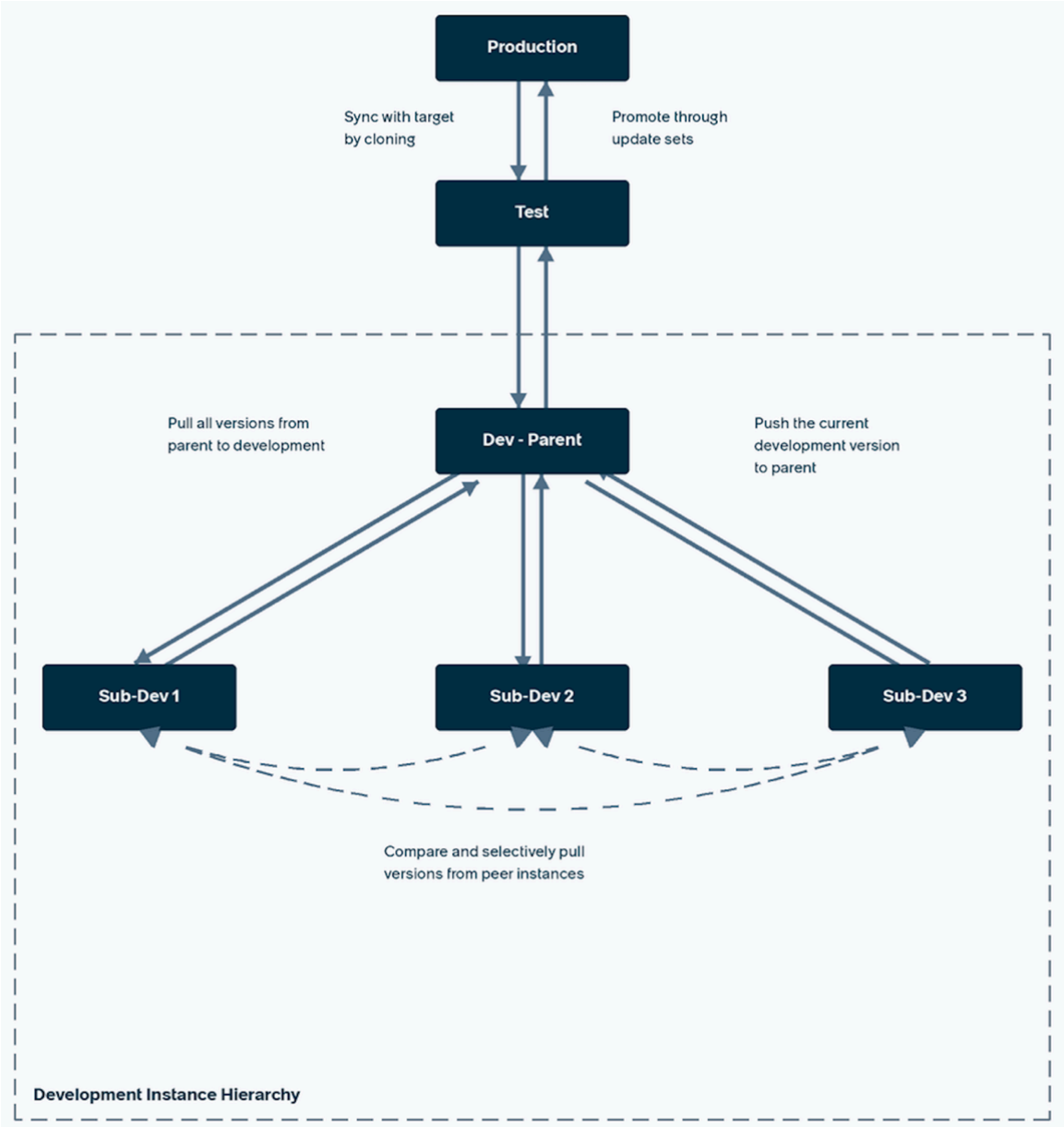
Version records track changes to a customizable record over time so that you can compare or revert to a specific version later.

A version record is created every time a developer changes a customizable record, so a single record can have multiple version records associated with it. A local change record is created or updated to reference the current version every time a developer modifies a customizable record, so a single record can have only one local change record associated with it.

Local change records track which customized records have changes on the development instance that are not on the parent instance so that you can collect changes in preparation for a push.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Team Development concepts

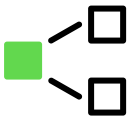
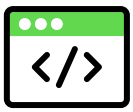


Developers can back out a local change to restore a previous version of a customizable record. The back out action sets the local customizable record to the last revision identified by a reconciliation action.

Developing your application

Build a custom application to meet the business needs of your organization. Choose a builder for the type of user experience or workflow that you want to create.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Agentic development and Build Agent  Expedite app development and enhance custom applications with the AI tools available on the ServiceNow AI Platform.	**Build no-code applications with Creator Studio**  Create and manage custom applications in a simple, no-code environment.	**Build low-code applications with App Engine Studio**  Accelerate innovation with more creators and less complexity. Empower developers of all skill levels to build applications at scale.
Build applications with ServiceNow Studio  Empower developers with a modern, unified environment for building on the ServiceNow AI Platform using integrated builders and tools.	**Build pro-code applications**  Extend or maintain existing applications and user experiences. Connect applications and data across the enterprise.	**Builder library**  Choose a builder tool to quickly develop an application.

Which app builder should I use?

Use the following table to choose the app building experience that matches your role and technical background.

App builders

Tool	Users	Features
Build Agent: Build an app using natural language prompts.	Developers of all experience levels (from low-code to Platform and source code)	Build full-stack applications via a conversational interface. You can use Build Agent in ServiceNow Studio and the ServiceNow IDE to generate a complete ServiceNow scoped or global application. For more information, see Exploring Build Agent.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

App builders (continued)

Tool	Users	Features
ServiceNow Studio: Build and deliver apps in a unified environment.	Citizen developers, Platform developers	Build apps in a unified development environment. ServiceNow Studio provides streamlined navigation, integrated low-code tools, and built-in tracking and packaging so you can develop and ship apps faster. Use Build Agent in ServiceNow Studio to create and update apps with a conversational interaction. For more information, see Exploring ServiceNow Studio.

Agentic development on the ServiceNow AI Platform

Agentic development of full-stack applications with Build Agent

Agentically develop applications on the ServiceNow AI Platform with Build Agent, an autonomous AI agent available with ServiceNow Otto for Creator. Describe what you need through back and forth conversations with Build Agent, and Build Agent automatically builds the application, including audit trails, security, and compliance.

ServiceNow Otto for Creator

Use the AI skills, agents, and agentic workflows available with ServiceNow Otto for Creator to generate and enhance applications.

Create and use AI assets in apps at runtime with ServiceNow Otto for App Engine

Enhance App Engine applications with AI skills, agents, and agentic workflows that users can use at runtime with ServiceNow Otto for App Engine.

Generate code with AI-powered code generation

Generate scripts from text, code, or a combination of both with AI-powered code generation.

Build applications with ServiceNow Studio

Build apps smarter and deliver them faster with the new ServiceNow Studio. ServiceNow Studio empowers platform developers with a modern, unified environment for building on the ServiceNow AI Platform. ServiceNow Studio features streamlined navigation to applications and metadata, integrated low-code tools, efficient tracking and packaging of development work that accelerates development processes and enhances productivity.

ServiceNow Studio quick start

Quickly familiarize yourself with new features in ServiceNow Studio.

Build applications

Admins can easily build custom or global apps using a guided experience and integrated builders.

Deploy applications

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Deploy your applications seamlessly using ReleaseOps, update sets, pipelines, or the Application Repository.

ServiceNow Studio Navigator panel taxonomy

Learn more about each type of file you can add to applications in ServiceNow Studio.

Develop in sandbox environments

Use Developer Sandboxes to create isolated development environments where users can develop on the ServiceNow AI Platform in parallel. Sandboxes help enhance Git-based workflows of the software development life cycle by reducing conflicts and enabling faster time to production.

Builder library

Each builder fulfills a specific need or produces a specific type of data, such as Decision Builder, UI Builder, and Workspace Builder. For a complete list of builders, see Builder Library.

App Engine products and offerings

App Engine is a ServiceNow AI Platform® suite of products to help boost productivity by automating workflows.

For a full list of App Engine products, see the ServiceNow product site.

Creator Studio

Creator Studio is a no-code application builder on the ServiceNow AI Platform that lets non-developers create request-fulfillment apps without writing code. Use Creator Studio to build apps that manage employee requests, approvals, and fulfillments, such as equipment requests or travel approvals, directly on the ServiceNow AI Platform.

https://player.vimeo.com/video/1024834247?badge=0&autopause=0&player_id=0&app_id=58479

Creator Studio specializes in Service Desk request-fulfillment applications, where one person submits a request and another person fulfills it. Service Desk apps enable users to submit requests, report issues, and access support related to services within your company. Use ServiceNow AI Platform to build apps that handle common workplace requests, so your team can submit, review, approve, and fulfill them in one place.

ServiceNow AI Platform supports apps for requests, such as:

- Ordering equipment, such as keyboards or monitors
- Approving travel or purchase requests
- Granting permissions, such as vacation approvals
- Distributing recognition gifts or awards

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Get started

Explore



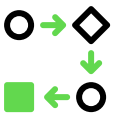
Learn about Creator Studio concepts and features.

Configure



Install and configure Creator Studio tools and access.

Administer



Administer Creator Studio requests and deployments.

Build



Build apps with forms and automation in Creator Studio.

Reference



Get details on Creator Studio components, such as fields, tables, and properties.

Which app builder should I use?

Use the following table to choose the app building experience that matches your role and technical background.

App builders

Tool	Users	Features
Build Agent: Build an app using natural language prompts.	Developers of all experience levels (from low-code to Platform and source code)	Build full-stack applications via a conversational interface. You can use Build Agent in ServiceNow Studio and the ServiceNow IDE to generate a complete ServiceNow scoped or global application. For more information, see Exploring Build Agent.
ServiceNow Studio: Build and deliver apps in a unified environment.	Citizen developers, Platform developers	Build apps in a unified development environment. ServiceNow Studio provides streamlined navigation, integrated low-code tools, and built-in tracking and packaging so you can develop

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

App builders (continued)

Tool	Users	Features
		and ship apps faster. Use Build Agent in ServiceNow Studio to create and update apps with a conversational interaction. For more information, see Exploring ServiceNow Studio.

What apps are in ServiceNow

A ServiceNow app is a package that performs a specific task for a specified group of users. Think of an app as a container with a set of rules around who can access and edit it. For example, ServiceNow apps can include an API, a table, a workspace, a form, a flow, or any combination of those things.

A Creator Studio app uses forms and automation to enable users to request something and move the request through the approval process. Creator Studio also enables you to the see the fulfiller's view of the Request App Workspace for the app.

What you can do after you build an app in Creator Studio

After you build an app in Creator Studio, you can open the app in any other ServiceNow builder so that you can add more advanced functionality without rebuilding from scratch.

For example:

- Open the app in Workflow Studio to add complex automations.
- Open the app in ServiceNow Studio to add different interfaces. Select **Open in ServiceNow Studio** from the app's tile on the Creator Studio home page to open the app directly.

Give Creator Studio a try

Ready to give Creator Studio a try? You can test it out using your own Personal Development Instance (PDI), which requires you signing in to the Developer Site. Find out more on PDIs in the Personal developer instance guide.

Select this button to try Creator Studio on a PDI

Try Creator Studio today

Try out Creator Studio now on a PDI! Login required.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Exploring Creator Studio

Creator Studio makes creating apps easier by dividing their creation into simple steps. In this section, we'll explain each one.

All the steps involve requests. So, we'll start by explaining how Creator Studio stores requests.

Summary: After reading this section, you'll understand:

- How requests are stored
- How people enter requests
- How your app automatically does things

Let's talk about storing requests

When people make a request, we have to store it somewhere so someone can review it later.

Tables hold requests

Creator Studio stores requests in a table. Each row in that table is a request. Here's a simplified version of a table with three requests.

Use cases:

- My monitor stopped working. I need a new one.
- I'm requesting a new mouse so I can have one at home and in the office.
- My laptop is over four years old. I want to request a replacement.

People enter requests using forms

How do people enter requests that eventually get stored in the app's table? They use your app to fill out a form. The questions on the form contain the information your fulfiller needs.

In our app's table, questions are stored as part of the record, which is composed of all the information on the form.

Request	Name	Email	Request date
I need a new monitor.	Sushma Singh	SushmaSingh@example.com	04-05-2024
I need a new laptop.	Peter Smith	PeterSmith@example.com	05-01-2024

In this example, the form has four fields:

- Request
- Requester's name
- Requester's email address
- Request date

These fields contain all the info the reviewer needs to accept or reject the request. You'll have to figure out all the information your reviewer needs to make a decision. Later, we'll show you how to create a form (using AI or on your own) in the section Working with forms in Creator Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Key terms:

Requester

Someone requesting something, like a piece of equipment or permission to do something.

Fulfiller

Someone who works on requests. Fulfillers may also approve or deny requests, depending on any approval automation for the app.

Okay, now you understand that requesters fill out forms to create requests and Creator Studio stores those requests in the app's table.

Try Creator Studio on a PDI

Want to play with Creator Studio on your Personal Development Instance (PDI)? It comes automatically installed from the Application Manager on your PDI. For more information, see Personal developer instance guide.

Example apps you can build in Creator Studio

Creator Studio enables you to build apps where people can make requests.

Service desk apps

You can use Creator Studio to build service desk apps, which support departmental processes that occur on a routine basis. For more information on service desks, see Service desks and Creator Studio.

List of example apps

The following are some examples of service desk apps that you can build in Creator Studio. These examples support ad-hoc requests that fall outside the scope of standard ServiceNow offerings, filling in gaps in non-routine business processes:

- Requesting marketing collateral
- Travel approval and booking apps
- Asset borrowing and returns, for example borrowing company art for an office
- New hire gift box
- Employee recognition awards and approval
- Help with using a product
- Request a head shot
- Backup coverage app where people can volunteer to cover for people on leave
- One-time funding requests, such as a retirement party
- Birthday party planning
- Internal skill sharing, where a department creates an app of skills they need for a project where people can sign up

Use App Engine instead of customization

App Engine development tools, such as Creator Studio, offer an excellent alternative to customizing existing applications on the ServiceNow AI Platform.

When your company needs to add new functionality to the ServiceNow AI Platform, you can customize existing applications, such as IT Service Management (ITSM), or create a new

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

application using App Engine developer products, such as App Engine Studio, Creator Studio, or ServiceNow Studio. A simple guideline for which path to choose is:

- If the customization extends the intended purpose of the application, it works better to customize. For example, you can add IT functionality to ITSM.
- If the customization doesn't extend the intended purpose of the application, it works better to create a new application using App Engine developer products. For example, do not repurpose the ITSM workflow to add a travel request workflow.

Examples of when to use App Engine

ServiceNow products work best when they're used as they were intended. If you find yourself heavily customizing an application to repurpose it, a better plan is to create a new application using App Engine developer products.

The following scenarios demonstrate where creating a new application works better than heavily customizing an existing ServiceNow application:

- Your company has a business process that augments existing product functionality but doesn't follow the same workflow exactly.
- You have a novel use case for an app that doesn't align with any product workflow.
- You have a use case that could be built by heavily customizing an out-of-the-box application, but it doesn't align with what the existing application was intended to do.

Let's dive deeper into the last use case.

Issues with repurposing existing products

ServiceNow applications come with roles, processes, and flows that are specially tailored to their use case. For example, ITSM apps help with IT users, IT issues, IT reports, and IT cases.

You might have an idea for an app that's similar to but doesn't exactly align with ITSM. Because ITSM gives you a starting point, you might be tempted to customize ITSM to add the new functionality. For example, ITSM tracks IT issues, and a travel app you want to create might track travel requests. While the workflows sound similar, in fact, they use very different data, different user interfaces, and the details of each workflow vary greatly. Rather than heavily customize ITSM to repurpose it, a better plan is to use App Engine developer products for the following reasons:

- Combining two workflows creates conflicts.
- Customizing applications has implications.

Combining two workflows creates conflicts

In the ITSM example, the repurposing of ITSM to include a travel workflow uses different data, different tables, different roles, and different workflows than ITSM. As ITSM, ITSM customizations, and the travel workflow grow over time:

- Their features will continue to diverge.
- Adding new functionality or fixing problems in one workflow might adversely impact the other.
- The performance of ITSM may suffer.
- The code base will grow and the two purposes of ITSM will make troubleshooting more difficult.
- Quality engineers will require two different testing frameworks.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

All these issues may cause unnecessary complications, poorer performance, upgrade delays, and software problems.

Customizing applications has implications

The ServiceNow AI Platform is built to embrace customization and configuration. The ServiceNow AI Platform is flexible enough to fit your company's business needs. How you customize ServiceNow applications, however, can have significant impacts on ServiceNow support, upgrading to future ServiceNow AI Platform versions, and the functionality of the platform.

Let's start by differentiating customization and configuration:

- Customization is any change made to the code that is part of the baseline installation of a ServiceNow instance. You use code to customize applications.
- Configuration is any change you make to the behavior of a product that does not touch the code in the baseline installation of a ServiceNow instance. You can use system properties, ServiceNow products, or code to configure an application.

The following are some of the implications that result from customizing applications:

- If you add code to an application, you own it whether or not it modifies the code in the baseline installation on a ServiceNow instance.
- The platform marks all customizations and skips them when you update to a new version of the platform. That means you are responsible for manually updating the customizations. This can have a significant impact on the time and resources required to update to new platform versions.
- The ServiceNow AI Platform uses a framework that supports applications in how they process tasks, how forms are rendered in multiple browsers, and the overall user experience. Introducing customizations can have unintended consequences on this framework.
- You own the burden of testing custom code and determining if it impacts platform functionality.
- ServiceNow Customer Support cannot troubleshoot custom code or issues caused by custom code.

Customization is one of the key features of the ServiceNow AI Platform. However, over-customizing an application to repurpose it is likely to generate technical debt, lengthen your upgrade cycle, and complicate future platform upgrades because the custom code may not easily migrate to new platform versions.

Conclusion

Customization and configuration are hallmarks of the ServiceNow AI Platform that enable your company to customize workflows to fit its specific needs. Proceed with these tasks in the following order:

1. Configure ServiceNow applications as much as you can before customizing them.
2. Customize an application only when it extends the intent of the application.
3. Use App Engine developer products, such as App Engine Studio, Creator Studio, and ServiceNow Studio, to create applications rather than customizing an application to create functionality that doesn't align with its original purpose.

Customization vs. configuration with Creator Studio

There are important differences between customizing and configuring ServiceNow applications. The ServiceNow platform is built to embrace customization and configuration but how you do so can have significant impacts on ServiceNow support, upgrading to future ServiceNow platform versions, and the functionality of the ServiceNow platform.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The general rules around customization are:

- Customize an application only if it extends the original intent of the application. For example, add IT functionality to ITSM but don't add a travel workflow. Instead of over-customizing an application, create a new application using App Engine products, such as Creator Studio or ServiceNow Studio.
- Configure as much as you can before customizing an application.
- If you add code or make other modifications to out-of-the-box functionality, you own them.

What is configuration

Configuration is the process of using ServiceNow built-in tools and features to modify an application's behavior without making changes to flows or the code that is part of the baseline installation on a ServiceNow instance.

Configuration can take the form of using ServiceNow built-in tools to add tables and more, setting instance-wide parameters, as well as using code to extend an application's functionality to meet business needs as long as the code does not modify the baseline code installation. The entire platform is designed for you to add configuration code.

If you add code, such as workflow scripts, you own it, even if it doesn't alter the baseline code installation. That includes owning the impact it has on the entire ServiceNow platform. Issues that arise from added code are beyond the scope of ServiceNow support to debug.

Reverting a configuration should not require any change to the baseline code.

Configuration examples include:

- Forms: Configure tables, fields, data types, default values, and field dependencies to configure the data you capture and display.
- UI elements: Modify layouts, add related lists, add buttons, and change field names.
- Service Catalog: Configure portals where your customers can request catalog items such as service and product offerings.
- ACLs: Restrict unauthorized users from accessing forms and data.
- System property values: Modify the application's experience for all users.

What is customization

Customization is any change made to the flows or the code that is part of the baseline installation on a ServiceNow instance. You use ServiceNow products or code to customize applications.

If you add code, you own it, even if it doesn't alter the baseline installation. That includes owning the impact it has on the entire ServiceNow platform.

Customization examples include:

- Scripting: Customize ServiceNow through scripting using JavaScript. This includes creating client scripts, server-side scripts, and business rules with intricate logic that modify the baseline code.
- Custom tables: Develop custom tables to accommodate specialized data that doesn't fit within standard tables.
- Integration: Customize integration with external systems, such as APIs and web services, for seamless data exchange.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Widgets and portals: Create custom widgets and portals to provide unique features and user experiences.
- Workflows: Create and modify workflows using Workflow Studio. Create and manage playbooks, flows, actions, decision tables, and integrations from one design environment to automate tasks. Upgrading to a new version of a flow requires reapplying your customizations.

Tools for customization and configuration

ServiceNow offers many tools and features, such as business rules, to modify the out-of-the-box behavior of ServiceNow applications. Whether they customize or configure an application depends on how they're used. Using these tools to modify the installed code base constitutes customization. Using these tools to add code that does not modify the flows, or the installed code base constitutes configuration. In both cases, you own the code you add as well as the impact it has on the ServiceNow platform.

ServiceNow tools include:

- UI Policies: Dynamically modify the visibility of fields and attributes on a form according to user inputs.
- Business rules: Automatically trigger actions based on specified conditions.
- UI Actions: Extend and customize forms and lists by adding buttons, context menu items, or other UI elements that perform specific actions when clicked.
- Client-side scripts: Scripts that execute within the user's browser when certain actions occur on a form or a UI page.
- Server-side scripts: Scripts that execute on the ServiceNow server or database, for example, to update record fields when a database query runs.

What is personalization

Personalization is when users use out-of-the-box application tools to modify an application's look and feel only for themselves. Admins can change the look and feel for all users and that is considered configuration. Personalization examples include a user choosing to use dark mode or choosing which table columns to display.

Personalization does not change the baseline code installation on a ServiceNow instance. So, personalization does not impact customer support or interfere with upgrades to new ServiceNow versions.

Ramifications of customizing ServiceNow products

The ServiceNow platform is extremely flexible and built to embrace customization and configuration to fulfill a wide range of business requirements. How you customize ServiceNow applications, however, can have significant impacts on ServiceNow support, upgrading to future ServiceNow platform versions, and the functionality of the platform. Instead of customizing ServiceNow applications, consider using App Engine development products, such as Creator Studio and ServiceNow Studio to create new applications.

The ServiceNow platform uses a framework that supports applications in how they process tasks, how forms are rendered in multiple browsers, and the overall user experience. ServiceNow relies on the framework's integrity to develop and provide support in a consistent manner. Customizations can impair this framework, change platform functionality, and impair workflows and upgradeability.

Customizations trigger the platform to create sys_update_xml records, which are stored in the Customer Update table. The platform marks all customizations and skips the customized records

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

when you update to a new version of the ServiceNow platform. That means you are responsible for manually updating the customizations. This can have a significant impact on the time and resources required to update to new platform versions.

Note: The Customer Update table also contains modifications or additions to configuration metadata, for example, creating a new catalog item or a new workflow.

For more information, see the Customer Updates table. Note that the complexity of maintaining customizations dramatically increases as the number of customizations increases.

Customizing the installed code base can be costly, generate technical debt, lengthen your upgrade cycle, and complicate future platform upgrades because the custom code may not easily migrate to new platform versions. Custom code can change the standard functionality of the ServiceNow platform in unintended ways. Evaluate demands for customization carefully and only resort to customization where there is clear business value and no alternative. Wherever possible, avoid customization by using configuration instead.

If you customize a product:

- You are responsible for maintaining the customization going forward.
- Customer Service and Support will not support problems caused by custom code. If it is the cause of the problems, the support team will likely advise you to revert to the out-of-the-box code.

What is the Customer Service and Support stance on supporting customization

The ServiceNow Customer Service and Support stance on customization is if you add code, you own it and its consequences. Why? Customer Support is not privy to your custom business logic, doesn't know what the expected behavior should be, is unable to reproduce the issue on an out-of-the-box instance, and customer support engineers are not certified implementation specialists, so they are not certified to review customized code logic.

Alternatives to customization

If you have requirements and ideas for enhancements, instead of customizing the installed code base, you can:

- Use configuration instead of customization.
- Submit an Enhancement Request to the ServiceNow development team. Each request is evaluated and, if approved, will be incorporated into a future release.
- Create an app using App Engine developer products to handle desired functionality.

When to use App Engine developer products instead of customizing

When your company needs to add new functionality to the ServiceNow platform, you can customize existing applications, such as ITSM, or create a new application using App Engine developer products, such as Creator Studio or ServiceNow Studio. A simple guideline for which path to choose is:

- If the customization extends the intended purpose of the application, it works better to customize. For example, you can add IT functionality to ITSM.
- If the customization doesn't extend the intended purpose of the application, it works better to create a new application using App Engine developer products. For example, do not repurpose the ITSM workflow to create a travel request workflow.

For example, ITSM is designed to handle IT issues. To customize it to handle travel requests goes beyond the original intention of ITSM. Because IT and travel requests have different workflows,

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

it's better to create a travel request app using App Engine developer tools, such as Creator Studio and ServiceNow Studio, instead of customizing ITSM.

For more information, see Creator Studio.

Examples of when to use App Engine developer products

ServiceNow products work best when they're used as they were intended. If you find yourself heavily customizing an application to repurpose it, a better plan is to create a new application using App Engine developer products.

The following scenarios demonstrate where creating a new application works better than heavily customizing an existing ServiceNow application:

- You have a novel use case for an app that doesn't align with any product workflow.
- You have a use case that could be built by heavily customizing an out-of-the-box application, but it doesn't align with what the existing application was intended to do.
- Your company has a user group or business process that should be separate from the OOTB product workflow.

Guidelines for customizing ServiceNow products

If you must make a customization, consider the following suggestions:

- Maximize configuration options first.
- Avoid copying objects. Instead, update objects in place wherever possible, except for Service Portal widgets and other items designated to be reused.
- Default to "add before edit." This means that you should, for example, add fields to forms rather than change the type of an existing field. Avoid using the same names as out-of-the-box objects, methods, or classes when adding them.
- Minimize the number of fields you add to a form. The more fields you have on a form, the longer it may take to load.
- Export original records as backups before customizing them. Keep track of their out-of-the-box sys_id's in case you must restore them in the future.
- Use scoped applications as your default for any new custom development.
- Document all customizations. Add comments explaining why you customized (including business justification). Review all comments before upgrading to determine if you can revert to out-of-the-box code.
- Create tests for all customizations. Write Automated Test Framework (ATF) tests for all customizations where possible.
- Use HealthScan regularly to identify unnecessary customizations.
- Customizations should be made to baseline objects where necessary so that conflict resolution and decision-making can be appropriately recorded in the updates. Hidden customizations may cause administrators to overlook updates in future assessments in case reverts or merges are necessary.
- Test your customization for all use cases. Include performance testing and the introduction of unintended consequences.
- Administrators are responsible for verifying that their customizations work after a ServiceNow platform upgrade, and for keeping track of what customizations are made.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Handling customizations when you upgrade

Customizations trigger the platform to create sys_update_xml records, which are stored in the Customer Updates table. These records are not updated during platform version upgrades. ServiceNow marks them as skipped records in the ServiceNow Upgrade Monitor. To make sure they're successfully ported to the upgraded instance, you must manually process the skipped changes. For more information, see the Customer Updates table.

Assuming you've documented all your customizations—including the business justification—take your documented inventory and compare it with the skipped records identified in the Upgrade Monitor. After filtering out low-risk changes that have resulted in skipped records (for example, field labels or form layouts), you'll need to decide whether to:

- Retain each customization
- Revert to out-of-the-box
- Merge your customization with the base system to resolve conflicts

Creating your first app with Creator Studio

Creator Studio helps you create your app by dividing it into smaller parts. Each does something special, and you work on them sequentially.

Summary: After reading this section, you'll understand:

- The general workflow for building an app
- What goes into the idea for an app
- Who to ask for help

Let's give you a very high-level overview of what you will do in each part.

1. **Let's get you started.** Create your app in a few simple steps before you can start building it out. Find out more in Create an app in Creator Studio.
2. **Or ask to work on someone else's app!** If you see an app that you want to work on but don't have access to, you can request to get access to it. Or, you can ask your admin to create an app for you that you can then work on. Find out how in Ask to work on an app in Creator Studio or Ask an admin to create an app for you in Creator Studio.
3. **Add a form to collect information.** A form is where you gather information by asking questions, but first you have to add the form to the app. See Add a form to an app in Creator Studio for more information.
4. **Ask for the info you need.** Customize the default catalog template to ask for the specific information your fulfiller needs. How do you know what customizations to make? Ask yourself what info the fulfiller needs to fulfill a request. We'll show you how to add, subtract, and modify the form fields in Customize your form for an app in Creator Studio.
5. **Make your forms available.** Once your form is ready, publish it to make it accessible to requesters. The process is straightforward, and we'll walk you through it in Publish a form for your app in Creator Studio.
6. **Specify the automatic actions that your forms trigger.** You'll want your app to take some action when a requester submits or revises a form. For instance, a new request might trigger approval from a manager. Another action could be emailing the requester to confirm their request submission and provide them with a tracking number. Discover more automation possibilities in Working with automation in Creator Studio.
7. **Customize the workspace fulfillers use.** Creator Studio automatically creates a workspace category where fulfillers work on requests. The workspace is ready to use as-is, but you can

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

tweak it to suit your needs. Find out how in Configuring workspace lists and records for form submissions.

8. **Test your app to see if it's ready.** Think your app is ready? Test it out to see your answers to form questions and automation results in the record your app generates. Get details in Test one of your forms for an app.
9. **Get your app deployed to production.** Once your app is spiffed up, tested, and ready to go, ask an admin to move it to a production environment for you so real users can begin using it to make requests. Find out how in Request deployment for your app from Creator Studio to production.

Use case: Sam's job at the Example Company is to answer employee requests for office equipment. They've done this for years by receiving emails and keeping spreadsheets of requests, requesters, equipment, prices, dates, and fulfillments. Their company is growing, and the number of requests is now overwhelming. "I'm spending all my time reviewing requests instead of doing all the other parts of my job." They'd like to create an app that enables people to request office equipment, automatically sends an email confirmation of the request, stores the request, and enables them to fulfill or reject the request with the click of a button. They've never written a line of code in their life, but with Creator Studio, they can create the exact application they want.

Creator Studio quick start

This quick start guides you through the process of building your first app in Creator Studio and requesting its deployment.

At a minimum, you need to create the app, and then add and customize a form, which is where users submit their fulfillment requests.

Building your first app is a good way to understand how Creator Studio enables easy app creation.

For a more thorough tutorial where you build a gift card request app, see Creator Studio tutorial.

Your system administrator must add you to the Creator Studio Users group.

Quick start part 1: Create an app

Create your app first. Later, we'll continue building it out.

Before you begin

Your system administrator must add you to the Creator Studio Users group.

Note: If you don't have permission to create apps yourself, you can ask your admin to create one for you. See Ask an admin to create an app for you in Creator Studio to find out how.

About this task

You can also watch a short video on how to create an app.

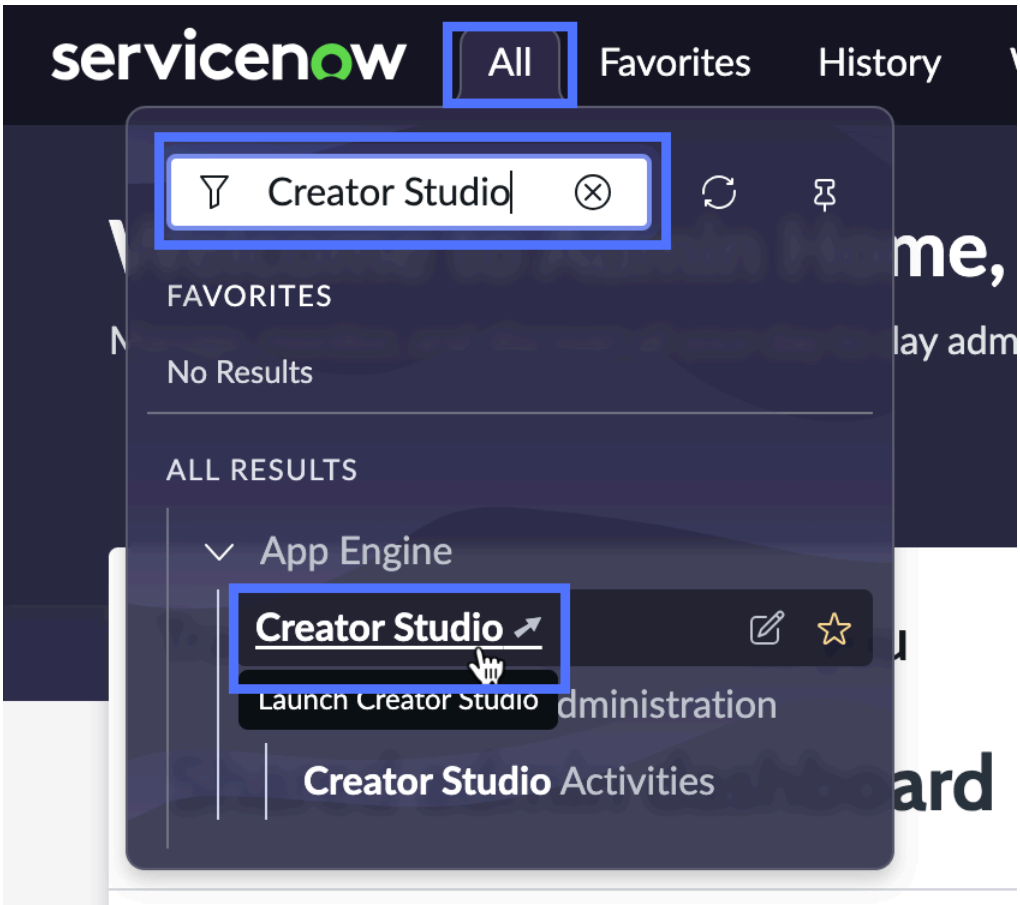
https://player.vimeo.com/video/1123376394?h=27cf6583d7&badge=0&autopause=0&player_id=0&app_id=58479

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Head over to the Creator Studio home page by going to **All** > **App Engine** > **Creator Studio**.

Start Creator Studio

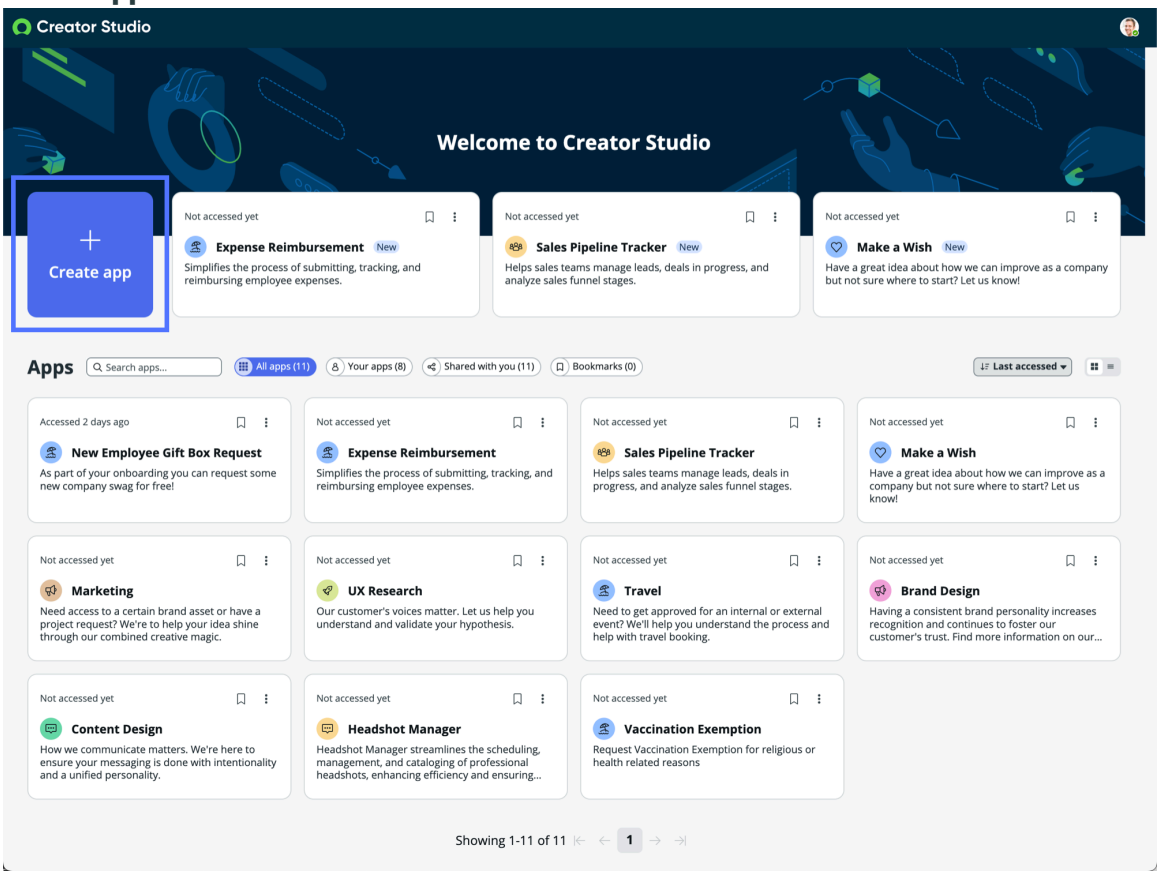


The Creator Studio home page appears. To learn more about working in the home page, check out Find existing apps in Creator Studio.

2. Select the **Create app** button to start the process of creating an app.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Create app button



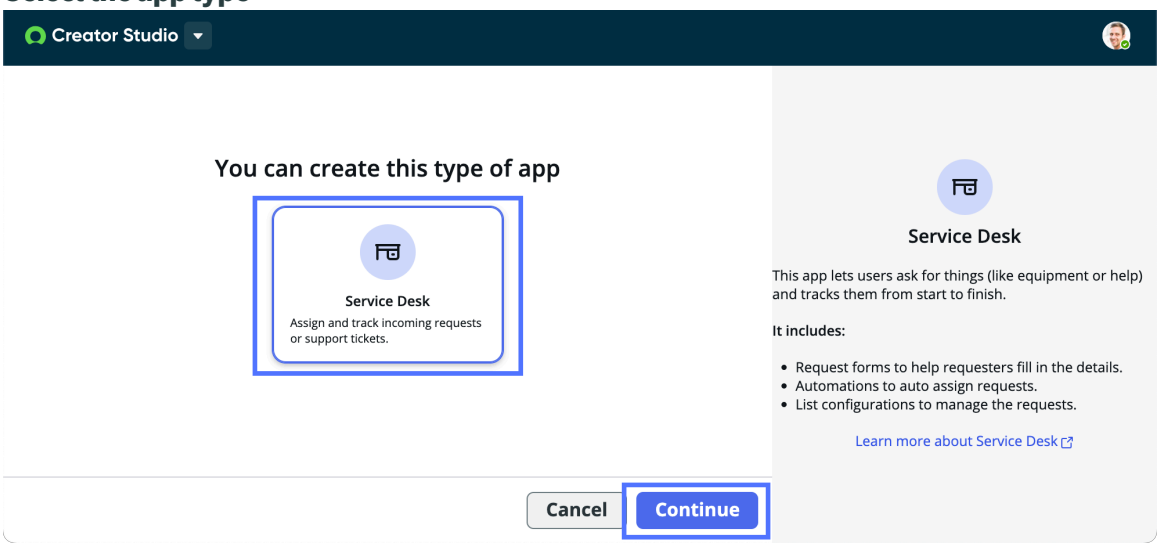
- If you're a system administrator, you can read more about this topic in Application collaboration.
- If you want to know how to request an admin to create the app for you, check out Ask an admin to create an app for you in Creator Studio.

3. Select the type of app you want to build, such as **Service Desk**.
   Service Desk apps enable users to submit requests, report issues, and access support related to services within your company.

   Note: Your admin may have hidden this page.

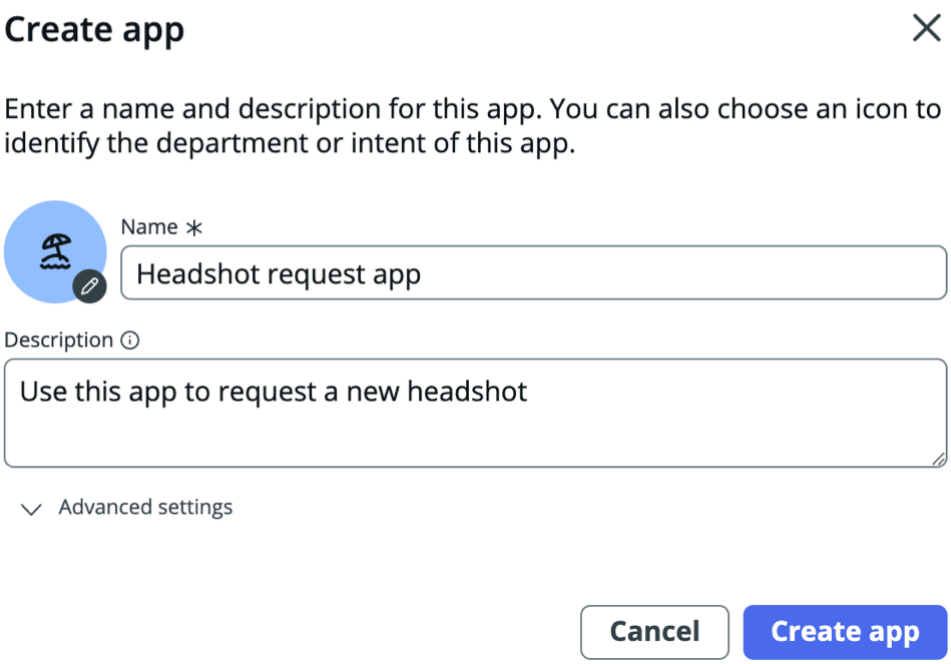
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Select the app type



4. Select the **Continue** button.
5. On the page that appears, enter a **Name** and brief **Description** of the app's purpose.

Create an app



6. Select **Create app**.

Result

Okay, you've created the beginning of your very own application! You've named and described it. Now, let's add a form.

Quick start part 2: Add a form

Let's add a form so people can make a request. Forms contain questions that people respond to when submitting a service desk request.

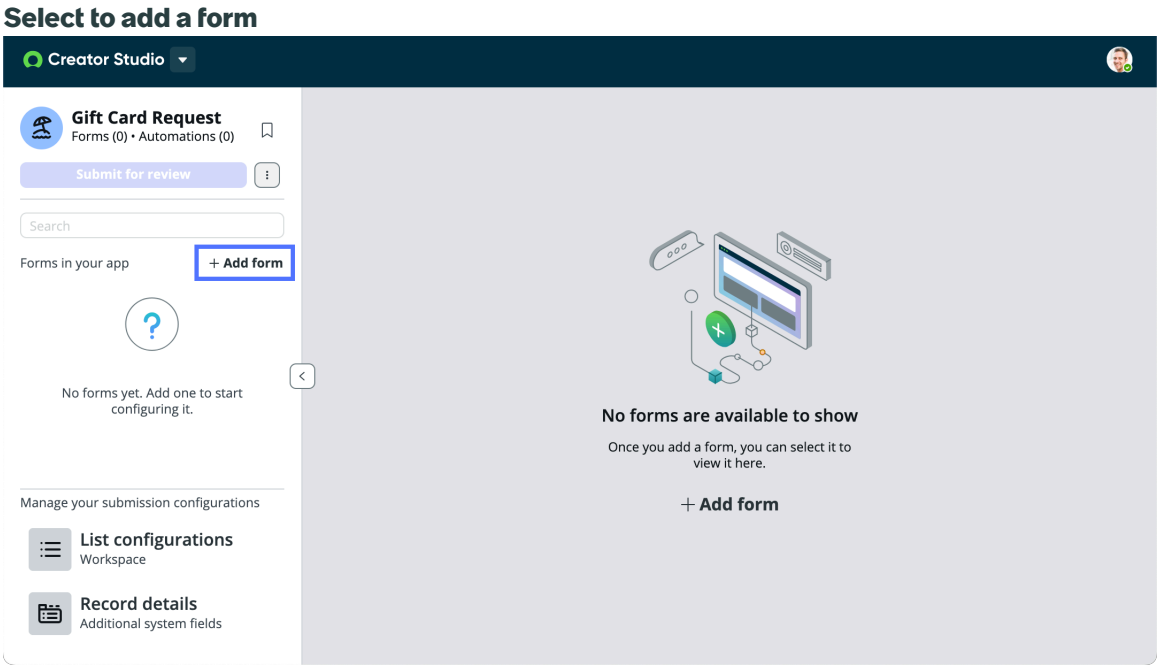
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

Your system administrator must add you to the Creator Studio Users group.

Procedure

1. Select **+ Add form** to get started.



2. Select a catalog template from the list, which your admin can customize, and then select **Apply template and continue**.

   You may not see a catalog template that is exactly what you want. So, choose one that's the closest. If none look close, select the **Creator Studio Default Template** option, if your admin hasn't removed it. It's your best bet.

   Tip: Feel free to click through the list of catalog templates. Previews will help you pick the one you want.

   Don't worry if you don't see other catalog templates to choose from! Your admins may not have created any custom catalog templates for you yet.

3. Choose how you want to build the form:
   - To use Now Assist, stay on the **Build with AI** tab and enter directions for the form you want. To learn more, see Generating forms with AI.

     Note: You can select **Try an example** to see some examples of the available directions. To see a quick mock up of what the form will look like, select the **generate form preview** button. If you want a different form, change the provided direction and select **Regenerate preview** for a quick look.

   - To manually build the form, select the **Build on your own** tab and enter the information for the form:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Form details

Field	Description
Form name	A descriptive name for the form.
Short description	Some information on what the form is used for.
Long description	Additional details for the form. You can include content like images and links to other sources of information (for example, a knowledge base article).

If you don't see the option to **Build with AI** tab, you need to check with your admin. To install AI for Form generation, see Configuring form generation in Creator Studio.

4. Select **Save and edit form**.

What to do next

Now that you've added a form to your app, it's time to customize it! Remember: You can add multiple forms to your app for if you need to.

Quick start part 3: Add questions to and publish a form

Add questions to the form and define what they are asking to gather the information that your fulfiller needs to evaluate the request. You can then publish it to make its catalog items available.

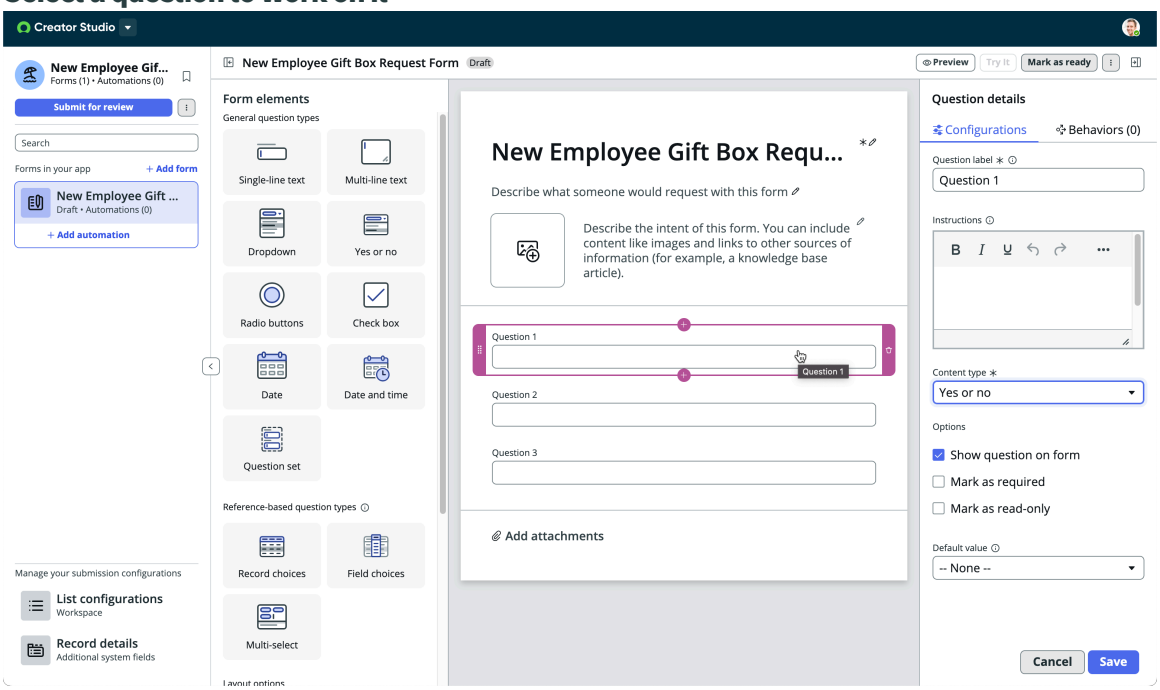
Before you begin

Your system administrator must add you to the Creator Studio Users group.

Procedure

1. Select the question that you want to customize.
   When you select a question, it's highlighted on the form so you know what you're working on.

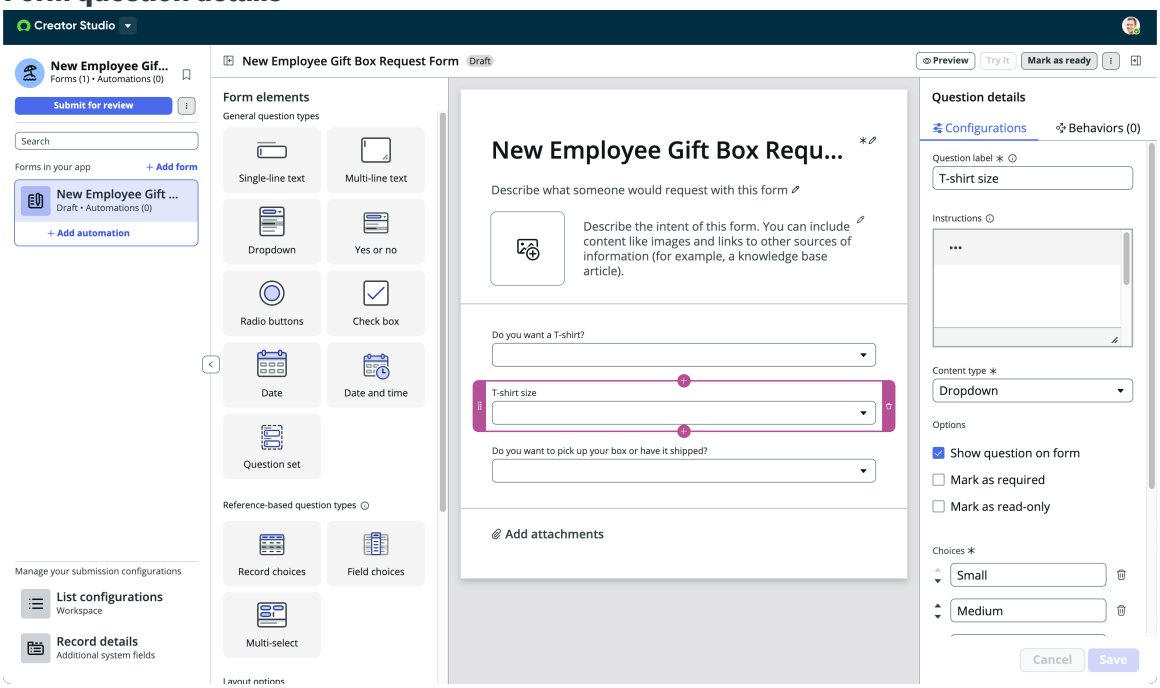
Select a question to work on it



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

2. In the **Configurations** tab of the Question details panel, specify information about the question you added, such as a label for what the question is asking and whether it must be answered for the requester to submit the form.
   The details vary by question type. For example, if you add a **Dropdown** question, you must supply the options to choose from.

Form question details



3. **Optional:** Make the form's appearance change based on how users answer questions by adding dynamic behavior to it on the **Behaviors** tab.
   For example, if a user says they want a T-shirt for an event they're attending, you can make a **T-shirt size** field required. Get the details on adding dynamic in Make a form change based on responses in Creator Studio.

4. Automatically populate a question's answer based on the answer to another question on the form (record choices questions only) by selecting the **Add auto-fill** option.
   For example, a form can enable someone to select a user for the **Choose the gift box recipient** field, which retrieves its values from the User table. An additional **Preferred name** field can be automatically populated from the **First name** field of the User table selection, but then edited if someone prefers a nickname.

   The **Add auto-fill** option is available only if the form has a record choices question on it. You must also complete the following fields:

Details for auto-populating a question based on a record choices answer

Field	Description
Select the dependent question	Question whose answer makes the current question automatically populate.
Select the value field	Field in the source table for the record choices question that contains the value you want to use as the answer to the current question.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
	You must select the edit icon ( ), navigate to the field in the source table that contains the auto-populated answer, and select to **Apply** it. For more info on selecting fields in a table, see Dot-walking to data in related tables.

5. Select **Save and close** when you finish modifying the question.
6. **Optional:** Add another question by dragging the type of question you want from the Form elements panel onto the form, dropping it on the canvas where you'd like it.
   You can also add questions by selecting the add icon (+) that appears when you click on an existing question on the form. If you're adding a pre-configured **Question set**, you must select the question set from the modal that appears when you drag it onto the form.

   For a description of question types and how they're used, see Available question types in Creator Studio.
7. **Optional:** Change a question type by selecting the question and then choosing the new question type in the **Content type** field of the Question details panel.
   Selecting a new type may introduce new values you must supply.

   Keep these specifications in mind as you create your questions:

   - You can't change an existing question into a question set. To include a question set on a form, you must newly add it to the form.
   - If you put two checkbox questions side-by-side on a form, they make a section. You can't add other types of questions to that section.
8. Now, let's arrange the questions and images that you've added to the form.

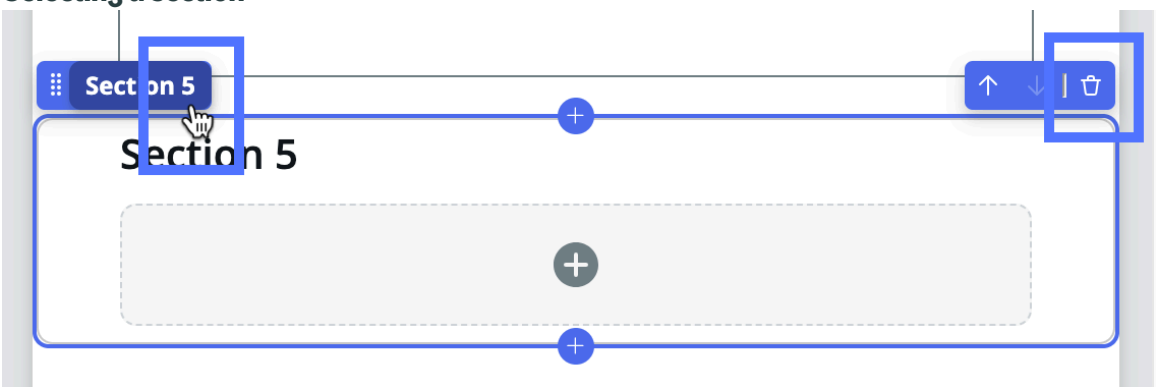
   a. From the Form elements panel, drag the layout option you like onto the form and drop it where you want it to appear, for example, a **Divider line**.
   Don't worry if you don't like the layout, just try another one by dragging it onto the form's canvas.

   b. Revise the form layout you chose using the Section details or Question details panel (depending on the layout you're working on). You can do things like make text bold, add links, and so forth.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: To edit or delete a section, you must hover over the section name and then select **Section** to see the section details in the properties panel, as well as the delete icon.

Selecting a section

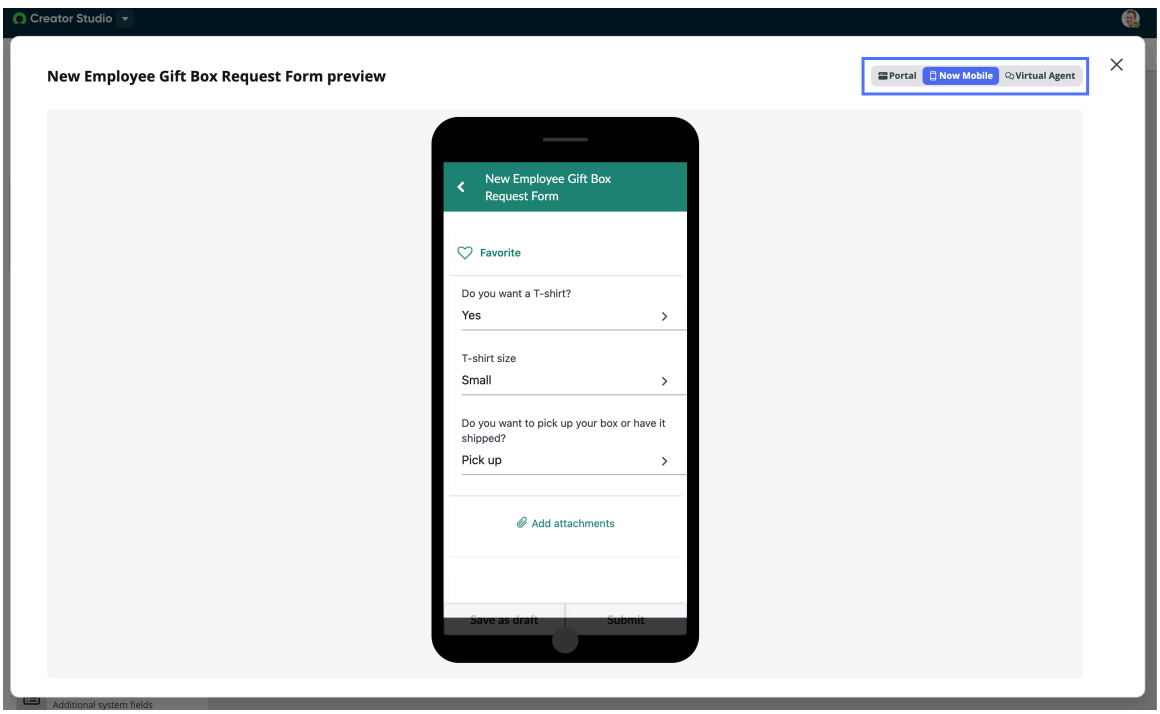


For more information, see Layout options for forms in Creator Studio.

c. Select **Save** in the Section details/Question details panel when you're done revising the form's layout.

9. **Optional:** See how your form will look in various experiences by selecting the **Preview** button. You can see a preview of the how the current version of the form will appear in various experiences by selecting **Portal** (such as Employee Portal), **Now Mobile**, or **Virtual Agent**. You can fill in the form when previewing, but selecting the **Submit** button doesn't generate a task record.

Preview a form

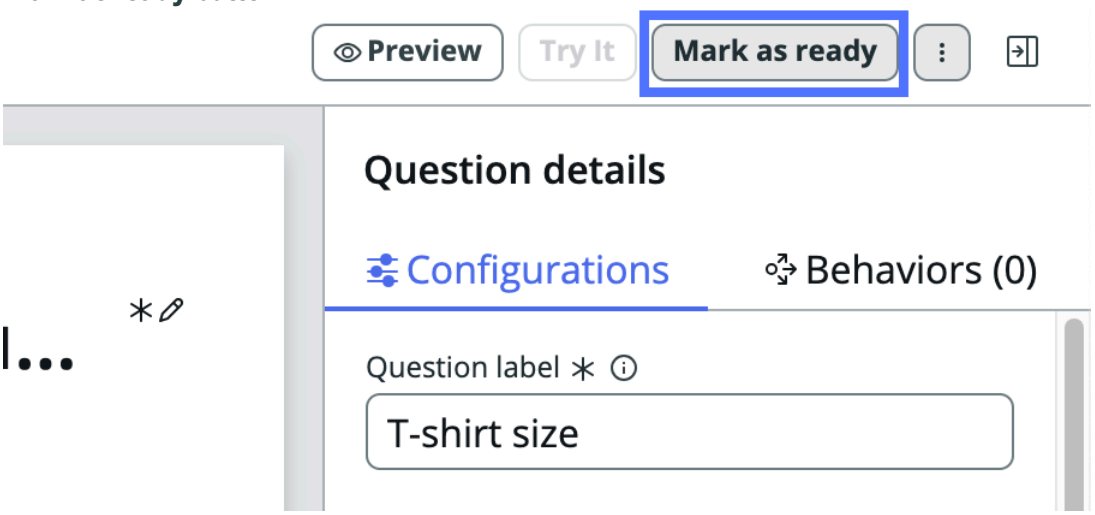


10. Publish the form when it's ready by selecting the **Mark as ready** button.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Publishing forms makes them available as catalog items in the production instance after the associated app has been deployed.

Mark as ready button



Result

Now that the form is available as a catalog item, we can use it to create an automated playbook.

Quick start part 4: Add a playbook

Add a playbook to create automation for your app, for example, to automatically assign a record to a manager for approval.

Before you begin

The form that you want to add automation to must already be published.

Your system administrator must add you to the Creator Studio Users group.

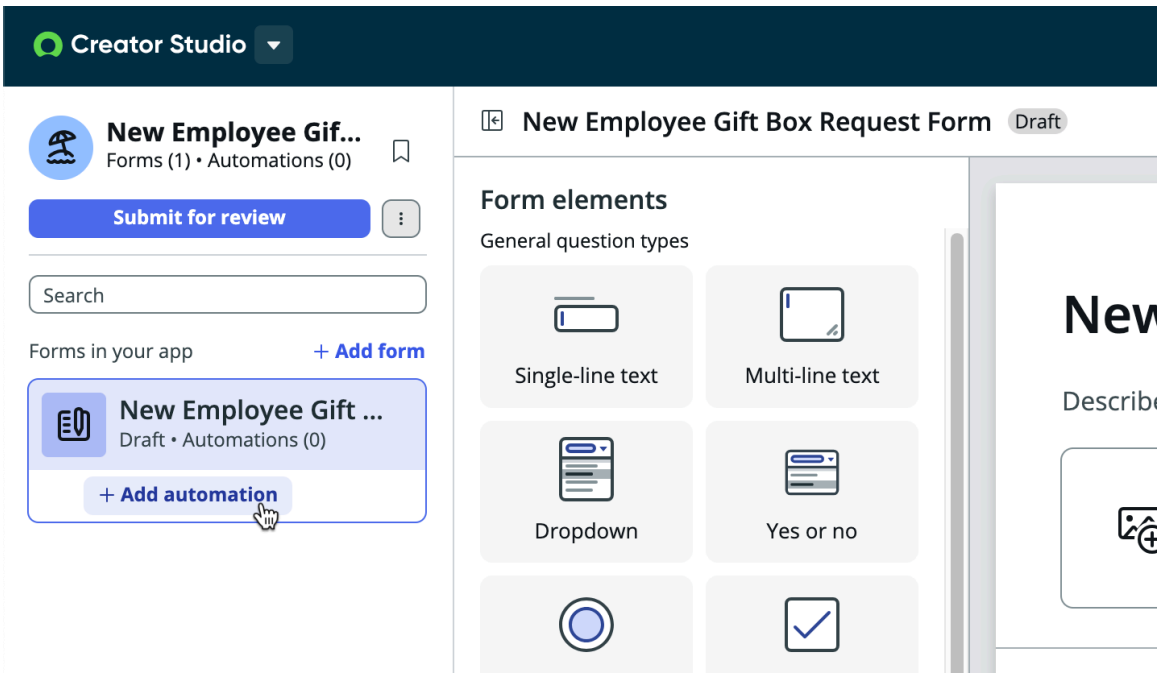
Procedure

1. Select **+ Add automation** in the navigation panel for the form you want to add a playbook to.

> **Note:** If the form isn't yet published, you must first mark it as ready. For more information, see Publish a form for your app in Creator Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add automation



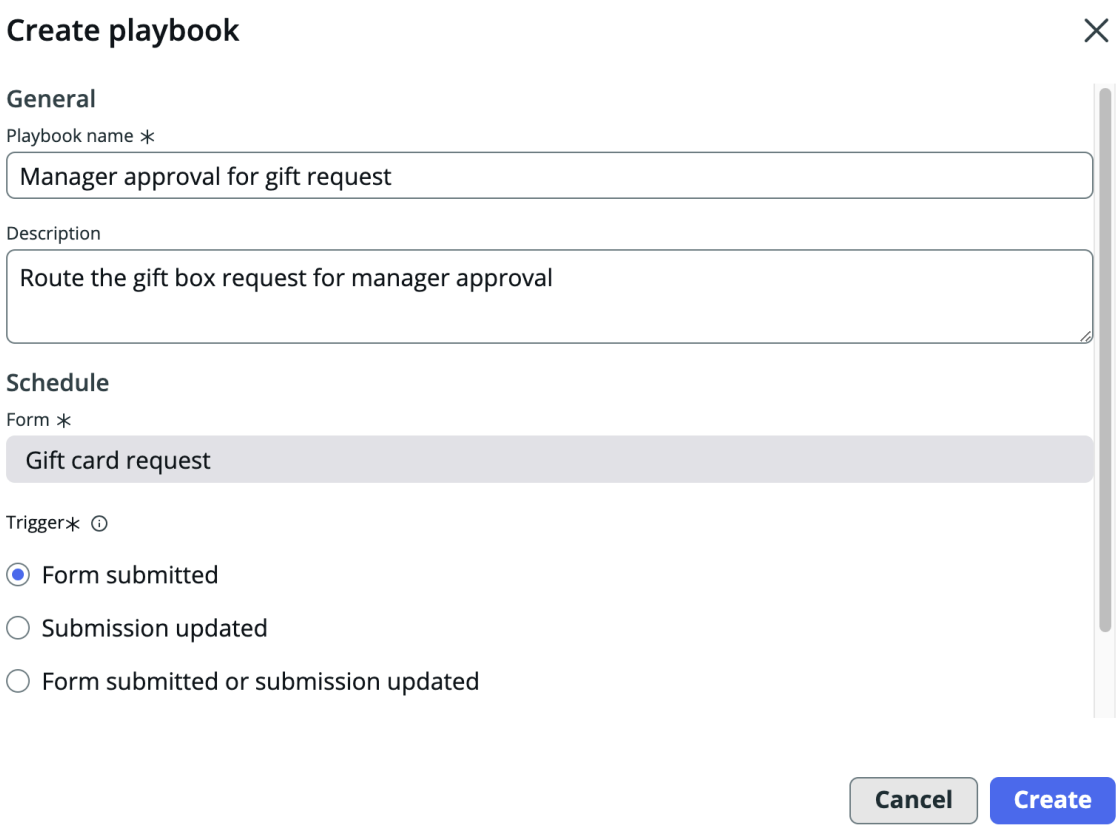
2. Specify the playbook's General attributes in the Create playbook modal.

General playbook definition fields

Field	Description
Playbook name	Descriptive name for the playbook you're creating.
Description	Brief explanation of what the playbook does, for example, the end goal for the record type.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Create a playbook



3. Specify the Schedule for the playbook.

a. Select the type of **Trigger** that initiates the playbook.

Playbook trigger options

Trigger	Description
Form submitted	Start running the playbook when a user submits the form you chose.
Form updated	Start running the playbook when a user updates the form you chose.
Form submitted or updated	Start running the playbook when a user submits or updates the form you chose.

Note: You can't change an playbook's trigger type after you finish creating the playbook. Instead, create a new playbook with a different trigger.

However, you can edit the trigger condition, such as making the playbook run conditionally based on a specific answer to a question. For more information, see Edit the trigger for a playbook in Creator Studio.

b. If you chose a trigger that includes a form being updated, specify how often that app should **Run your playbook**.
The options are:

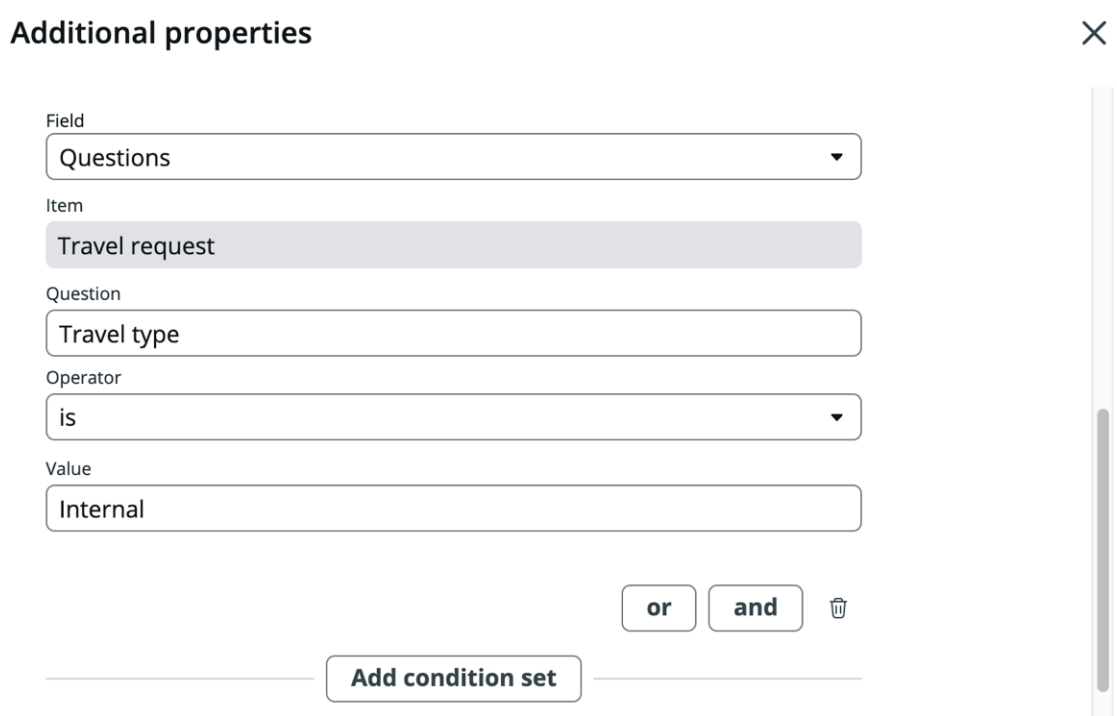
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- **Once**
- **For each unique change**
- **Only if not currently running**
- **For every update**

c. Specify the conditions that must be met for the playbook to begin running by selecting **Add conditions**.

- If you want to trigger the playbook based on the value of a column in a table, select the **Field** that you want to be the trigger, as well as its condition **Operator** and the specific trigger **Value**. For example, when a **Start date** is **after** the **Date** needed.
- If you want to trigger the playbook based on the response from a form, select **Questions** as the trigger **Field**. Then select the question you want in the **Question** field, the condition **Operator** and the answer's **Value**.

Question answer as trigger for an automation



Add as many conditions as you need. For more information, see Create a condition statement using the condition builder.

d. Select **Create**.

Result

Next, we'll add an activity to the playbook to define what the automation does.

Quick start part 5: Add an activity and activate a playbook

Add an activity to your playbook to define what the automation does.

Before you begin

Your system administrator must add you to the Creator Studio Users group.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

About this task

If you want to include an if/then statement to define circumstances for the activity, add a decision. See Add a decision to an app's playbook in Creator Studio for details.

Procedure

1. Ensure that you're viewing the playbook that you just created.
2. Select the add icon on the connector where you want to add an activity and choose the square **Add an activity** icon ( ) in the menu that pops up.
3. Choose the type of activity that you want from the Activity library pop-up.

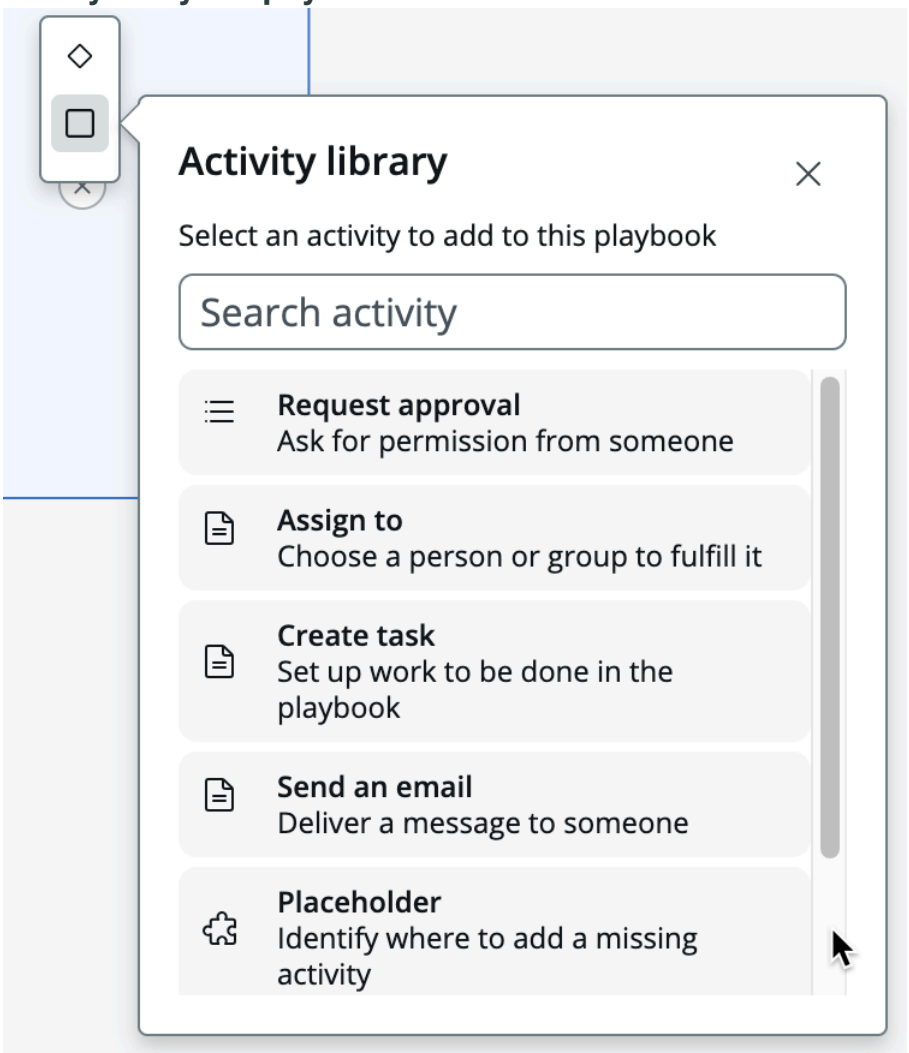
Note: In addition to the following standard activities, you may see some custom activities that your admin created.

Types of activities

Activity type	Description
Request approval	Ask someone for permission to accomplish a task.
Assign to	Choose a person who should fulfill the task.
Create task	Specify a process that must be done as part of the playbook.
Send an email	Send an email to one or more people. You can specify images and enrich text for the email that gets sent automatically.
Placeholder	Set an undefined activity to be specified later, or a more advanced activity such as an email notification, when an activity is completed. Placeholder activities don't have any logic assigned to them yet, and must be edited in Workflow Studio. Or, you can swap them out later for another type of activity in Creator Studio.
Update submission	Update the value of the specified fields on the record that the form generates when a user submits it.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Activity library for a playbook



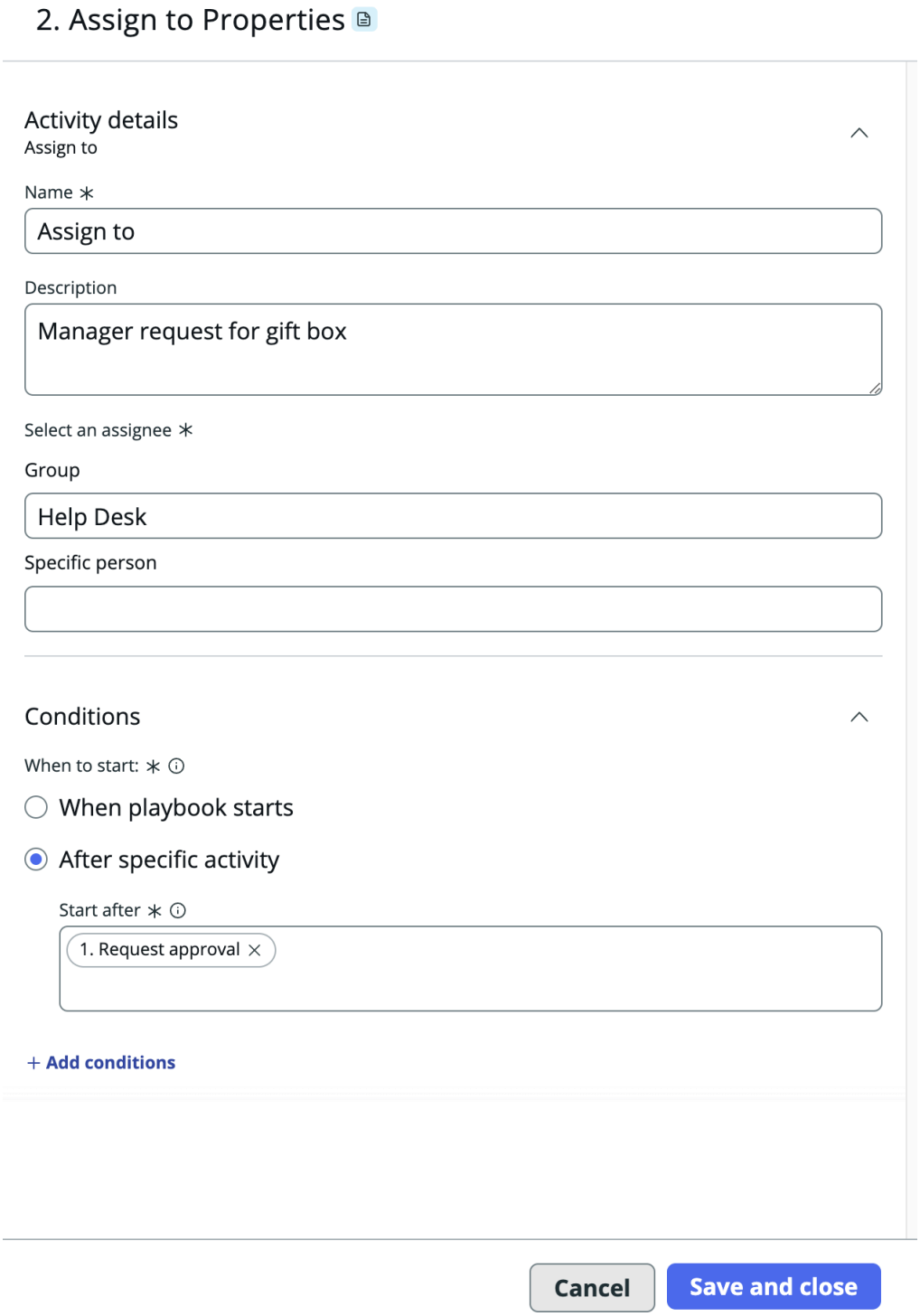
4. Enter the basic details for the activity.

General activity details

Field	Description
Label	Unique, user-facing name for your activity, which appears to agents and fulfillers while the playbook is running.
Description	Optional details about what the activity accomplishes.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Activity details panel

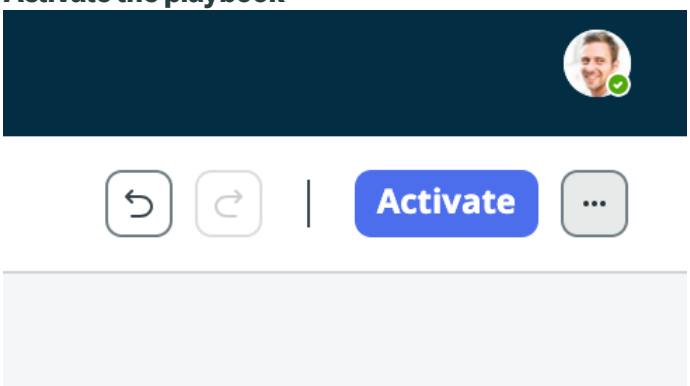


5. Complete the details for the activity, using Add activities to an app's playbook in Creator Studio for information on how to finish creating the activity.
6. Select the **Save and close** button to finish setting up your activity.
7. Once the automation is done, activate it by selecting the **Activate** button.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Activating a playbook makes it available to run when its related form is created or updated on your non-production, development instance.

Activate the playbook



Result

The app is now ready to be deployed to production! Let's request deployment in the final step of this quick start guide.

Quick Start part 6: Submit your app for deployment

Now that you've made an app, it's time to submit it for review so admins can approve and deploy it.

Before you begin

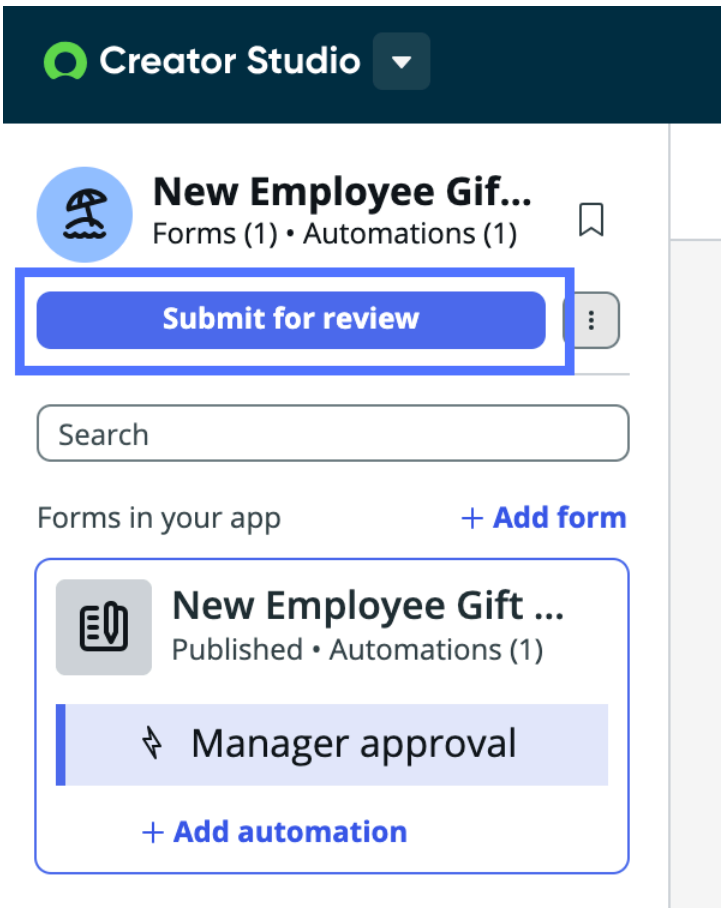
Your system administrator must add you to the Creator Studio Users group.

Procedure

1. **Optional:** Test the form by selecting the **Try it** button.
   You can then enter test values in the fields and select **Submit** to view the record your form submission created.
2. In the navigation panel, select **Submit for review**.

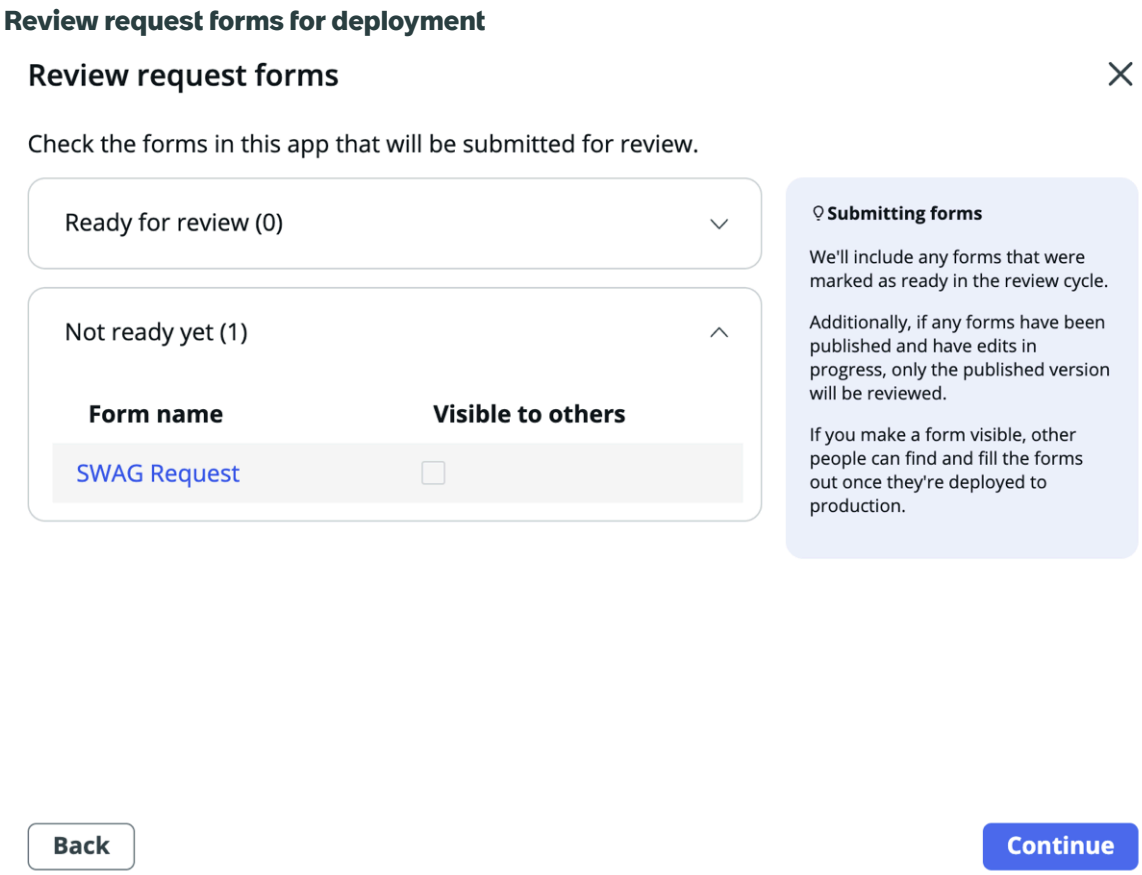
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Submit for review



3. Select **Continue** on the Submit app for review modal.
4. Now you need to choose which published forms are visible to users in the catalog. In the Ready for review section of the Review request forms modal, select which of the app's published forms that you want to be available after the app is deployed by selecting the **Visible to others** option.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



5. Select **Continue** when you're happy with the forms being deployed to production.
6. Next, you must decide which of the app's activated playbooks will run on production after the app is deployed. In the Review playbooks modal, select the **Run on production** option for each playbook that you want to run on records that the app generates.

   Note: If you can't select a playbook, you need to go back to the **Automations** section of Creator Studio and activate it. If you need a refresher on that, check out Activate a playbook in Creator Studio.

7. Select **Continue** when you're happy with the playbooks being deployed to run on production.
8. Finally, make sure that all the release details for the published app are correct.

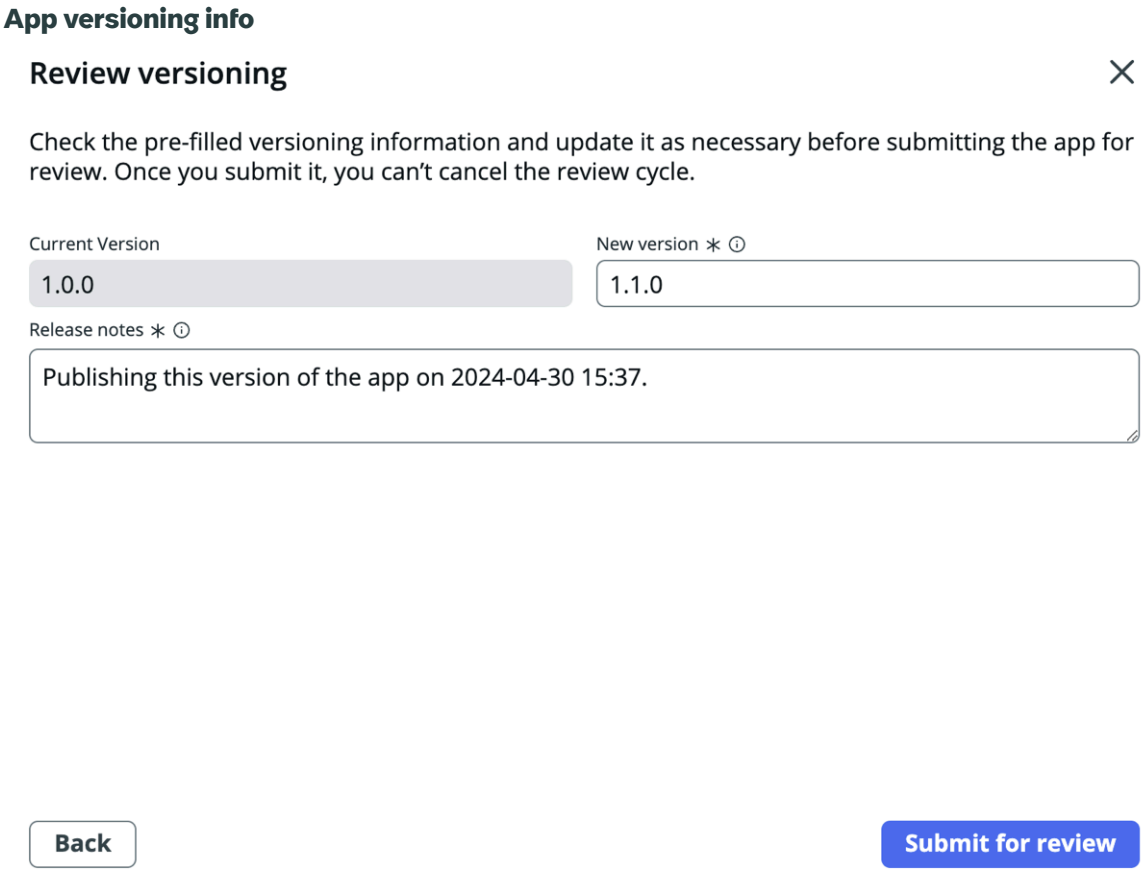
Versioning options for deployment

Field	Description
New version	Version number of the app you're requesting for deployment. Creator Studio automatically generates an updated version number, but you can change it. Follow your organization's versioning guidelines, or use the x.y.z format, where x = major update, y = minor update, and z = patch.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Release notes	Details on what's changed in this new version of the app, or a general description of what the app does if this is its first version.

Read more about this step of requesting deployment in App versioning and release notes for Creator Studio apps.



9. Select **Submit for review** when everything is correct and ready for your admin to review and deploy.

Result

Congrats: Hooray! You've created your app, customized the form, added automation, and submitted it for review. After your admin reviews and deploys it, people will use your brand-new app to make requests.

Get help with Creator Studio

To get help with Creator Studio, your ServiceNow instance, plugins, permissions, and more, watch a short video to contact the ServiceNow admin who works in your company. You can also check some helpful resources.

Get help now from your Now Support administrator

The ServiceNow admin in your company can help you with things like activating plugins, kicking off upgrades, unlocking accounts, updating user permissions, and much more.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

https://player.vimeo.com/video/1056112088?h=1a5cd3b245&badge=0&autopause=0&player_id=0&app_id=58479

Helpful resources

Some ServiceNow resources that can provide you with helpful information are:

Video

Watch Build apps fast with Creator Studio.

ServiceNow Community

Creator Studio Community

FAQs

Creator Studio FAQs

Choosing your development experience

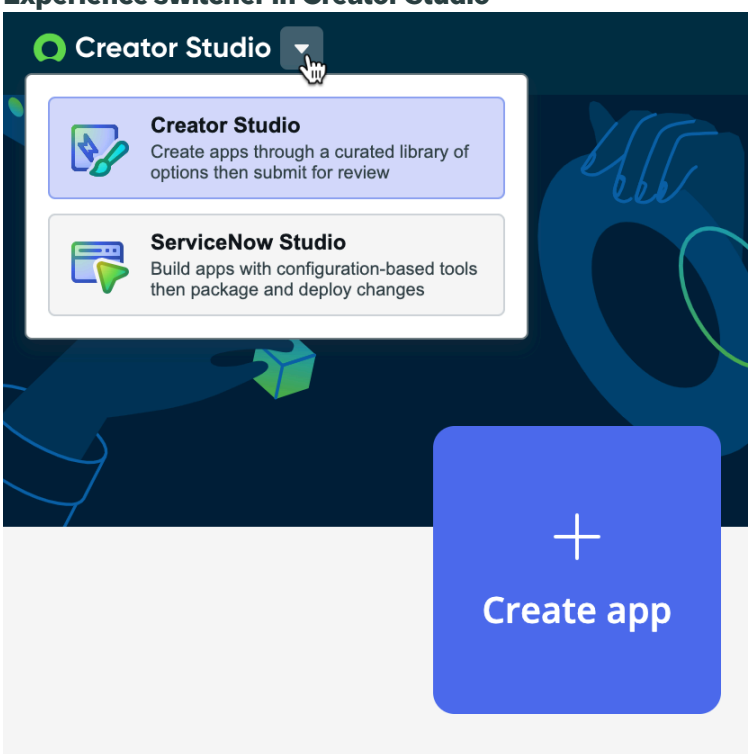
If you want to develop apps in a more robust environment than Creator Studio, which was designed for no-code developers, you can select a different experience directly from within Creator Studio.

A development environment for every level

Use the experience switcher to select from any of the following development environments directly within Creator Studio, without navigating to the **All** menu.

Important: No matter what your role is (including admin and delegated developer roles), if you have one of the Creator Studio roles, you won't be able to access ServiceNow Studio or ServiceNow IDE.

Experience switcher in Creator Studio



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Development experiences

Development experience	Level	Description
Creator Studio	No-code	Enables quick creation of basic applications through an intuitive, guided environment, ideal for beginners or those needing fast results.
ServiceNow Studio	Platform developer	Offers a balance of simplicity and power, facilitating the development and management of metadata and integrated builders within a navigable interface.
ServiceNow IDE	Pro-code	Provides advanced developers with a professional coding environment, using VS Code for direct metadata editing, suited for intricate and detailed application development.

Being able to easily switch between development experiences eliminates the fragmentation of development tools, reducing confusion and centralizing development activities across all skill levels.

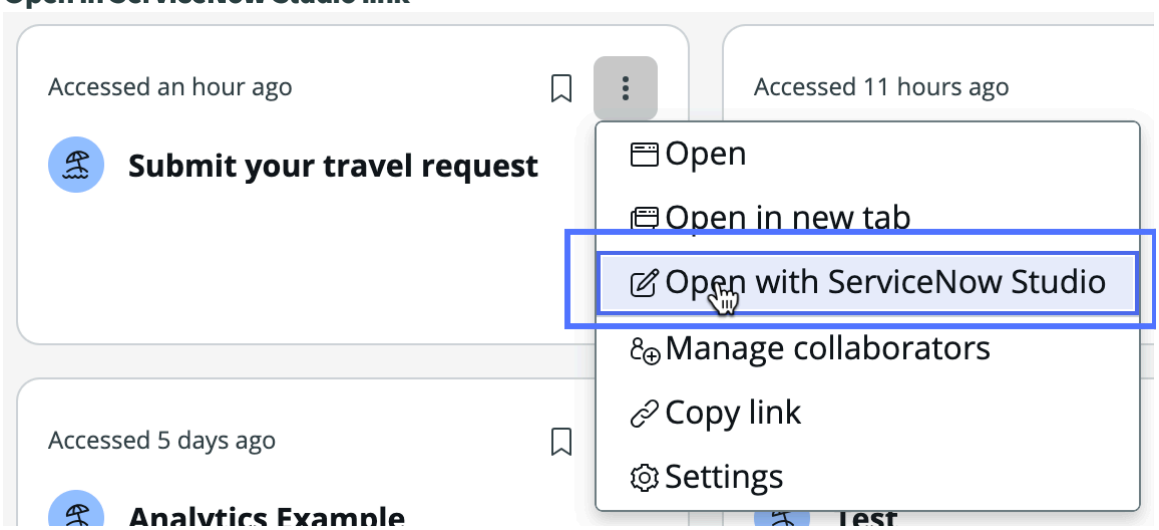
For example, you can build a basic request app in Creator Studio, and then switch to ServiceNow Studio to add more robust functionality.

For details on switching experiences, see Change your development experience.

Open an app directly in ServiceNow Studio

If you have access to ServiceNow Studio, you can open an app there by selecting the more actions icon on the home page (···) and choosing to **Open in ServiceNow Studio**.

Open in ServiceNow Studio link



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

You can also select the more actions icon in the navigation panel and then select **Open in ServiceNow Studio**.

Sharing apps between development experiences

All apps that are made and bookmarked in Creator Studio are bookmarked and available to work on in ServiceNow Studio. However, apps that are made in ServiceNow Studio are not available to work on in Creator Studio.

Who can access the studios in the experience switcher

The experience switcher can provide access to Creator Studio, ServiceNow Studio, ServiceNow IDE. However, whether you can see and select all of those depends on your role or access level. For more information, see Administering access to the experience switcher.

Note: In addition to Creator Studio, you must have access to at least one other product (ServiceNow Studio or ServiceNow IDE) to see the experience switcher.

The following describes default role access to the experience switcher:

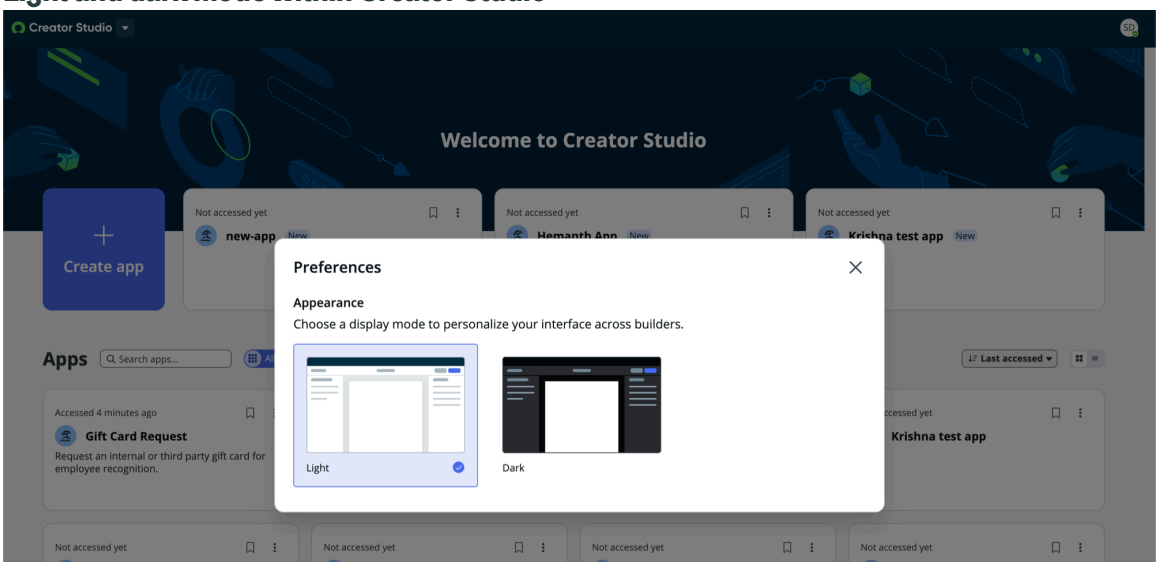
- Admins and delegated developers can use the experience switcher because they may need access to any product where they've been delegated to administer or develop an app.
- Creator Studio users and Creator Studio restricted users don't generally have access to the experience switcher because administrators limit them to a more curated experience.

Dark mode in Creator Studio

Creator Studio is available in light and dark mode.

You can choose light or dark via **User menu** > **Preferences** while in Creator Studio.

Light and dark mode within Creator Studio



Service desks and Creator Studio

Service desk apps enable users to request things, such as IT equipment or travel bookings, and track their fulfillment.

Service desk apps are usually accessed through a portal, such as Employee Center, or a service catalog.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Parts of a service desk app

Service desk apps contain the following parts:

- Forms that collect information from the user about the request
- Automated playbooks, such as routing approval when a request is made or sending automatic email notifications
- List configurations for a workspace where fulfillers can work on requests, where they can do the following:
  - Communicate with the requester and store messages in the record, which is accessed in the workspace
  - Save attachments related to the request
  - Share notes with other employees in different departments when fulfilling a request
  - See automation results

Example service desk catalog items

Example catalogs and their items

Service catalog	Example app	Sample catalog items
HR self-service	Human Resources global people portal app	• Benefits coverage question • Time off request • Change of address needed
Branding resources	Marketing and branding app	• PowerPoint templates • Approved Zoom backgrounds • Logo files
IT fulfillment	Hardware request app	• Mouse device • Headset • Replacement charging cable

Other types of apps

See Example apps you can build in Creator Studio for some other, less-recurring ideas for apps.

Migrating to Creator Studio from Service Creator

Creator Studio is replacing the older Service Creator tool, which will be retired in the Australia release.

Note: For information on older versions of Service Creator, refer to the previous release documentation.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Why you should use Creator Studio instead of Service Creator

Creator Studio has a more modern interface than Service Creator. Creator Studio uses Record Producers directly, implements workflows with Playbooks, and allows fulfillment via Workspaces. It also integrates with deployment pipelines to promote applications created with Creator Studio to production environments.

How to try Creator Studio

If you're currently using Service Creator and want to try Creator Studio, follow these steps:

- Confirm that your instance is running at least Xanadu patch 3.
- Purchase an App Engine Enterprise license.
- Download Creator Studio from the ServiceNow Store. For more information, see Installing Creator Studio from the ServiceNow Store.
- Activate Creator Studio.

Playbooks vs. workflows

Creator Studio uses playbooks instead of workflows to automate processes in the apps you build.

Playbooks are a series of steps triggered by an event. You can add multiple playbooks to an app if needed

The steps in a Creator Studio playbook, which include activities and decisions replace the way Service Creator used to define service configurations.

In Service Creator, you had to first create a service category, then define the service and its table. In Creator Studio, the table is automatically created when you make an app, simplifying the category selection process.

For more details, check out Working with automation in Creator Studio.

App compatibility with Creator Studio

You can open apps built in Creator Studio in other ServiceNow products, but only apps built in Creator Studio can be opened in Creator Studio.

Opening Creator Studio apps in other products

After you build an app in Creator Studio, you can open it in several other products. For example, you could build an app in Creator Studio and then use the **Open in ServiceNow Studio** link from the app on the home page open it in ServiceNow Studio, where you can add more complex flows in the automation.

If you add complex functionality to an app in another builder and then re-open the app in Creator Studio, you'll be warned in the appropriate context. For example, if you add complexity to a playbook that's not supported in Creator Studio, then Creator Studio will display a minimal version of the playbook or activity and instruct you to open it in Workflow Studio.

For more information, see the following products:

- ServiceNow Studio
- Catalog Builder

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Opening apps built in other products in Creator Studio

You can't open apps created outside of Creator Studio in Creator Studio. For example, you can't open apps built in ServiceNow Studio in Creator Studio.

However, if you create an app in Creator Studio and then add a form to it from outside of Creator Studio (for example, from Service Catalog), then Creator Studio tries to show the outside form.

Creator Studio apps and tables

Creator Studio uses catalog templates to streamline the app creation process. Apps use the Request Task table by default, generating a new row for each submitted request, though admins can change the associated table for an app.

Summary: After reading this section, you'll understand:

- The parts that go into building an app
- Where requests from the apps go
- How tables are used to store requests

Parts of building an app

Apps contain one or all of the following parts.

Parts of an app

Functionality	Description
Form	When filled out, forms create a record with an associated task or workflow on the ServiceNow AI Platform.
Automation	Automation contains playbooks, or automated processes with limited actions that enable users to update records and complete tasks across multiple activities.
Workspace list configurations and record details	Workspace configuration that shows requests for the app built in Creator Studio, with standard and customizable filtered lists. Selecting a record in the **List configurations** section displays the record, which you can interact with in Creator Studio, enabling you to see how it will look.

Your admin can create catalog templates to guide the process of building forms. For more information, see Working with forms in Creator Studio.

Where requests from the apps go

Every app in Creator Studio automatically creates a table where all opened records go when people use the app. By default, the tables that Creator Studio apps generate extend the Request Task table that comes with Creator Studio, which extends the Task table. Your admin may change the associated app for a table, but it should still extend the Task table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note:

- If an app's associated table is changed to one that doesn't extend the Request Task table, automations won't work.
- You can change the **Request type** field, which specifies the form, on the Request Task table or a table that extends Request Task. To do so, you must be an admin or have the sn_creatorstudio.configuration_admin or sn_creatorstudio.task_admin role.

Extending a table means the new table inherits the parent (extended) table's columns, as well as its business logic. For more information on Task tables, see Working with the Task table.

For sys admin eyes only: Every app built in Creator Studio adds a record in the Request App Config table, which contains the table that the app uses. Your admin can change the table for an app, see Administering an app's associated table for more information. The name of the table follows the format of scope_request, for example, x_snc_02_03_request.

Key term:

Record

A record is what the ServiceNow AI Platform generates for each request submitted through your app. Each record corresponds to a row in the app's Request Task or associated table.

Publishing, activation, and deployment workflow for forms, automation, and apps

When you build an app in Creator Studio, you must create forms and automation. You can also customize the workspace list configurations and records that fulfillers use before everything is deployed to production.

As you develop apps in Creator Studio, you should move sequentially through the following sections. For example, you should mark all of an app's forms as ready in the **Forms** section of Creator Studio before you begin to work on the app's **Automations** section.

1. When you're done building forms, publish them by marking them as ready.

   Published forms are then available in the catalog on the non-production instance that you're using for development.

2. When automated playbooks (which are based on the forms you already created) are ready, activate them.

   Published playbooks can be triggered when a form is created or updated on the non-production instance you're using for development.

3. The app's fulfiller workspace is already available on the non-production instance that you're using for development for you to test and customize. Additionally, you can interact with test records by selecting a record in a **List configuration** section. You can then change how the record appears using the **Record details** of the navigation panel.
4. Test out submitting a form using the **Try it** button.
5. When everything looks good and you're ready to deploy the app to production, you submit it for review.
6. After your admin deploys the app to production, the app is available to users whose role gives them access to it. On the production instance:
   - The app is available to users
   - The catalog contains all published forms

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Activated playbooks are triggered on applicable records
- Unactivated playbooks are deployed but must be activated by an admin before they can run
- Users can access the workspace to review the app's submitted requests

7. To update a published app, make the changes on your non-production instance and then request that your admin redeploys the updated app to production.

Configuring Creator Studio

Admins need to install Creator Studio before it can be configured for users to start building apps.

Configuration overview

The following is a general overview of installing and configuring Creator Studio.

1. Decide on an instance strategy, for example, what are your development (non-production) and production instances. For more information, see Creator Studio development instance strategy.
2. Download Creator Studio from the ServiceNow Store and install it. For more information, see Installing Creator Studio from the ServiceNow Store.
3. Run Guided Setup to configure administration and collaboration. For more information, see Configure Creator Studio using Guided Setup.
4. Ensure all users who need access have the correct role. For more information, see Creator Studio roles and personas.

Using Guided Setup to implement Creator Studio

Guided Setup provides a sequence of tasks that help you configure Creator Studio on your ServiceNow instance. Currently, the Guided Setup helps you to set up the following:

- The admin group to define who can administer Creator Studio
- Collaboration descriptors to manage what app owners and editors can do
- Who has full and restricted access to creating apps in Creator Studio

To open Guided Setup for Creator Studio, navigate to **All** > **App Engine** > **Guided Setup – Shared**.

For more information, see Guided Setup .

Catalog configuration requirement for Creator Studio

To ensure that forms appear correctly for users, the non-production and production instances must have the same Service Catalog and all of its categories.

Configuring Virtual Agent chatbot previews

If you want forms to appear in the Virtual Agent chatbot, you must install the necessary plugins. For more information, see (Legacy) Activate Virtual Agent for NLU .

Installing Creator Studio from the ServiceNow Store

Installing Creator Studio from the ServiceNow Store makes it available for people to build apps on your instance.

Your company must have an App Engine Enterprise license to install and configure Creator Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

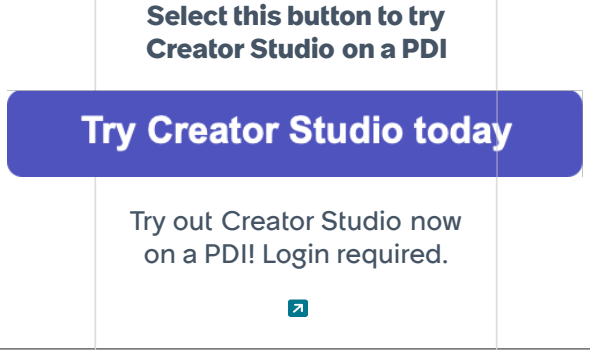
For instructions on installing Creator Studio and other products from the ServiceNow Store, see Install an application or plugin.

Note: If you want forms to appear in the Virtual Agent chatbot, you must install the necessary plugins. For more information, see (Legacy) Activate Virtual Agent for NLU.

After Creator Studio is installed on an instance, admins have to configure it. The easiest way to do that is using Guided Setup, find out how in Configure Creator Studio using Guided Setup.

Give Creator Studio a try

Ready to give Creator Studio a try? You can test it out using your own Personal Development Instance (PDI), which requires you signing in to the Developer Site. Find out more on PDIs in the Personal developer instance guide.



Creator Studio development instance strategy

Make sure to install Creator Studio on all ServiceNow instances where users will be building applications, including the production instance.

Deciding on your instance strategy

You need to decide how you want to manage access to Creator Studio within your company. Consider the following options:

- **Open Access:** Allow everyone in your company to use Creator Studio to create apps.
- **Restricted Access:** Limit access to a specific group of users.
- **Request-Based Access:** Set up a form where users can apply for access. Admins will review these requests and decide whether to grant access.

Development and deploying to production instances

A non-production instance that is similarly configured to your production instance may be the best candidate for your test environment. You can then more accurately find issues that may arise if the application is deployed to production.

You should ensure that developers build apps in Creator Studio on a non-production instance, and then deploy apps that are ready and approved to production.

When you deploy an app, the records are referenced in the Store Apps [sys_store_app] table on the production instance. But when you're developing an app, the records are referenced in the System Applications [sys_app] table. So if you develop in production, you're developing using the [sys_app] table instead of [sys_store_app].

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

After you establish your instance strategy, you must also establish and automate your approval or review process. Creator Studio runs on your non-production environment, and admins then deploy apps to the production environment. For more information on the deployment process, see Deploying your Creator Studio app.

If your organization has multiple non-production environments, you must decide which non-production environment Creator Studio will run on. You must also determine which pipeline to use for promoting apps from a particular non-production instance to your test instance, and then finally to production where the app will be running live. For more information, see Pipelines and Deployments.

Catalog configuration requirement for Creator Studio

To ensure that forms appear correctly for users, the non-production and production instances must have the same Service Catalog and all of its categories.

Developer roles and testing apps on instances

If you have a Creator Studio role of sn_creatorstudio.user or sn_creatorstudio.restricted_user, you won't be able to test the apps you build on the non-production instance's Request App Workspace. You should be able to test the app on the non-production instance using Creator Studio's app previews. You will be able to test the apps as a fulfiller in the workspace on the app that's been deployed to production.

Use case:

Let's say that a user is in the Creator Studio Users group, so when that user builds an app, that user gets delegated development permissions for that app. That user can then publish a request form, and if there are no roles required for the form, that user can submit requests with the form.

However, that user won't be able to fulfill requests or access the Request App Workspace because that user won't have the x_acme_user_app.agent role, and that user can't give that role to themself. Administrators must assign additional roles as necessary.

Components installed with Creator Studio

When you activate the Creator Studio plugin, various components like tables and user roles are automatically installed.

Note: The Application Files table lists the components that are installed with this application. For instructions on how to access this table, see Find components installed with an application.

What plugins are required to activate Creator Studio?

You must install the following plugins for Creator Studio:

- Creator Studio [sn_creatorstudio]
- Creator Studio - Global [com.glide.creator_studio.global]

Properties installed with Creator Studio

com.glide.creator_studio.template_deny_list

Hides App Engine Studio template from appearing for users when they create an app.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Type: string
- Default value: 211f71ca73202010ae42d31ee2f6a785
- Hide additional templates by adding their system values (sys_id for the template), separated by commas, in the **Value** field. For more information on the sys_id, see Unique record identifier (sys_id).

For more information, see Show the AES template.

sn_creatorstudio.history_record_limit

Limits the last accessed history, for example in recent apps and sorting on the Creator Studio home page.

- Type: integer
- Default value: 1000
- Set this value to -1 to disable the limit
- Reduce this value from the default if the home page isn't loading fast enough

Roles installed with Creator Studio

The following table lists all roles installed with Creator Studio. For details on how the roles work with Creator Studio, see Creator Studio roles and personas.

Roles installed with Creator Studio

Role	Description	Contains role
sn_creatorstudio.user	• Provides access to Creator Studio • Users can create new apps	• catalog_builder_editor • sn_collab_request.basic_write • sn_udc.basic_read • sn_creatorstudio.basic_read • sn_collab_request.basic_read • app_template_runner • sn_g_app_creator.app_creator
sn_creatorstudio.restricted_user	• Provides access to Creator Studio • Users can't create new apps	• catalog_builder_editor • sn_creatorstudio.app_requestor • sn_collab_request.basic_write • sn_udc.basic_read • sn_creatorstudio.basic_read • sn_collab_request.basic_read • app_template_runner
sn_creatorstudio.basic_read	Primitive read access to Creator Studio resources	
sn_creatorstudio.basic_write	Primitive write access to Creator Studio resources	

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Roles installed with Creator Studio (continued)

Role	Description	Contains role
sn_creatorstudio.admin	Admin write access to Creator Studio **Note:** App Engine admins (app_engine_admin) with this role are not restricted from accessing ServiceNow Studio in the experience switcher.	sn_creatorstudio.admin_write
sn_creatorstudio.admin_write	Admin write access to Creator Studio	• sn_creatorstudio.basic_read • sn_creatorstudio.basic_write
sn_creatorstudio.app_requestor	Primitive app requester access to Creator Studio resources	
sn_creatorstudio.basic_fulfiller	Gives some fulfillers the following: • Access to the Request App Workspace • Ability to edit some fields in the sn_creatorstudio_task table	canvas_user

Tables installed with Creator Studio

Tables installed

Table	Description
Request Subtask [sn_creatorstudio_child_task]	Extends the Task table.
New Application Admin Task [sn_creatorstudio_new_application_admin_task]	Extends the Task table.
New Application Task [sn_creatorstudio_new_application_task]	Extends the Task table.
Request App Config [sn_creatorstudio_request_app_config]	Extends the Application File table.
Request Task [sn_creatorstudio_task]	Table where all requests from apps made in Creator Studio are stored.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Tables installed (continued)

Table	Description
	Note: • You must have the Request Task table installed on the production instance as well as the non-production instance. • Admins can change the associated table that each app in Creator Studio uses. However, the tables in this topics are still installed by default. For more information, see Administering an app's associated table. • You can change the **Request type** field, which specifies the form, on the Request Task table or a table that extends Request Task. To do so, you must be an admin or have the sn_creatorstudio.configuration_admin or sn_creatorstudio.task_admin role.
Creator Studio Activities [sn_creatorstudio_activity]	Table where all standard and custom activities for Creator Studio automations are stored. **Note:** This table is readable by Creator Studio users and delegated developers.

Configure Creator Studio using Guided Setup

So you've installed Creator Studio on an instance. Now what? You must configure it before users can start building apps.

Before you begin

Creator Studio must be installed on the instance before you can configure it. Find out more about that in Installing Creator Studio from the ServiceNow Store.

To ensure that forms appear correctly for users, the non-production and production instances must have the same Service Catalog and all of its categories.

Role required: admin

About this task

Guided Setup streamlines configuring Creator Studio, and keeps track of what you have completed, so you can stop and start again where you left off.

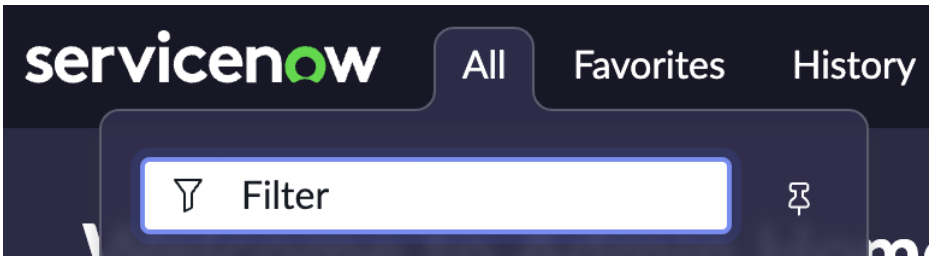
In this Guided Setup, you'll configure user access and collaboration settings to control who can do what in Creator Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

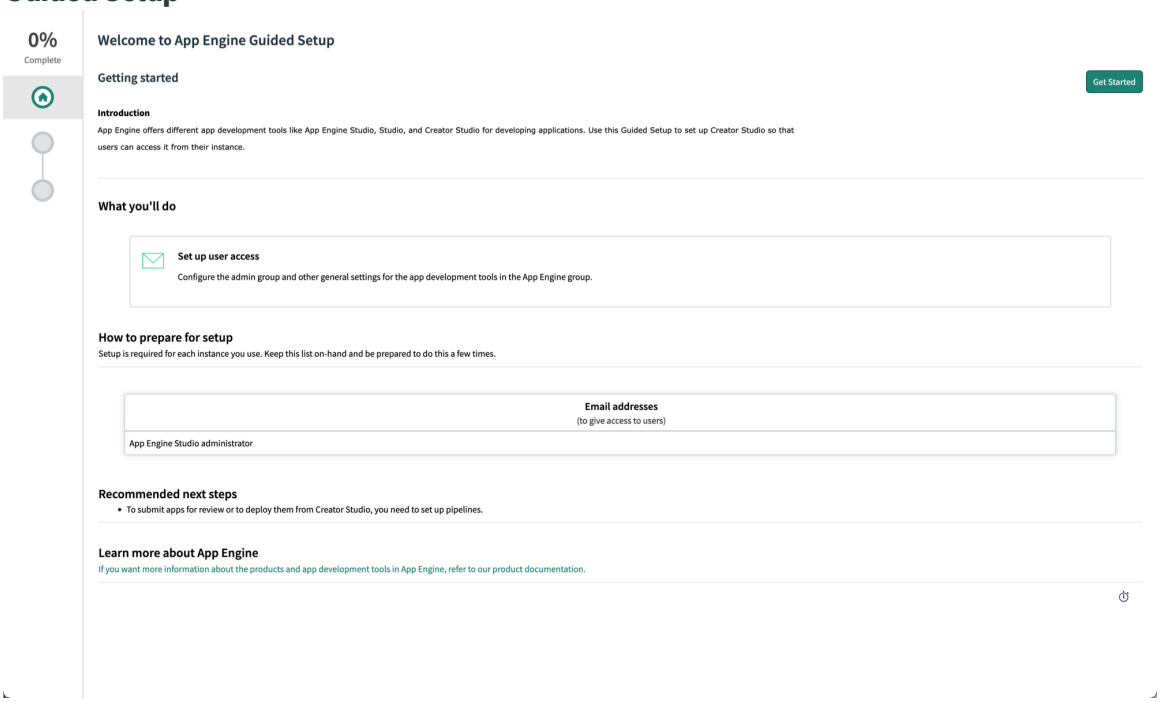
1. Let's open Guided Setup. On your instance, select the **All** tab.

All tab



2. In the white text field, enter `App Engine`.
   The list of apps displayed underneath the text box now pertains only to App Engine.
3. In the list, select **Guided Setup – Shared**, which starts the configuration process.
   The Guided Setup app opens, where you make some basic configurations.

Guided Setup



4. Start the setup process by selecting the **Get Started** button.
5. Select the **Welcome to App Engine Guided Setup** breadcrumb link to return to the main Guided Setup page.

What to do next

Next, it's time to configure specific Creator Studio features, which include the following:

- Set up the Creator Studio admin group
- Set up Creator Studio collaboration descriptors
- Set up Creator Studio full access users
- Set up Creator Studio restricted access users
- Set up Creator Studio playbook activities

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Configure the table for Creator Studio apps
- Set up Creator Studio custom notification emails

What to do next

After you finish configuring Creator Studio, you can configure Pipelines and Deployments to enable apps built in Creator Studio to be deployed to production. For information on how, see Configure Pipelines and Deployments.

Set up the Creator Studio admin group

Set up the admin group to enable ServiceNow administrators access to work on and configure Creator Studio.

Before you begin

Initiate the Guided Setup as described in Configure Creator Studio using Guided Setup.

Before you begin

Role required: admin

Procedure

1. Select the **Get Started** button in the Set up user access section.
2. Select the **Configure** button for the Set up the admin group section.
3. Make any changes to the App Engine Admins group.
   App Engine Admins approve tasks for app development, such as when restricted users need an app created for them.
4. Update the record to save your changes.
5. Select the **Mark as complete** button on the Guided Setup page to indicate that you're done with this step.

What to do next

Next, Set up Creator Studio collaboration descriptors.

Set up Creator Studio collaboration descriptors

Set up collaboration descriptors to manage what app Owner and Editor collaboration types can do in Creator Studio.

Before you begin

Initiate the Guided Setup as described in Configure Creator Studio using Guided Setup.

Role required: admin

Procedure

1. Select the **Configure** button for the Set up collaboration descriptors to manage your user's capabilities section.
   By default, owners can do anything on an app, while editors are more restricted in what they can do. For more information, see Collaborating with others to build apps in Creator Studio.
2. Customize the collaboration roles or add new ones.
   For more information, see Application collaboration.
3. Update the record to save your changes.
4. Select the **Mark as complete** button on the Guided Setup page to indicate that you're done with this step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

What to do next
Next, Set up Creator Studio full access users.

Set up Creator Studio full access users

Set up who has full access to Creator Studio to decide which users get full access to work on app in Creator Studio, and who has more limited capabilities.

About this task
For example, restricted users must request that an admin create a new app for them. For more information, see Creator Studio roles and personas.

Before you begin
Initiate the Guided Setup as described in Configure Creator Studio using Guided Setup.

Role required: admin

Procedure
1. Select the **Get Started** button in the Set up access to Creator Studio section.
2. Set up full access for users by selecting the **Configure** button for the Grant full Creator Studio access to users section.
3. Add users to the Creator Studio Users Group.
   People in the Creator Studio Users Group have full access to working with apps, and can create apps without needing help from an admin.
4. Update the record to save your changes.
5. Select the **Mark as complete** button on the Guided Setup page to indicate that you're done with this step.

What to do next
Next, Set up Creator Studio restricted access users.

Set up Creator Studio restricted access users

Set up limited access for Creator Studio users. People in the Creator Studio Restricted Users Group have limited access to working with apps, and must request that admins create apps for them.

Before you begin
Initiate the Guided Setup as described in Configure Creator Studio using Guided Setup.

Role required: admin

Procedure
1. Navigate to **All** > > .
2. Select the **Configure** button for the Grant Creator Studio Restricted access to users section.
3. Add users to the Creator Studio Restricted Users Group.
4. Update the record to save your changes.
5. Select the **Mark as complete** button on the Guided Setup page to indicate that you're done with this step.

Set up Creator Studio custom notification emails

Configure customized email notifications that Creator Studio apps send. For example, you can add branding to notification emails that a request was opened.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

About this task
Customized email templates can be applied to notifications that are sent when users request something from an app, or the request was changed or closed.

Before you begin
Initiate the Guided Setup as described in Configure Creator Studio using Guided Setup.

Role required: admin

Procedure
1. Select the **Configure** button for the Configure Email Notifications Creator Studio Requests section.
2. Add users to the Creator Studio Restricted Users Group.
   A table displays all email notifications for Creator Studio.
3. Select the **Name** of the email that you want to customize.
   For example, **Request opened**.

   Note: Depending on your scope, you may need to switch to the Creator Studio Configurations scope to make changes.

4. Make any updates and customizations to the content of the email.
   You can select into any of the following tabs to edit their details:
   a. **When to send**
   b. **Who will receive**
   c. **What it will contain**

   For more information on editing notifications on the ServiceNow AI Platform, see Create an email notification.
5. Update the record to save your changes.
6. Select the **Mark as complete** button on the Guided Setup page to indicate that you're done with this step.

Set up Creator Studio playbook activities
Configure the activities that appear when users build playbooks in Creator Studio.

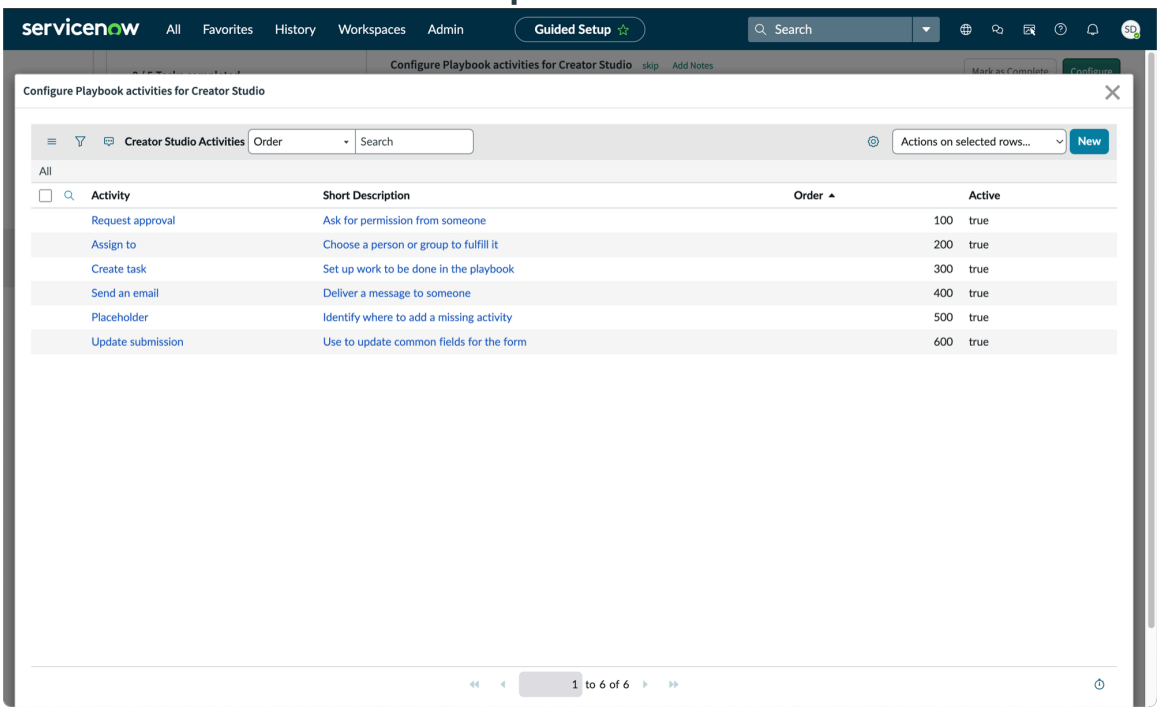
Before you begin
Initiate the Guided Setup as described in Configure Creator Studio using Guided Setup.

Role required: admin

Procedure
1. Select the **Configure** button for the Configure Playbook activities for Creator Studio section.
   All of the default Creator Studio playbook activities appear.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

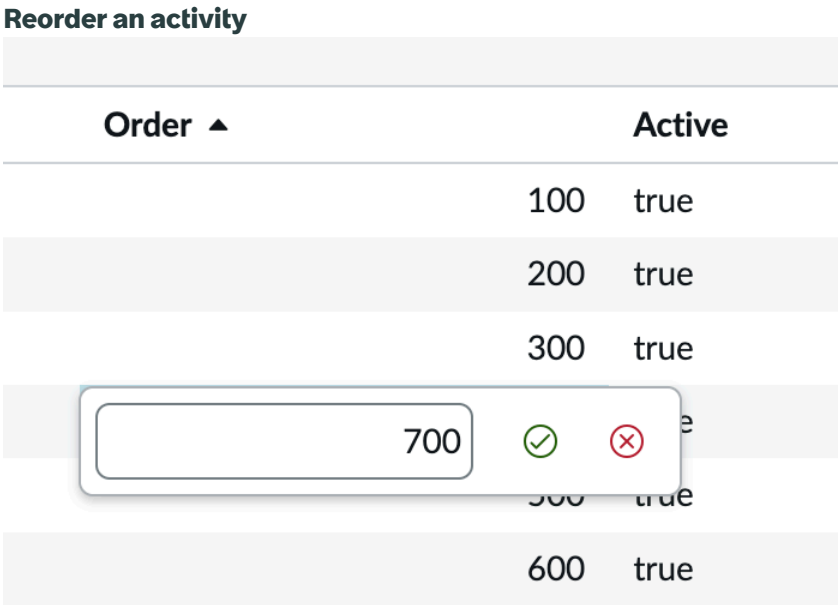
Creator Studio activities in Guided Setup



2. **Optional:** Rearrange the order activities appear in the selection list.
   Order numbers are relative to the other activities, which you can view on the Configure Playbook activities for Creator Studio landing page in Guided Setup.

   a. Select into the **Order** column for the activity you want to renumber.

   b. Enter a new order number for the activity.



   c. Select the Save icon or press Enter.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

3. **Optional:** Add a new activity.

   Note: The activity must already be configured in the activities table for you to add it to the Creator Studio playbook list.

   a. Select the **New** button.
   b. Search for the activity you want to add in the **Activity** field.
   c. Enter the **Order** it appears in.
   d. Enter a **Short Description**, which will appear in the activity picker under the activity name.
   e. Leave the **Active** check box selected to make the activity visible immediately.
   f. Select the **Submit** button.

What to do next
Users who are actively working in Creator Studio must refresh their browser to see the changes you just made.

Configure the table for Creator Studio apps

Use Guided Setup to change the table that the app saves its requests to.

About this task
Some general guidelines for changing the table are:

- If you change an app's table to one that doesn't extend a Request Task-extended table, it could affect automations.
- If the new table doesn't have the **request_type** field, the app's automations won't be correctly triggered.

  Note: You can change the **Request type** field, which specifies the form, on the Request Task table or a table that extends Request Task. To do so, you must be an admin or have the sn_creatorstudio.configuration_admin or sn_creatorstudio.task_admin role.

- The request_type field for the new table should have the label **Request type**, and it should be a reference to the Record Producer table.
- If the new table isn't in the same scope as the app, the scope of the table must allow updates from other scopes.

Before you begin
Initiate the Guided Setup as described in Configure Creator Studio using Guided Setup.

The app must already be created before you can change the table for it.

Before you begin
For more information on changing an app's table, see Changing the table for an app.

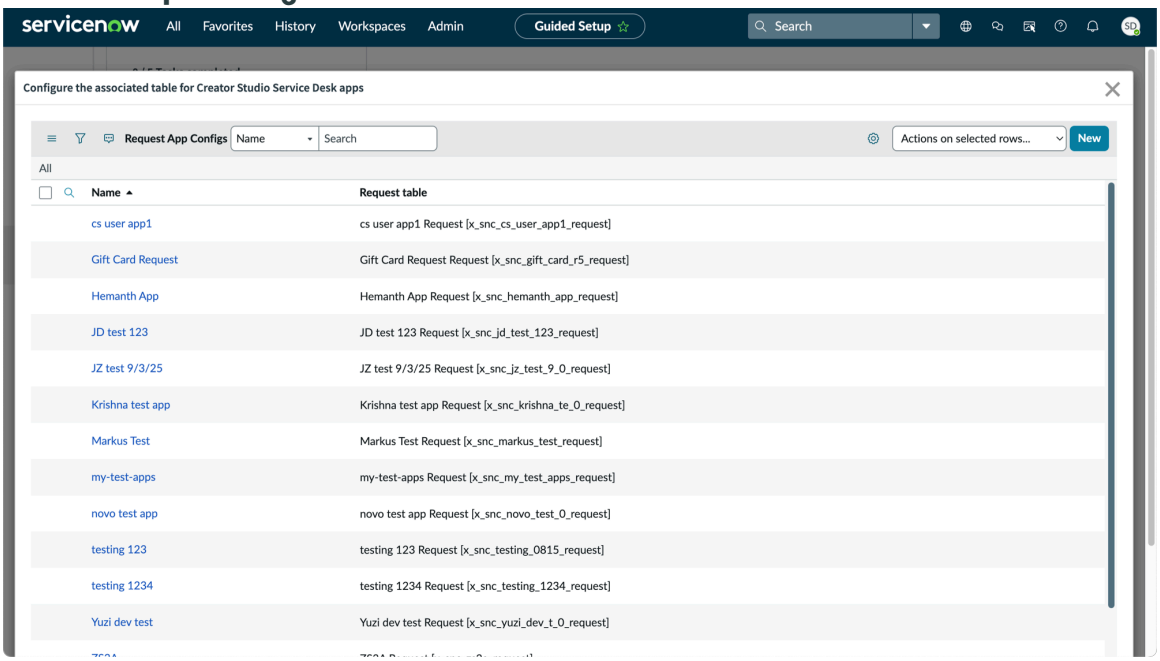
Role required: admin, app configurator (sn_creatorstudio.app_configurator), or Creator Studio configuration admin (sn_creatorstudio.configuration_admin). For more information on working with granular roles, see Granular admin roles.

Procedure

1. Select the **Configure** button for the Configure the associated table for Creator Studio Service Desk apps section.
   All of the apps built on the instance in Creator Studio appear.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Guided Setup to configure the associated table



2. Select the app that you want to change the table for.

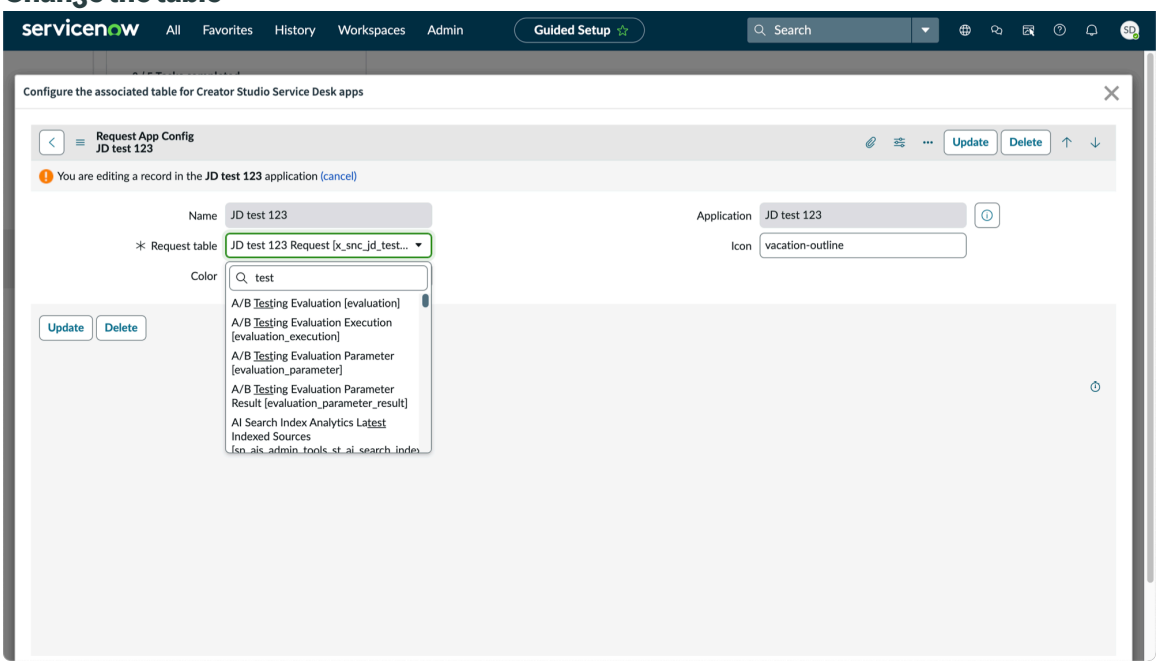
> **Note:** You may need to change the scope that you're in.

3. Change the table.

   a. Select into the **Request table** field.

   b. Search for and select the table that you want to associate the app with.

Change the table



4. Select the **Update** button.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Result
The associated table for an app changes to the new table. Any users currently in Creator Studio may need to refresh their browser to see the change.

Configuring Pipelines and Deployments to deploy apps built in Creator Studio

You can install Pipelines and Deployments and configure a controller instance to deploy apps built in Creator Studio to production instances.

For details on how to set up Pipelines and Deployments, see Configure Pipelines and Deployments.

Creator Studio roles and personas

Roles control what everyone you work with can do in Creator Studio. Administrators assign roles to give team members permission to configure or use Creator Studio.

The two roles for Creator Studio are used to restrict access from creating new apps, which helps make sure your instance isn't overfilled with redundant, unplanned, or unused apps.

Personas that use Creator Studio

Personas aren't explicitly part of Creator Studio, but administrators assign roles to give team members permission to configure or use Creator Studio.

Low-code/citizen developer

Low-code/citizen developers are tech savvy and interested in creating apps. Though they might not have formal coding or app development training, citizen developers can submit ideas for new apps and, if approved, build them using Creator Studio.

Low-code/citizen developers have either the sn_creatorstudio.user or sn_creatorstudio.restricted_user role.

App Engine admin

App Engine admins manage all processes related to app development in Creator Studio. They review new app ideas, handle app deployment, and manage collaborators, usually in the App Engine Management Center.

App Engine admins have the app_engine_admin role and must be in the app_engine_admin group.

Security admin

The security admin creates and modifies roles and access control lists for apps. This role is set on the platform level, and it is required for making updates to roles in Creator Studio.

System administrator

The system administrator has access to all system features, functions, and data, regardless of security constraints. Grant this privilege carefully. If you have sensitive information, such as HR records, that you must protect, create a custom admin role for that area and train a person who is authorized to see those records to act as the administrator.

Roles and what they can do in Creator Studio

In addition to the roles in the following table, users with the admin and delegated_developer roles can also access Creator Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For complete details on which roles each role contains, see Components installed with Creator Studio.

Creator Studio roles

Role	Name	Description
Creator Studio User	sn_creatorstudio.user	• Users can create apps in Creator Studio. • The user is automatically delegated as the app owner. For more information, see Delegated development and deployment. • Contains sn_g_app_creator.app_creator. **Note:** This role gets assigned the delegated_developer role when they create or get access to an app.
Creator Studio Restricted User	sn_creatorstudio.restricted_user	• Users can't create apps in Creator Studio. • Users can request apps to be created for them, and to work on an app. • Users can work on apps that they've been designated as developers for. • When assigned to work on an app, this user gets the delegated_developer role for that app.
App Engine Admin	app_engine_admin	• Approve requests from restricted users to create an app. • Approve collaboration requests. • Select other development environments from the experience switcher. • Contains sn_creator_studio.admin_write and

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Creator Studio roles (continued)

Role	Name	Description
		sn_creator_studio.basic_write to enable admins to see the apps they need to approve.
ServiceNow Otto for Creator	now.assist.creator	Grants users access to AI skills to create forms in Creator Studio.
Creator Studio configuration admin	sn_creatorstudio.configuration_admin	Granular admin role that contains the following Creator Studio granular admin roles: • sn_creatorstudio.task_admin • sn_creatorstudio.app_configurator • sn_creatorstudio.reports_viewer For more information on working with granular roles, see Granular admin roles.
Task admin	sn_creatorstudio.task_admin	Granular admin role that grants users access to change several fields on the Request Task table or a table that extends Request Task. This role contains the following: • Table-level access for sn_creatorstudio_task: Create, Write, Delete • Field-level access for: ○ sn_creatorstudio_task.request_type, which enables you to change the associated form ○ sn_creatorstudio_child_task.parent, which enables you to change the parent table for any subtask tables created from a task activity added to an app's playbook.
App configurator	sn_creatorstudio.app_configurator	Granular admin role that grants users access to change the associated table for an app built in Creator Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Creator Studio roles (continued)

Role	Name	Description
Reports viewer	sn_creatorstudio.reports_viewer	Granular admin role that grants users access to run reports on tables.

Note:

To ensure that users can use the Collaboration Approval Workflow regardless of instance versions, admins must assign the catalog _builder_editor role to Creator Studio user groups.

User groups and what they can do in Creator Studio

Groups are a standard functionality that help you quickly control people's access to Creator Studio by adding them to a group.

Creator studio user groups

Group	Description
Creator Studio Users	• Users are automatically approved to create apps in Creator Studio. • Contains sn_creatorstudio.user.
Creator Studio Restricted Users	• Users in this group need to request applications be created in Creator Studio on their behalf. • Contains sn_creatorstudio.restricted_user.

Developer roles and testing apps on instances

If you have a Creator Studio role of sn_creatorstudio.user or sn_creatorstudio.restricted_user, you won't be able to test the apps you build on the non-production instance's Request App Workspace. You should be able to test the app on the non-production instance using Creator Studio's app previews. You will be able to test the apps as a fulfiller in the workspace on the app that's been deployed to production.

Use case:

Let's say that a user is in the Creator Studio Users group, so when that user builds an app, that user gets delegated development permissions for that app. That user can then publish a request form, and if there are no roles required for the form, that user can submit requests with the form.

However, that user won't be able to fulfill requests or access the Request App Workspace because that user won't have the x_acme_user_app.agent role, and that user can't give that role to themself. Administrators must assign additional roles as necessary.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Collaboration roles and instances on different versions

As admins implement Creator Studio, they may have it installed on a non-production instance while their production instance is on a previous version of the ServiceNow AI Platform that doesn't have Creator Studio. This mis-match of instance versions affects the Collaboration Approval Workflow, which specifies the non-production instance as the source and the production instance as the controller. If the controller doesn't have the version of the collaboration plugin that supports Creator Studio, collaboration is unsupported.

To ensure that users can use the Collaboration Approval Workflow regardless of instance versions, admins must assign the catalog _builder_editor role to Creator Studio user groups.

Roles and app development collaboration

Roles define user access to Creator Studio. Permission to work on individual apps is controlled on an app-by-app basis. That is, you must manage the collaborators for each app by inviting other citizen developers to work on the app with you, or request to join someone else's app. For more information, see Collaborating with others to build apps in Creator Studio.

Creator Studio and domain separation

Domain separation is not supported in Creator Studio. This means that you can't separate data, processes, and administrative tasks into distinct groups, called domains, within the ServiceNow AI Platform.

No support for domain separation

Though there may be a domain specified on data tables that Creator Studio uses, Creator Studio doesn't provide any special handling of data based on domains.

For more information on support levels, see Application support for domain separation.

Related topics

- Domain separation for service providers

Configuring form generation in Creator Studio

Install and configure ServiceNow Otto for Creator in Creator Studio.

Configuration overview

Along with the installation of ServiceNow Otto for Creator in Creator Studio application, you must enable the Catalog item generation skill and the app generation skill.

Related topics

- Install plugins for ServiceNow Otto
- Activate an AI skill

Install ServiceNow Otto for Creator for form generation in Creator Studio

Install the ServiceNow Otto for Creator application from the ServiceNow® Store to get AI in Creator Studio.

Before you begin

Review the ServiceNow Otto for Creator application listing in the ServiceNow Store to learn about dependencies, licensing, and subscription requirements, and release compatibility. ServiceNow Otto for Creator installs AI for form generation in Creator Studio.

Role required: admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to the ServiceNow Otto for Creator application on the ServiceNow Store .
2. On the ServiceNow Otto for Creator application page, select **Request App**.
3. After your request is approved, navigate to **All** > **System Applications** > **All Available Applications** > **All**.
4. Find the ServiceNow Otto for Creator application (sn_now_creator) by using the filter criteria and search bar.
5. Verify that ServiceNow Otto for Creator is installed:

   a. Navigate to **All** > *AI Admin Hub* > **Features**.

   b. In the workflow list, select **Creator**.

   c. On the Service Catalog card, verify that the catalog item generation skill is active.

   > **Note:** If the Service Catalog card displays **Not started** or **Inactive** for the catalog item generation skill, you must activate it. To learn more, see Activate an AI skill.

   For more information about using the AI Admin Hub to access information about setting up, configuring, and monitoring ServiceNow Otto applications, see Now Assist Admin console.

What to do next

Grant the now.assist.creator role to the Creator Studio users to create forms using AI in Creator Studio. To learn more, see Assign a role to a user and Creator Studio roles and personas.

Administering Creator Studio

Admins have special privileges to approve and update tasks in Creator Studio that regular users can't.

As a Creator Studio administrator, you manage the processes that keep app development running smoothly on your instance. Admin tasks include approving app creation requests, managing collaborators, configuring forms and activities, and deploying apps to production.

Some of the things that admins can do for Creator Studio include the following:

- Create apps for people who don't have permission.
- Administer who can access the experience switcher.
- Manage collaboration requests.
- Create and edit templates for forms.
- Change the table an app uses.
- Configure custom activities for playbooks.
- Administer the App type page.
- Deploy apps to production.
- Configure user access to deployed apps.

Approve Creator Studio requests to create an app

Respond to app creation requests made by users who can't create apps themselves, ensuring that your organization doesn't have too many or redundant apps.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin
Role required: app_engine_admin

About this task
People with the Creator Studio Restricted User role aren't allowed to create apps themselves, but they can work on them after you approve their creation in the Application Tasks table of Request App Administration.

If your organization has App Engine Management Center, you can approve requests there, too. For more information, see Manage app requests from Creator Studio.

Procedure
1. Navigate to **All** > **App Engine** > **Request App Administration** > **Application Tasks**.
   You can also select the link in any emails you receive notifying you of app creation requests.
2. Find the request you want to approve or decline.
   Filter on columns or use the search bar if you need help finding requests.
3. Select the **Number** of the request to review.
4. Select the **Approve** or **Reject** button to approve or decline the app creation request.

Result
The user who made the request receives an email notification with the approval result. If you approved, they can start building out their app in Creator Studio!

Administering access to the experience switcher

The experience switcher can provide access to Creator Studio, ServiceNow Studio, ServiceNow IDE. However, whether you can see and select all of those depends on your role or access level.

There are several factors that determine what you can choose from the experience switcher: your role, whether the selected studio is installed on the instance, and which version is installed.

Roles determine access

What role you have, such as user group and security levels, enables access to products in the experience switcher.

The following describes default role access to the experience switcher:

- Admins and delegated developers can use the experience switcher because they may need access to any product where they've been delegated to administer or develop an app.
- Creator Studio users and Creator Studio restricted users don't generally have access to the experience switcher because administrators limit them to a more curated experience.

Admins can check the Experience Configurations table [sn_udc_experience_configuration] to see which default roles have access to each of the products in the experience switcher. The table also shows which roles will see the discovery page that appears if Creator Studio isn't installed, or isn't the minimum version needed for the experience switcher.

If you want to give more roles access to products in the experience switcher, you must do so in the Experience Visibility Controls table [sn_udc_experience_visibility_control]. For more information, see Edit a role's access to the experience switcher.

Product installation determines access

You must have an application installed on the instance to select it from the experience switcher.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For example, if you're working in ServiceNow Studio or ServiceNow IDE but don't have Creator Studio installed, selecting Creator Studio in the experience switcher displays a page where you can find out more on using Creator Studio.

Product version determines access

To appear in the experience switcher, all products must be on the Yokohama version at a minimum.

For example, if you choose Creator Studio in the experience switcher but have the Xanadu version of Creator Studio installed, selecting it displays a page directing you to update the version.

Edit a role's access to the experience switcher

Add more roles to the Experience Visibility Controls table [sn_udc_experience_visibility_control] to give them access to other products in the experience switcher, such as ServiceNow Studio.

Before you begin

Role required: admin

About this task

This topic is for giving non-default roles to the experience switcher. The following roles are already assigned by default:

- Admins and delegated developers can use the experience switcher because they may need access to any product where they've been delegated to administer or develop an app.
- Creator Studio users and Creator Studio restricted users don't generally have access to the experience switcher because administrators limit them to a more curated experience.

Procedure

1. Navigate to **All** > **sn_udc_experience_visibility_control.list**.
2. Select the **New** button.
3. Enter the experience that you're configuring roles for.
   For example, start typing `cre` to select Creator Studio.
4. Specify the roles that should get access to the experience.
   a. Unlock the **Available for roles** field by selecting the lock icon .
   b. Start typing and then select the name of the role you're giving access to.
   c. Lock the **Available for roles** field by selecting the unlock icon .
5. Specify the roles that should not have access to the experience.
   a. Unlock the **Not available for roles** field by selecting the lock icon .
   b. Start typing and then select the name of the role you're restricting access for.
   c. Lock the **Not available for roles** field by selecting the unlock icon .
6. Select **Submit** to save the changes.

Administering collaboration in Creator Studio

In Creator Studio, as an administrator or App Engine admin, you have the power to manage how people work together on projects.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

You can approve collaboration requests, enabling delegated developers to work together on an app.

Collaboration descriptors are like permission levels that determine what someone can do in an app. Admins can assign and customize collaboration descriptors.

The following collaboration descriptors are available for Creator Studio:

Collaboration descriptions

Permission	Description	What they can do in the app
Manage collaborators	You can control who has access to the app and what they can do.	• Change permissions for users and groups. • Remove people from working on the app.
Invite collaborators	You can bring in new people to work on the app.	• Invite users and groups to join the app. • Invited people to become editors of the app, meaning they can make changes.

Approve a collaboration request

Admins and App Engine admin can review and approve (or reject) people's requests to join an app's development in Creator Studio.

Before you begin

If you have a controller instance configured, you must approve collaboration requests on your controller instance, not the non-production instance with Creator Studio. The controller instance is typically the production instance, but it can vary depending on how your Pipelines and Deployments application is configured. For more information, see Configure your controller instance.

Role required: app_engine_admin

About this task

Note: Users with a delegated developer role are automatically approved for collaboration.

Procedure

1. Navigate to **All** > *App Engine* > **Collaboration** > **Collaboration Tasks**.
2. Select the collaboration request in the **Number** column of the collaboration table.
   You can search or filter on the **App name** or **Invitee name** columns to find the request faster.
3. Update the approval status in the **State** field.
   - **Closed – Approved**
   - **Closed – Rejected**
   - **Canceled**
   - **Closed – Failed**
4. Update the record to save your changes.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Manage Creator Studio collaboration permissions on the ServiceNow AI Platform

In Creator Studio, you can control how people work together on apps by setting up permissions called collaboration descriptors. These descriptors define what users can and cannot do.

Before you begin

Role required: app_engine_admin, and permission to Manage collaborators

About this task

Users with the Manage collaborators permission can adjust who can collaborate on the apps they own. For more information, check Manage collaborators for an app in Creator Studio.

Note: The way that collaboration permissions appear may vary depending on whether your instance has App Engine Studio installed.

Procedure

1. Navigate to **All** > *App Engine* > **Collaboration** > **Descriptors**.
2. Edit the permissions for a collaboration descriptor, such as owner or editor.
   a. Find and select the **Name** of the descriptor you want to modify.
   b. Choose a **Development permission set**, such as **Service Catalog**, to update its permissions. For a list of permissions, see Collaboration permissions.
   c. Select **Save** and **Update**.
3. Assign the updated descriptor to users or groups.
   a. Select the descriptor **Name**, such as **Owner** or **Editor**.
   b. Select the applicable tab to assign either users or groups:
      - **App Collaboration Users** tab: For individual users
      - **App Collaboration Groups** tab: For groups of users
   c. Select the **New** button.
   d. On the form, fill in the fields.
      For more information, see the following topics:
      - Assign collaboration descriptors to users
      - Assign collaboration descriptors to groups
   e. Select **Save**.
4. **Optional:** If you need a new collaboration descriptor, create a custom one.
   a. Select the **New** button.
   b. On the form, fill in the fields.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

App Collaboration Descriptor form

Field	Description
Name	Name for the descriptor.
Description	Description of what the descriptor can do. Examples are as follows: ▪ Owner ▪ Editor
Application	Scope of the collaboration descriptor. Currently, all collaboration descriptors must be created in the global scope.
Standard	Option for inviting other collaborators with this role.

c. Select **Submit**.

Result

You've now set up collaboration descriptors, defined permissions, and assigned roles to users and groups for specific apps. For complete details on using the Collaboration app, see Application collaboration.

Customized app collaboration permissions in Creator Studio

When you customize collaboration permissions, you can choose more granular actions and parts of the app that users can work with in Creator Studio.

Use these permissions when managing collaboration, either as an admin or app owner. For more information, see the following topics:

- Manage Creator Studio collaboration permissions on the ServiceNow AI Platform
- Manage collaborators for an app in Creator Studio

File types custom collaboration permissions

File type collaboration permissions

Permission	Description	Owner default setting	Editor default setting
All File Types	Grants access to work with all types of files, including additional file types not granted by the other options.	Yes	Yes
Integrations	Grants access to web service APIs, REST APIs, data sources, and Integration Hub - Import.	Yes	Yes
Reporting	Grants access to reports and scheduled reports.	Yes	Yes

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

File type collaboration permissions (continued)

Permission	Description	Owner default setting	Editor default setting
Mobile builders	Grants access to build mobile experiences, such as with Mobile App Builder.	Yes	Yes
UI Builder	Grants access to work with UI Builder to build out more complex interfaces.	Yes	Yes
Workflow	Grants access to Workflow Editor and Activity Creator.	Yes	Yes
Service Portal	Grants access to work with Service Portal editors and tools.	Yes	Yes
Workflow Studio	Grants access to the Workflow Studio design environment to create flows and actions.	Yes	Yes
Service Catalog	Grants access to work with catalog-related file types such as catalog items, record producers, and variables to add catalog items to apps.	Yes	Yes
Tables & forms	Grants access to model and layout-related file types such as table columns, form layout, and list layout.	Yes	Yes
Decision Tables	Grants access to work with decision tables to create decision logic based on multiple if-else rules.	Yes	Yes
Playbooks	Grants access to work with the Playbooks design environment to create processes. Editing activity subflows or actions requires the **Flow Designer** permission.	Yes	Yes
Notifications	Grants access to create automatic email notifications	Yes	Yes

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

File type collaboration permissions (continued)

Permission	Description	Owner default setting	Editor default setting
	and work with email templates.		

Security/Entitlement custom collaboration permissions

Manage access control grants access to security management files, such as Access Control Lists and roles.

The default setting for both owners and editors is selected.

Programming tools custom collaboration permissions

Allow scripting grants access to script fields, such as those in Business Rule, UI Action, and Client Script.

The default setting for both owners and editors is de-selected.

Application management custom collaboration permissions

Application management collaboration permissions

Permission	Description	Owner default setting	Editor default setting
Delete application	Grants access to delete the app.	Yes	No
Manage collaborators	Grants access to change the developers and their permissions to collaborate on an app.	Yes	No
Source control	Grants access to work with source control tool integrations.	No	No
Invite collaborators	Grants access to invite developers to collaborate on an app.	Yes	Yes

Deployment custom collaboration permissions

Deployment collaboration permissions

Permission	Description	Owner default setting	Editor default setting
Upgrade app	Grants access to upgrade applications.	No	No
Submit for deployment	Grants access to request deployment for an app.	Yes	No

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Deployment collaboration permissions (continued)

Permission	Description	Owner default setting	Editor default setting
Publish app to repo	Grants access to publish the app to your repo.	No	No
Publish to app store	Grants access to publish the app to your app store.	No	No

Administering templates and forms for Creator Studio

As an admin, you can create custom templates and pre-configured questions to help streamline and guide users when they build apps in Creator Studio.

You can also choose to display the App Engine Studio template to users.

Check out an article on Creating catalog templates for Creator Studio.

Creating catalog templates for use in Creator Studio apps

As a ServiceNow admin, you can create customized catalog templates in Service Catalog to guide users through adding forms in Creator Studio.

For example, if your users will be building several apps for IT fulfilment, you may want to create a catalog template that contains only IT service requests.

When you add a catalog template to help users create forms in Creator Studio, you must use the following configurations:

- Create a record producer template.
- Leave the **Destination table** (which is called the **Record submission table in Creator Studio**) as undefined.
- Ensure that the Creator Studio Users group can access the template. The template can also be available to all users.

Additionally, you may want to use the catalog template to restrict the catalogs that users can choose when creating an app.

For more details on how to add catalog templates, see Creating or editing catalog item template.

Check out an article on Creating catalog templates for Creator Studio.

Creating question sets for use in Creator Studio forms

Admins can create pre-configured question sets that can be reused across multiple forms without being changed in Creator Studio.

Question sets enable admins to create curated, uneditable questions that users can include on forms without worrying about tweaking their content.

Additionally, you can create question sets with multiple questions if you want several reusable questions to always be grouped together.

See Create a variable set and add it to an item to find out how to create a question set in Service Catalog.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Show the AES template

You can make the App Engine Studio (AES) template, which is hidden by default, appear for users when they create an app in Creator Studio.

Before you begin
Role required: admin

Procedure

1. Open the **All** menu.
2. Type `sys_properties.list` and press Enter.
3. Enter *`com.glide.creator_studio.template_deny_list`* in the **Name** column, and then press Enter.
4. Clear the entry from the **Value** field.
   If you want to hide any other templates, enter their system values (sys_id for each template) in the **Value** field, separated by commas. For more information on the sys_id, see Unique record identifier (sys_id).

Administering an app's associated table

Admins can change the default table that apps built in Creator Studio save their requests to.

Note: There are two ways to change an app's associated table:

- You can use Guided Setup, which is an easier, more streamlined process. For more information, see Configure the table for Creator Studio apps.
- You can update several tables on the ServiceNow AI Platform. For more information, see Change a Creator Studio app's table.

Changing the table for an app

Admins can change the associated table for an app built in Creator Studio. That is, you can change the table that the app saves its requests to.

View the current table for the app by selecting the **Data management** tab in the App settings. For more information, see Edit an app's settings in Creator Studio.

Note: There are two ways to change an app's associated table:

- You can use Guided Setup, which is an easier, more streamlined process. For more information, see Configure the table for Creator Studio apps.
- You can update several tables on the ServiceNow AI Platform. For more information, see Change a Creator Studio app's table.

Reasons to change the table for an app

Some reasons that you may want to change the table for an app include:

- You have an existing table that has business logic or handling of specific fields, you can have the app write to that table to use the existing logic.
- You can't extensively modify the Request App table, so you may want to make more complex modifications and use a different table.
- You want to use a table that inherits components from a federated app.
- You already have a large federated application and want to put data from the new Creator Studio into that federated app table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Requirements for changing the table for an app

The app must already be created before you can change the table for it.

A general guideline is to use a table that extends the Request Task table.

- If you change an app's table to one that doesn't extend a Request Task-extended table, it could affect automations.
- If the new table doesn't have the **request_type** field, the app's automations won't be correctly triggered.

  Note: You can change the **Request type** field, which specifies the form, on the Request Task table or a table that extends Request Task. To do so, you must be an admin or have the sn_creatorstudio.configuration_admin or sn_creatorstudio.task_admin role.

- The request_type field for the new table should have the label **Request type**, and it should be a reference to the Record Producer table.
- If the new table isn't in the same scope as the app, the scope of the table must allow updates from other scopes.

For more information on the Task table, see Working with the Task table.

Role for changing the table for an app

To change the associated table for an app, you must have either the admin role, the sn_creatorstudio.app_configurator role, or sn_creatorstudio.configuration_admin, which are granular admin roles. For more information, see Granular admin roles.

Repercussions of changing an app's table

The following table describes how different parts of building an app in Creator Studio are affected by changing the app's table.

How changing the table affects parts of an app

Part of building an app	Effect
Forms	If you change the table for an app after a form is created, users get an error when they view a form that was created against a table that's different from the app's current table. In that case, you should change the table back to the original table, or users should create new forms that use the new table.
Automations	If you change the table to one without the **request_type** field, users can't add a playbook to the app.
Workspace list configurations	If you change the table after a user created a filtered list, the filtered list retains the original table. If multiple filtered lists use different tables, users will get errors based on those discrepancies. For example, they can't

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

How changing the table affects parts of an app (continued)

Part of building an app	Effect
	manage columns for a table that they don't have edit access to.

Change a Creator Studio app's table

Change the table associated with an app if you want one that differs from the default.

Before you begin

Note: You should perform this task only if you did not change the table during Guided Setup. For more information, see Configure the table for Creator Studio apps.

The app must already be created before you can change the table for it.

Some general guidelines for changing the table are:

- If you change an app's table to one that doesn't extend a Request Task-extended table, it could affect automations.
- If the new table doesn't have the **request_type** field, the app's automations won't be correctly triggered.

  Note: You can change the **Request type** field, which specifies the form, on the Request Task table or a table that extends Request Task. To do so, you must be an admin or have the sn_creatorstudio.configuration_admin or sn_creatorstudio.task_admin role.

- The request_type field for the new table should have the label **Request type**, and it should be a reference to the Record Producer table.
- If the new table isn't in the same scope as the app, the scope of the table must allow updates from other scopes.

See Changing the table for an app for more information.

Role required: admin, app configurator (sn_creatorstudio.app_configurator), or Creator Studio configuration admin (sn_creatorstudio.configuration_admin). For more information on working with granular roles, see Granular admin roles.

About this task

To ensure that forms and automations work, a general guideline is to change the table to a table that extends the Request Task table.

Procedure

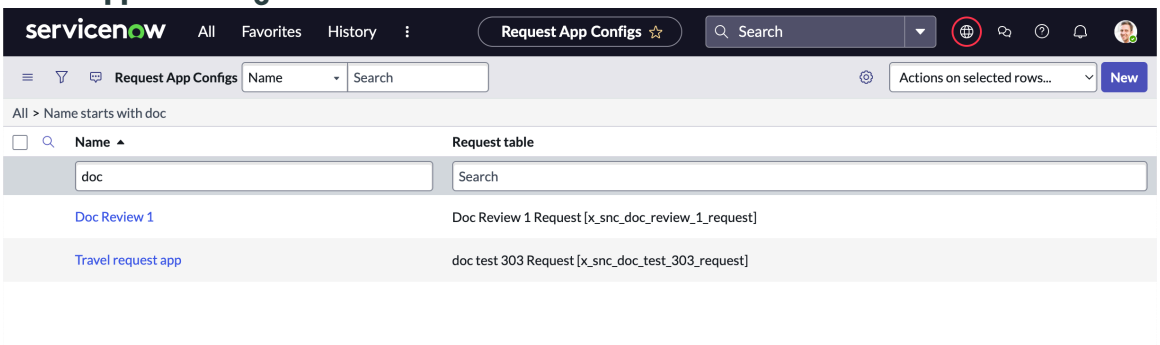
1. Open the **All** menu and enter `sn_creatorstudio_request_app_config.list`.
2. Enter the app name in the search box for the **Name** column and press Enter.

   Note: You must enter the first word of the app name, not subsequent words in the app name.

3. Select the app from the list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

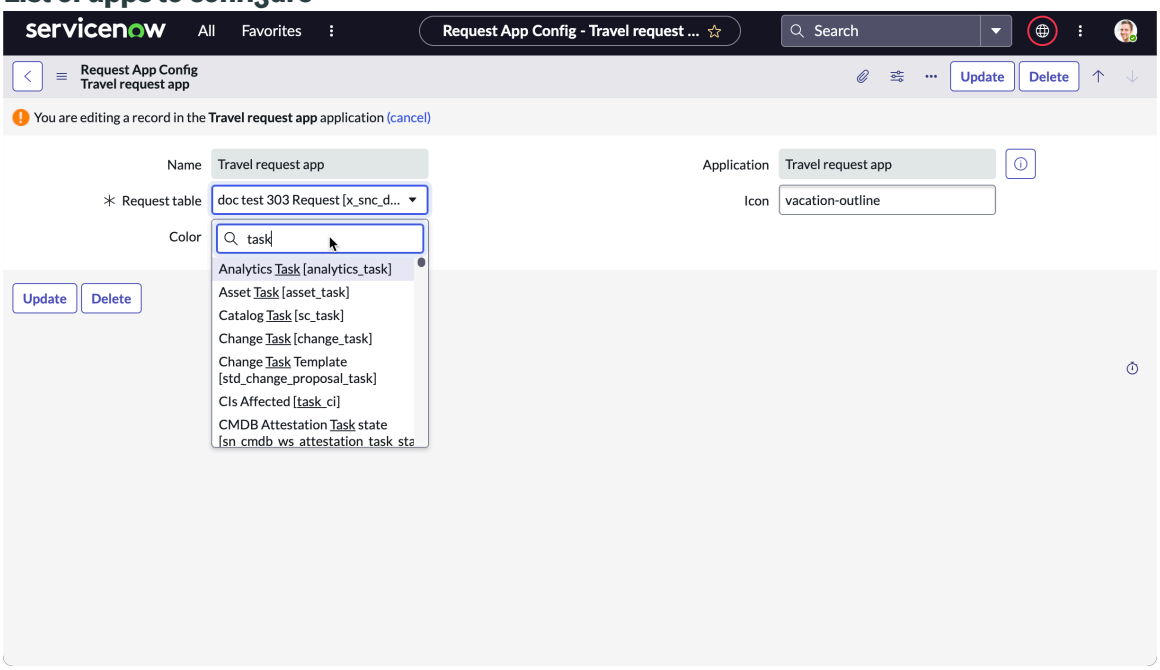
List of apps to configure



4. **Optional:** Change the scope to the app's scope or a global scope if you're not in the correct scope.
   You can change the scope in one of the following ways:
   - Selecting the **here** in the message "To edit the record click here."
   - Selecting the application scope icon and choosing the appropriate scope.

   For more information on scopes, see Application scope.
5. Enter the name of the table you want the app to write to in the **Request table** field.
   The table should extend the Request Task table and have the **request_type** field. For more details on what table you can use, see Requirements for changing the table for an app.

List of apps to configure



6. Select **Update** to save your changes.

Administering notification emails in Creator Studio

Admins can customize existing email notifications and create templates for new email notifications that Creator Studio apps send. For example, you can add branding to notification emails that a request was opened.

Customized email templates can be applied to notifications that are sent when users request something from an app, or the request was changed or closed.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Ways to customize email notifications

Admins can customize notification emails in two ways:

1. Using Guided Setup.
2. Using the [sysevent_email_action] table. For more information on creating and editing notification emails on the ServiceNow AI Platform, see Create an email notification.

Scopes for email notifications

The following default email notifications are in the public scope, and can thus be edited by all admins:

1. Request assigned to me
2. Request assigned to my group
3. Request closed
4. Request opened

Note: Depending on your scope, you may need to switch to the Creator Studio Configurations scope to make changes.

In addition to the default emails, you can create custom email notifications that the ServiceNow AI Platform sends when it runs a playbook.

Administering activities in Creator Studio

As an admin, you can define custom activities, make them available in Creator Studio, and hide existing activities to help guide users when they add automation to apps.

You may want to create custom activities for complex or company-specific workflows. For example:

- The standard Request approval activity has a **Requester's manager** option, but you may need a **Department manager** instead.
- You may need an approval or flow with more complex decision logic than is easily built in the Creator Studio automation. You could use Workflow Studio to build that flow, and then create a custom action that calls it.
- You may want more notification types than the standard Email notification activity. You could create a flow that sends a notification to a MS Teams integration, and create an activity that calls that flow.

Steps to create a custom activity in Creator Studio

There are multiple steps to add custom activities for use in Creator Studio automations.

Adding a completely new activity

The steps to create a custom activity for a playbook are as follows:

1. **Create the flow, subflow, or action**: An admin or someone with the correct role must first create the flow, subflow, or action in Workflow Studio. For more information, see General guidelines for Workflow Studio flows, subflows, and actions.
2. **Define the activity**: An admin then defines the activity by adding it to the Activity Definition table and assigning the flow/subflow/action to a playbook activity. This step is also when the admin specifies which inputs can be made available as fields in the activity settings in Creator Studio playbooks. For more information, see Activity definitions and Create an activity definition.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: If an activity definition already exists for the activity you want, you can skip this step.

3. **Make the activity available in Creator Studio**: An App Engine admin adds the activity to the Creator Studio Activities table. For more information, see Make a custom activity available for playbooks in Creator Studio.

Making existing activities and playbooks available in Creator Studio

Admins can make existing flows, subflows, and actions available in Creator Studio by defining the activity and then adding it to the Creator Studio Activities table (the preceding steps 2 and 3).

Admins can also make existing playbook activities available in Creator Studio by adding them to the Creator Studio Activities table (the preceding step 3).

Note: Creator Studio admins can see the Activity Definitions [sn_pd_activity_definition] table by default, which also enables them to administer processes in Workflow Studio.

Make a custom activity available for playbooks in Creator Studio

You must add custom activities to the Creator Studio Activities table [sn_creatorstudio_activity] before users can add them to playbooks when building apps.

About this task

You may want to create custom activities for complex or company-specific workflows. For example:

- The standard Request approval activity has a **Requester's manager** option, but you may need a **Department manager** instead.
- You may need an approval or flow with more complex decision logic than is easily built in the Creator Studio automation. You could use Workflow Studio to build that flow, and then create a custom action that calls it.
- You may want more notification types than the standard Email notification activity. You could create a flow that sends a notification to a MS Teams integration, and create an activity that calls that flow.

Before you begin

The following steps must be completed before you can make a custom activity available in Creator Studio:

1. **Create the flow, subflow, or action**: An admin or someone with the correct role must first create the flow, subflow, or action in Workflow Studio. For more information, see General guidelines for Workflow Studio flows, subflows, and actions.
2. **Define the activity**: An admin then defines the activity by adding it to the Activity Definition table and assigning the flow/subflow/action to a playbook activity. This step is also when the admin specifies which inputs can be made available as fields in the activity settings in Creator Studio playbooks. For more information, see Activity definitions and Create an activity definition.

Note: If an activity definition already exists for the activity you want, you can skip this step.

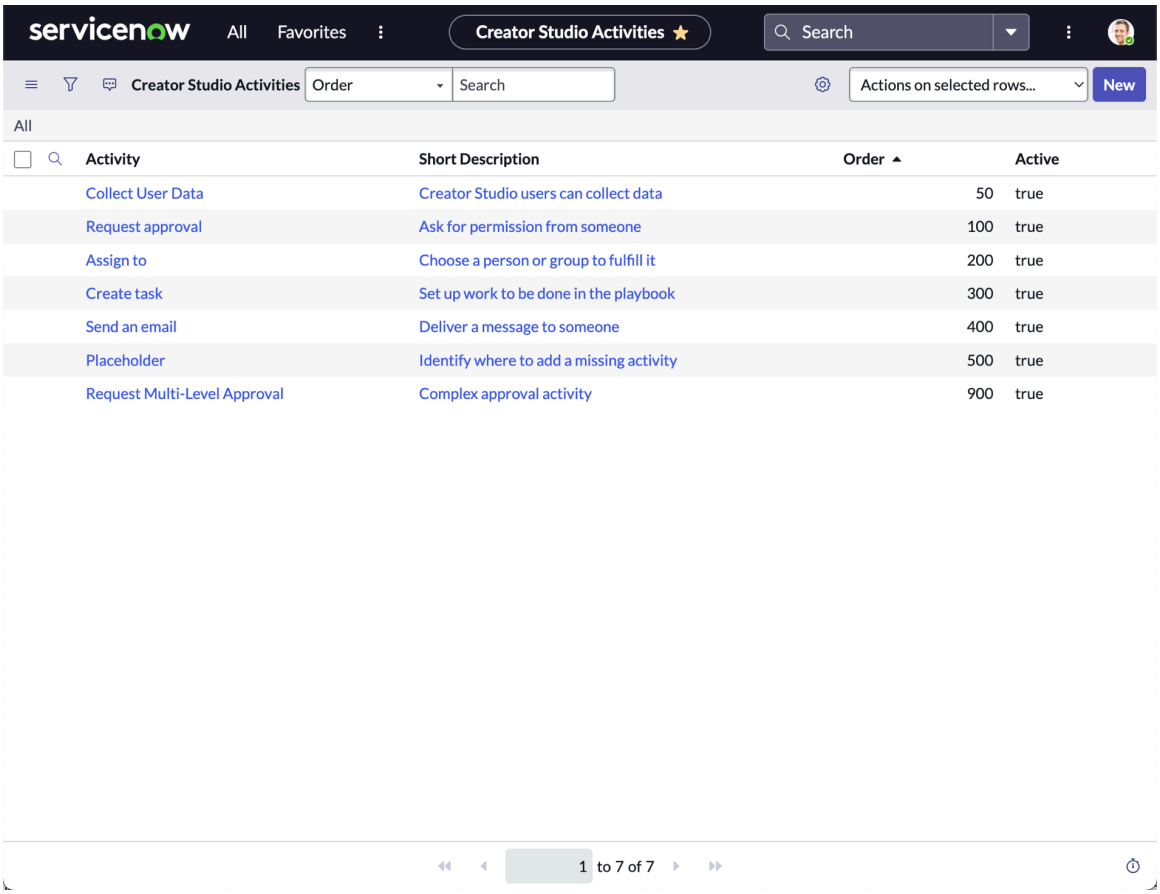
Role required: admin or app_engine_admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > *App Engine* > **Request App Administration** > *Creator Studio* **Activities**.
   All of the standard Creator Studio activities appear in the list, not just custom activities.

Activities table



   The Creator Studio Activities table appears.

2. Create the basis of the new activity.

   a. Select the **New** button.

   b. On the record, fill in the fields.

Field	Description
Activity	Search for and select the type of activity that will be the basis of your custom activity. For example, create a complex approval activity by selecting **Request Multi-Level Approval**. **Note:** Once you choose an activity, you can select the information icon to preview its Activity Definition record.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Order	Order that the activity appears in the Creator Studio activity picker list. For example, an activity with order number 400 appears before an activity with order number 500.
Short description	Brief description of the custom activity, which appears in the activity picker.

To see a list of all the available activities you can add:

- See Workflow Studio actions for all standard actions.
- See Workflow Studio flow logic for a list of all flows.
- See your list of all available subflows in Workflow Studio.

c. Confirm that the **Active** option is selected to make the activity appear in Creator Studio.

d. Select **Submit**.

3. Specify which fields appear in the playbook activity's properties panel.

Note: The fields that appear may have already been configured in the process of defining the activity. If so, follow these steps to verify that the available fields are correct.

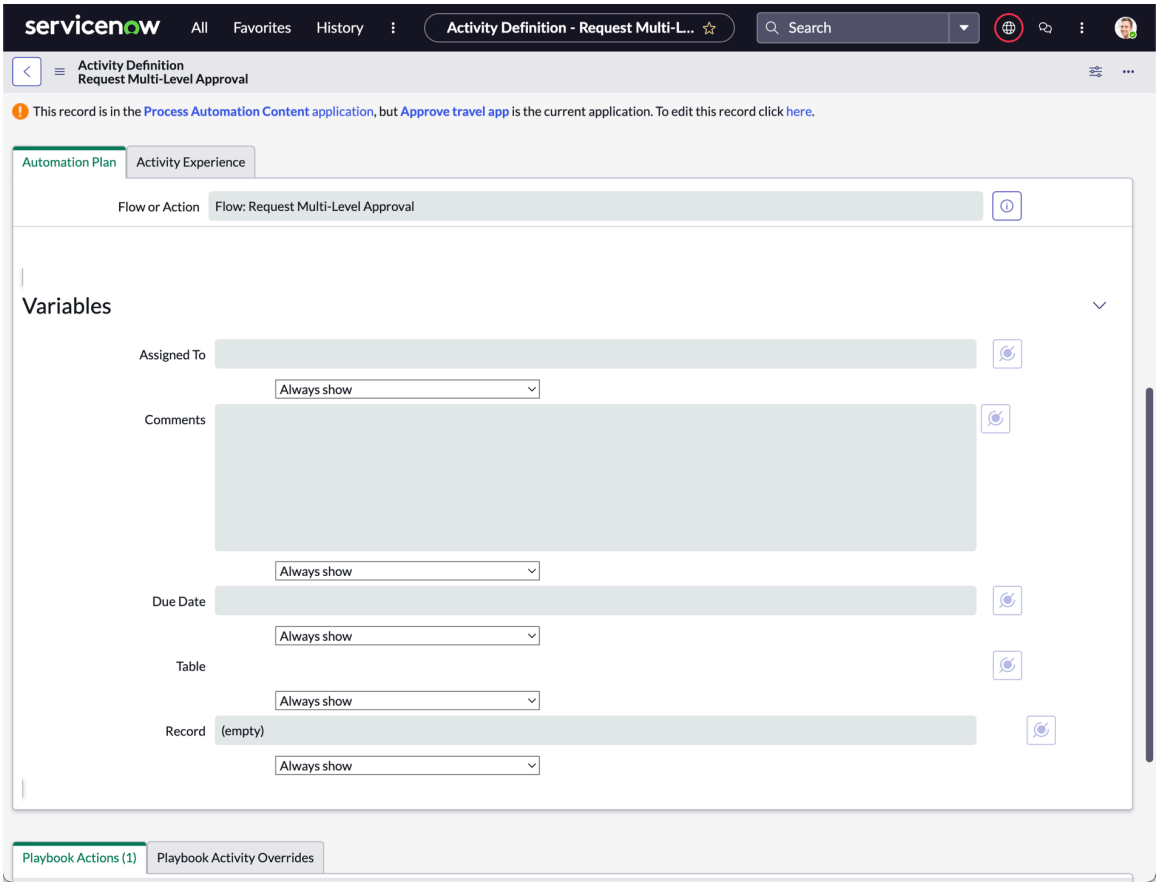
a. Open the definition for the newly created activity by selecting it from the **Activity** column of the Creator Studio Activities table.

b. Confirm that the **Automation Plan** tab is selected.

c. Make the field appear for the activity's properties in a Creator Studio by selecting **Always Show**.
For example, set the **Assigned To** field to **Always Show** to users building a playbook in Creator Studio.

For more information, see Add dynamic inputs to an activity.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



d. **Optional:** Hide a field from the properties panel by setting it to **Show as additional property for admins only.**

e. Repeat these steps for every field that should be visible on the properties panel.

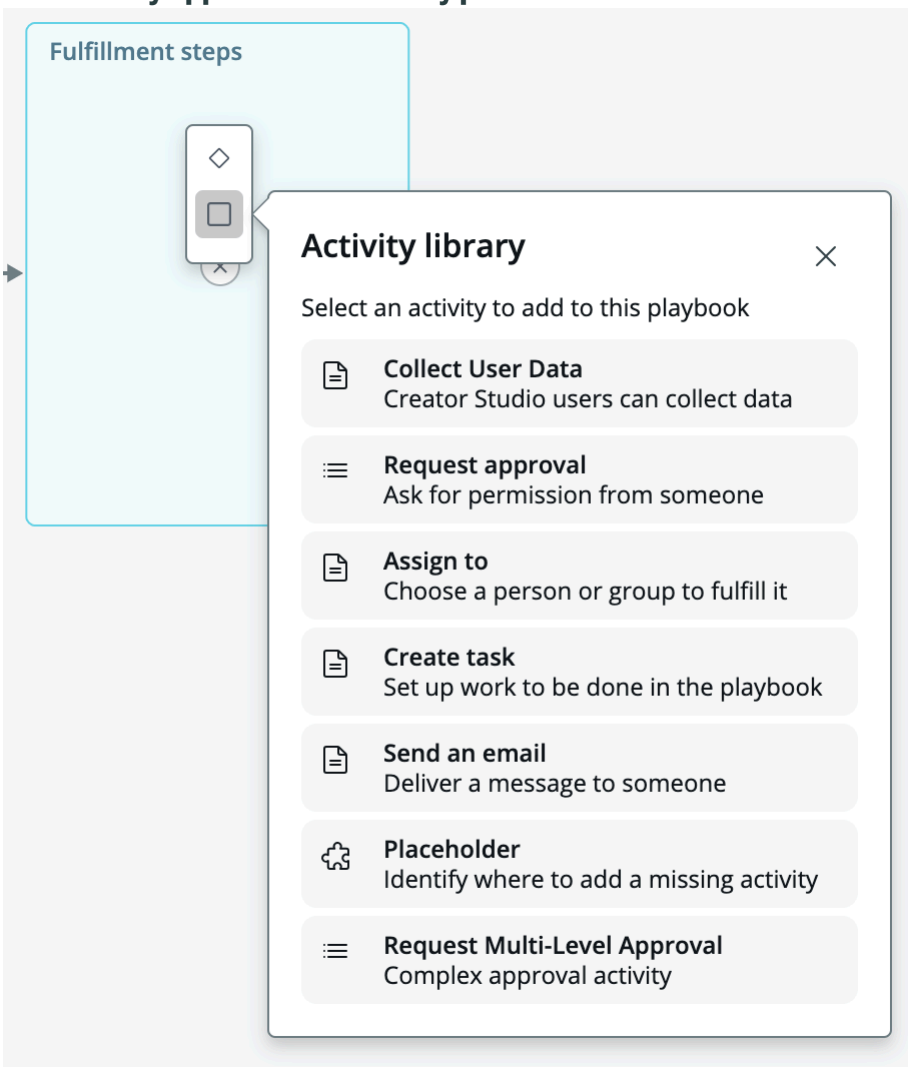
f. Save the record.

4. Return to the Creator Studio Activities table.
5. Review the Creator Studio Activity record by selecting is **Short Description** field.
6. Confirm that the **Active** option is selected, and that everything else is correct.
7. **Optional:** Save the record if you made any additional changes by selecting the **Update** button.

Result
The new activity appears in the activity picker in Creator Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

New activity appears in the activity picker



Reorder how activities appear in a playbook's activity picker

Change the order that activities appear in the activity picker in Creator Studio to present them in the order you want.

Before you begin

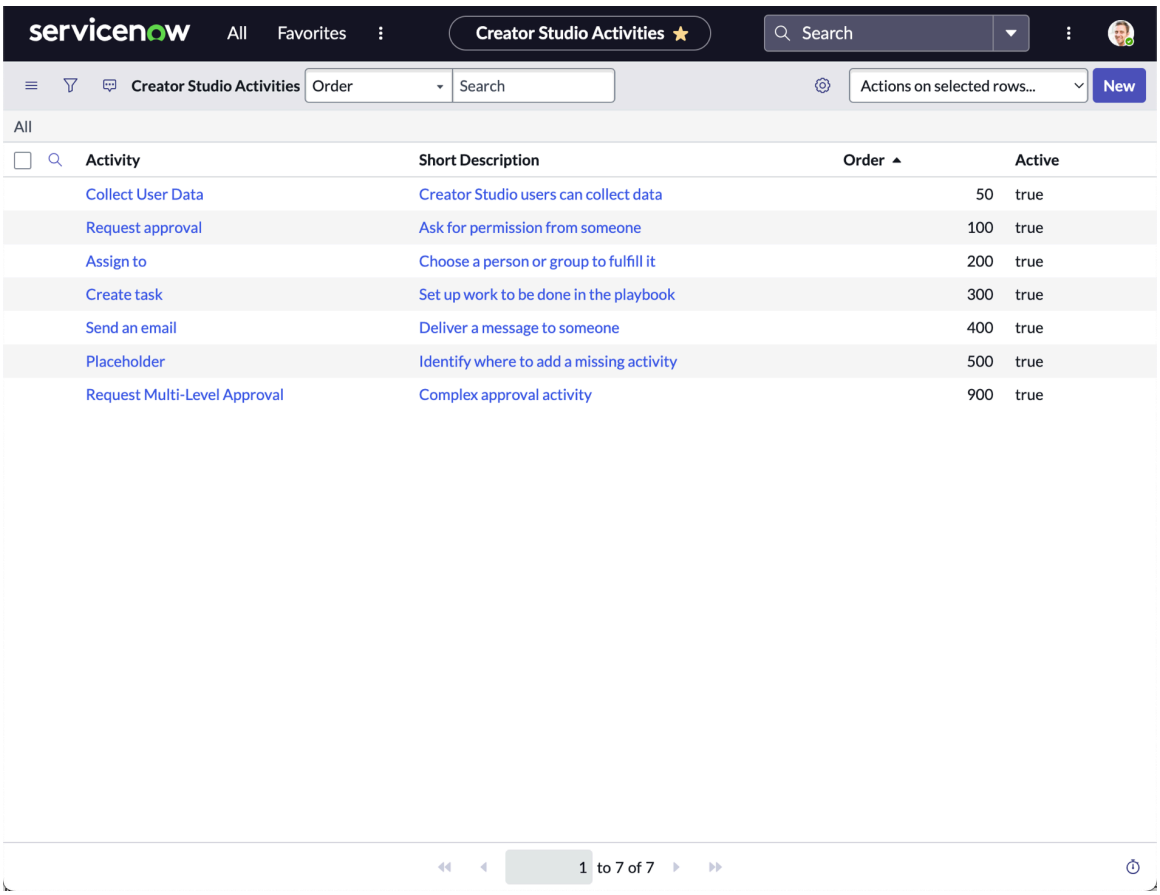
Role required: admin or app_engine_admin

Procedure

1. Navigate to **All** > *App Engine* > **Request App Administration** > *Creator Studio* **Activities**.
   All of the standard Creator Studio activities appear in the list, not just custom activities.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Activities table



The Creator Studio Activities table appears.

2. Select into the **Order** column for the activity that you want to change the order of.
3. Enter a number to represent where in the activity picker list the activity should appear.

   For example, an activity with order number 400 appears before an activity with order number 500.

4. Select the Save icon to complete the update.
5. Change the **Order** for every activity that you want.

What to do next

Alternatively, you can open each Creator Studio Activity record and update the **Order** one record at a time.

Make an activity unavailable in playbooks

Deactivate an activity to stop users from adding it to playbooks in Creator Studio.

Before you begin

For example, you could hide an base system activity, like sending an email, if it's not relevant to your workflow.

Note: Hiding an activity doesn't change anything about the processes using that activity. Hiding an activity only makes it unavailable in Creator Studio.

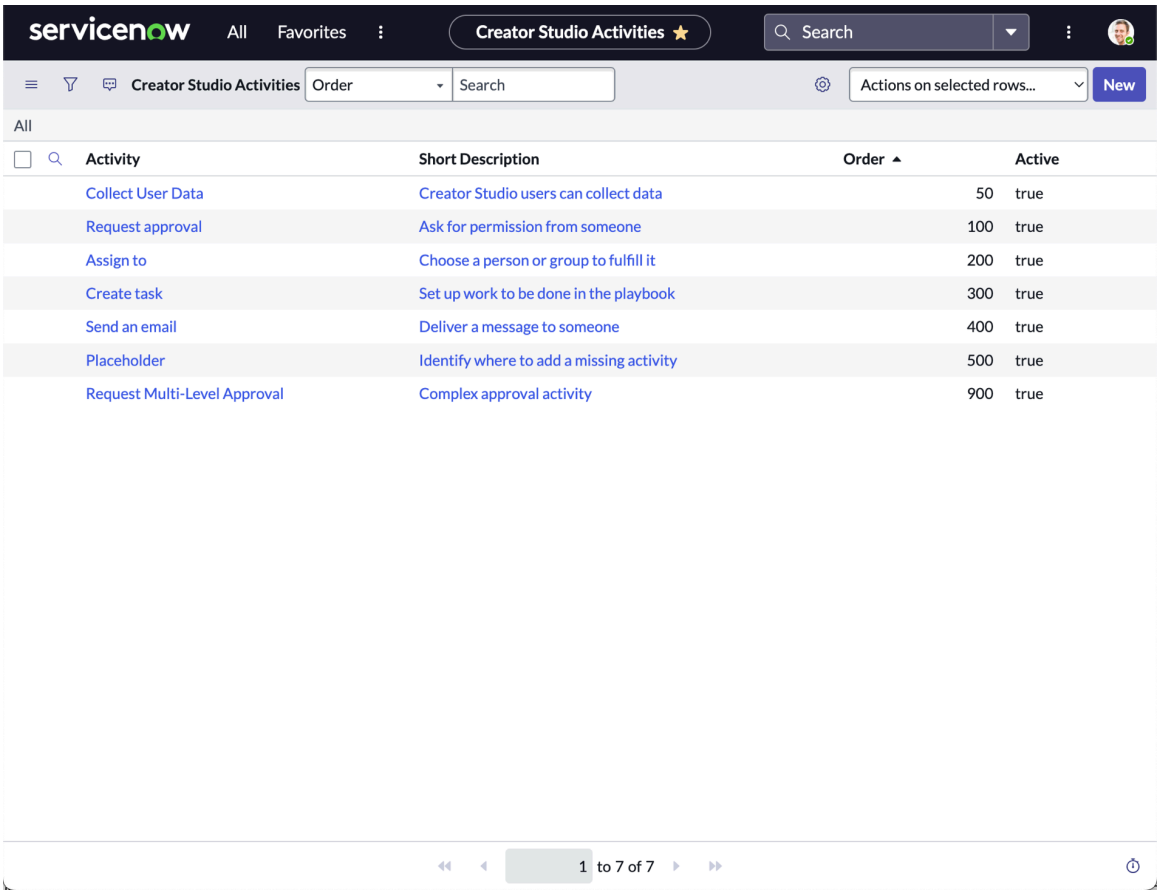
Role required: admin or app_engine_admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > *App Engine* > **Request App Administration** > *Creator Studio* **Activities**.
   All of the standard Creator Studio activities appear in the list, not just custom activities.

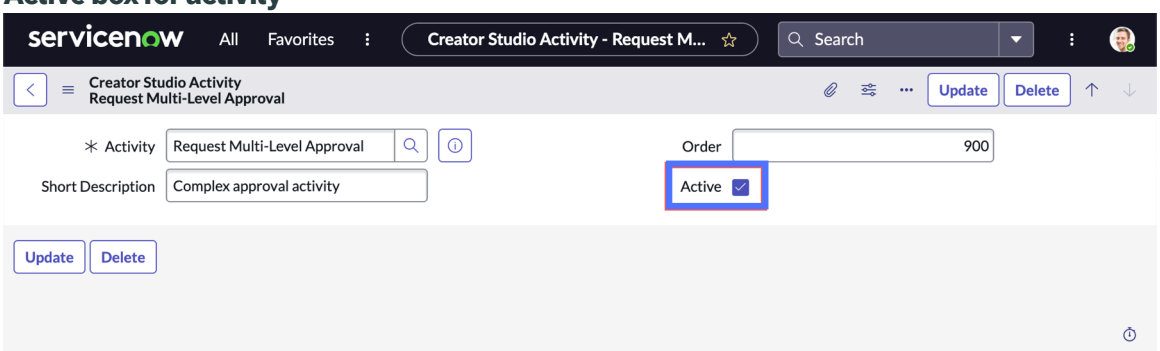
Activities table



   The Creator Studio Activities table appears.

2. Open the Creator Studio Activity record for the activity that you want to hide by selecting is **Short Description** field.
3. Hide the activity by clearing the **Active** check box.

Active box for activity



4. Select **Update** to save the record.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Disable the App type page in Creator Studio

Hide the App type page that appears when users select to create an app by adding a system property to streamline the app creation process.

Before you begin

You must change your scope to global. For more information, see Global scope.

Role required: admin

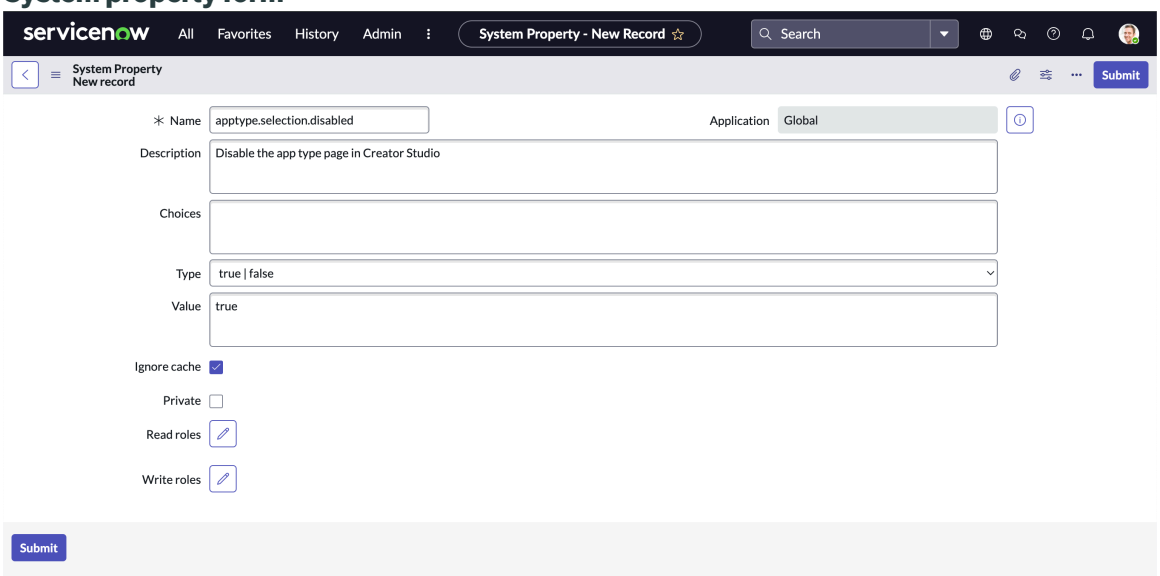
Procedure

1. Open the **All** menu.
2. Type `sys_properties.list` and press Enter.
3. Select the **New** button.
4. On the form, fill in the fields.

Values for the property

Field	Description
Name	Enter `apptype.selection.disabled`.
Description	Disable the app type page in Creator Studio.
Choices	Leave blank.
Type	Select **true | false**.
Value	Enter `true`.
Ignore cache	Leave as-is.
Private	Leave as-is.
Read roles	Leave as-is.
Write roles	Leave as-is.

System property form



5. Select **Submit**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Result
Users on the instance will no longer see that App type selection page after they initiate creating an app.

To reinstate the App type page, set the *apptype.selection.disabled* property to `false`, or just delete it.

Managing Creator Studio deployments and monitoring app usage

After users create apps, they submit them for you, the admin, to review and deploy. How you deploy them depends on whether you have App Engine Management Center (AEMC) installed on the instance.

Managing app deployment with AEMC

Apps aren't deployed directly from Creator Studio. Instead, use App Engine Management Center (AEMC).

Find details on deploying apps in App Engine Management Center.

Managing app deployment without AEMC

If you don't have AEMC installed, you can use Pipelines and Deployments and the Deployment Tasks table to publish to the Application Repository. For more information, see Pipelines and Deployments workflow version 24.1.2.

You can view and edit details of the deployment on the **Scheduled Deployments** tab of Pipelines and Deployments For more information, see Managing deployments using Pipelines and Deployments.

Monitoring deployed app usage

After you deploy an app to production, you can monitor its usage, adoption, and response rate, for each request form in the same way that you do for catalog items/record producers or scoped apps.

For example, you can use AEMC to monitor apps. For more information, see Manage custom apps using AEMC.

Administering user access for deployed Creator Studio apps

After users build apps in Creator Studio and you deploy them to production, as an admin, you are responsible for granting access to those apps to users.

Assign user access in production

Applications must be deployed to the production instance, which is where you assign user access.

Users with the agent role are fulfillers, and should automatically get access to apps built in Creator Studio.

Assign users the agent role for the app's scope, for example, `x_snc_jd_it_fulf_0.agent` for an app with the scope `x_snc_jd_it_fulf_0`. For details on assigning roles to users, see Managing roles.

Restricting user access to forms

By default, all users on the production instance get access to forms built in Creator Studio, but Creator Studio developers can choose from existing access criteria if they want to restrict

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

access. The **Available For** and **Not Available For** form settings are lists of User Criteria Records. For more information, see Set security for items and categories.

Developers can work with admins to define new access criteria for their app's forms to define the proper access from within Creator Studio. For more information, see Edit the settings for a form in Creator Studio.

Personas for apps built in Creator Studio

Requester

Someone requesting something, like a piece of equipment or permission to do something.

Fulfiller

Someone who works on requests. Fulfillers may also approve or deny requests, depending on any approval automation for the app.

Artifacts that Creator Studio generates when users create an app

When a user creates an app in Creator Studio, the ServiceNow AI Platform creates several metadata artifacts, such as tables.

For each app users add in Creator Studio, the following artifacts are added to the ServiceNow AI Platform.

Installed with each created app

Artifact	Details
New table that extends from the base Request Task table	• Name: [scope]_request (for example, x_snc_marketing_request) • Label: [App name] + Requests (for example, Marketing Request) **Note:** All of the form questions are stored as catalog variables in the app's Request Task table. However, admins can change the associated table for an app, see Administering an app's associated table for more information.
Fulfiller role	• Name: [scope].agent • Provides access to the table created by the app
Access Control Lists (ACLs)	Provides CRUD (create, read, update, and delete) access to the new table for the fulfiller and requester roles.
Category for the new app's workspace	Name: [App Name] + Requests
Default lists for the new workspace category	• Open (active=true) • Open – Unassigned (active=true^assigned_toISEMPTY)

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Installed with each created app (continued)

Artifact	Details
	• Closed (active=false) • All (no conditions)
Dashboard in the new workspace category	Dashboard name: [App Name] + Requests Contents: • A row of single scores: ○ Open requests (active=true) ○ Unassigned requests (active=true and assigned to is empty) ○ Closed last 30 days (closed >= 30 days ago) • A simple list showing all open requests

Creator Studio tutorial

Use this tutorial to create a Gift Card Request app to streamline the processes of providing recognition to a team.

Tutorial objectives

In this tutorial, you'll assume the following roles to create and deploy the app:

- As a marketing manager who is looking to streamline the processes of providing recognition to their team, you'll create the Gift Card Request app.
- As an application development administrator, approve and deploy the new app to a production environment.
- As a marketing manager, use your live app to submit a recognition request.

Tutorial scenario

You're a manager with a rapidly expanding team. To date, the team has exceeded its goals every quarter. You want to reward them and provide gift cards as part of that recognition. However, securing approval for monetary awards is an intricate process that can span hours and involve repetitive interaction with many systems and upper management.

To tackle this issue, you check in with the IT admin and confirm that you can create an app to request gift cards. The admin is happy to help you make a new end-to-end app that draws on existing business process knowledge. To enable this, the admin adds you to the Creator Studio Users group in their non-production instance.

Note: To complete this tutorial, you must log in to Creator Studio as two different users or impersonate two users: one in the Creator Studio Users group and one admin.

It's time for you to get started on creating a simple app.

Tutorial part 1: Create the app

Begin by creating an app from the Creator Studio home page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin
Role required: Creator Studio User

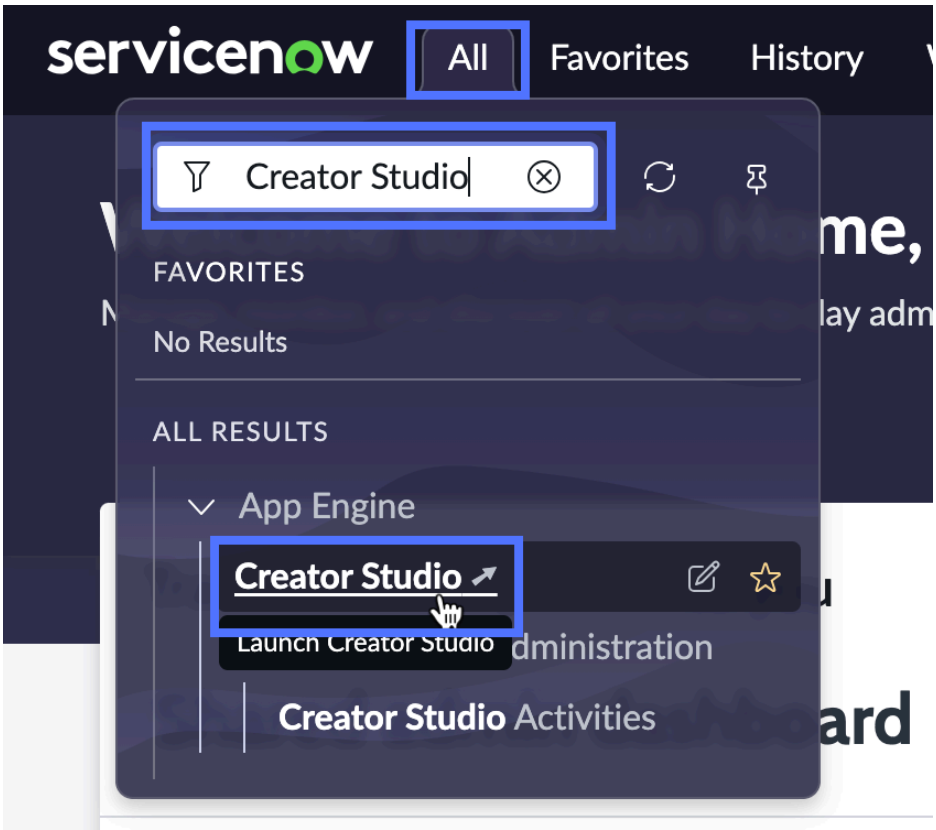
About this task
You can also watch a short video on how to create an app.

https://player.vimeo.com/video/1123376394?h=27cf6583d7&badge=0&autopause=0&player_id=0&app_id=58479

Procedure

1. Go to **All** > **App Engine** > **Creator Studio**.

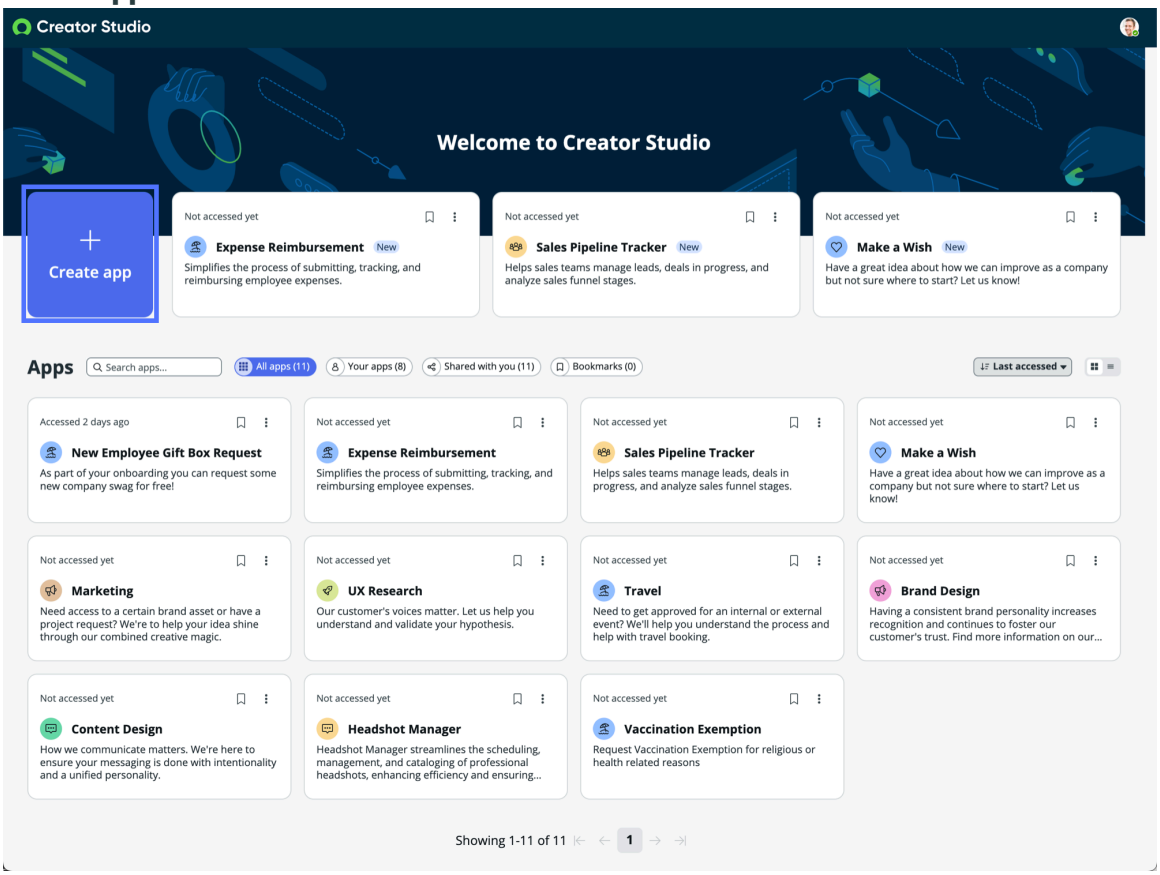
Start Creator Studio



2. Select the **Create app** button.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Create app button



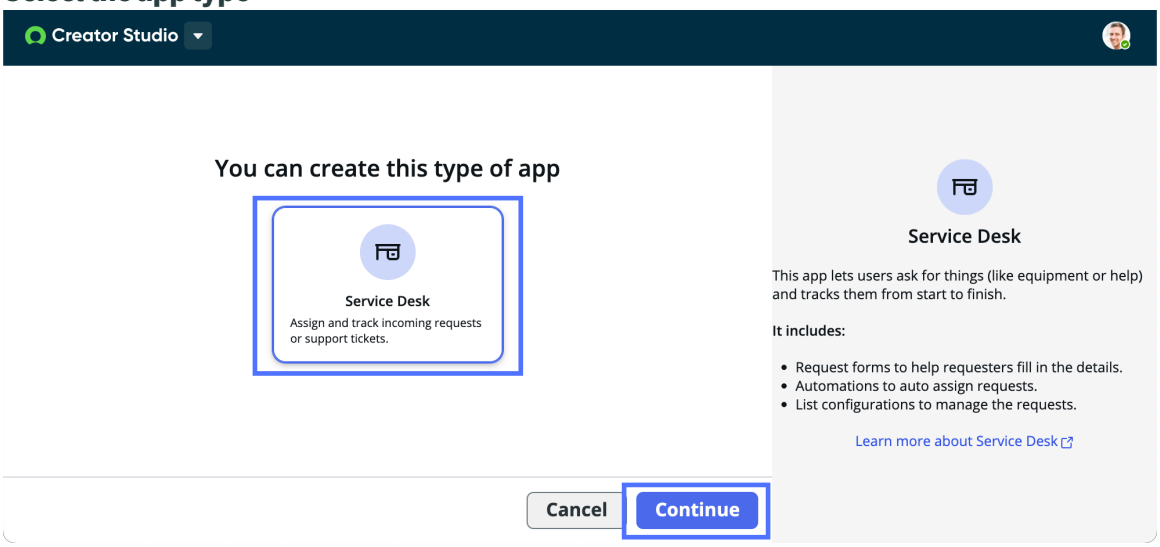
- If you're a system administrator, you can read more about this topic in Application collaboration.
- If you want to know how to request an admin to create the app for you, check out Ask an admin to create an app for you in Creator Studio.

3. Select the type of app you want to build, such as **Service Desk**.
   Service Desk apps enable users to submit requests, report issues, and access support related to services within your company.

   Note: Your admin may have hidden this page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Select the app type

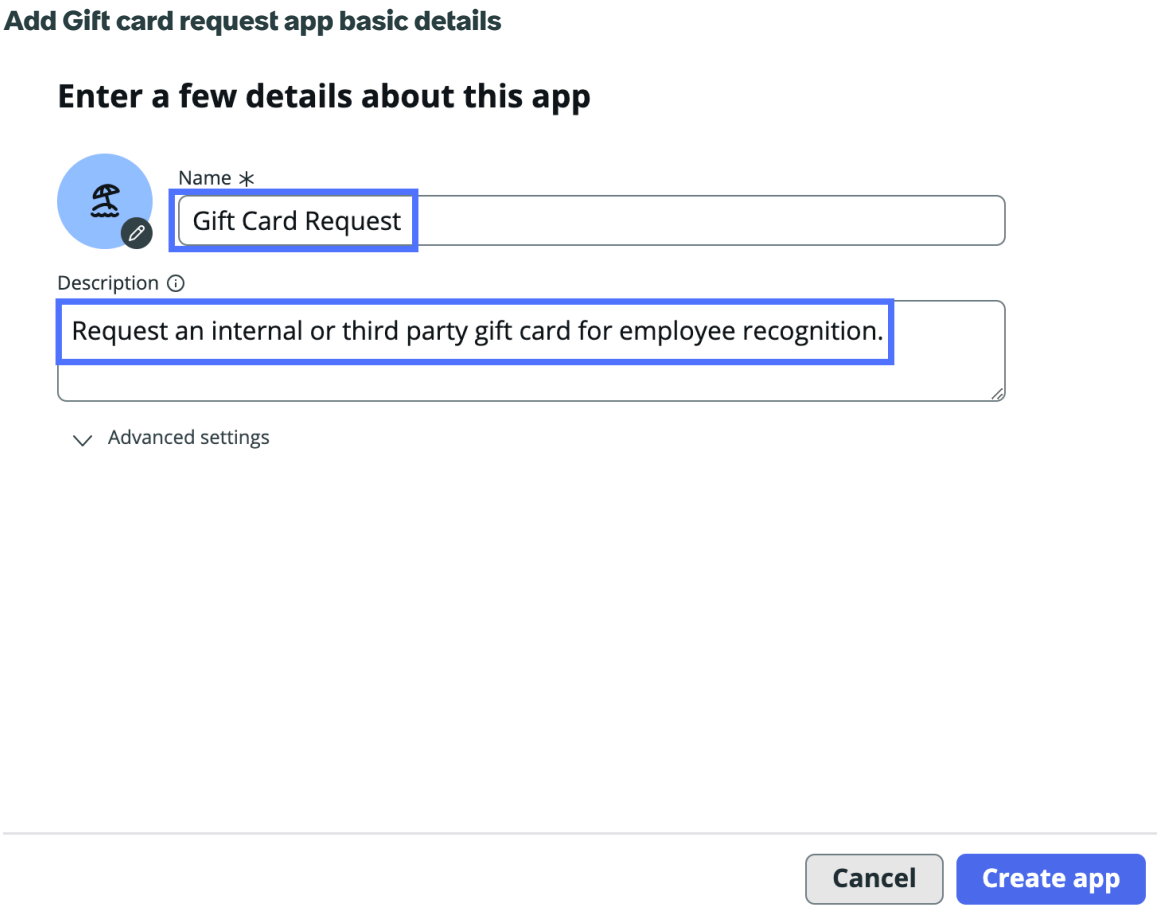


4. Select the **Continue** button.
5. Enter the following values on the Create app page.

Create app modal fields

Field	Value
Name	`Gift Card Request`
Description	`Request an internal or third party gift card for employee recognition.`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



6. Select the **Create app** button.

What to do next
Next, let's add a form to your app.

Tutorial part 2: Add a form

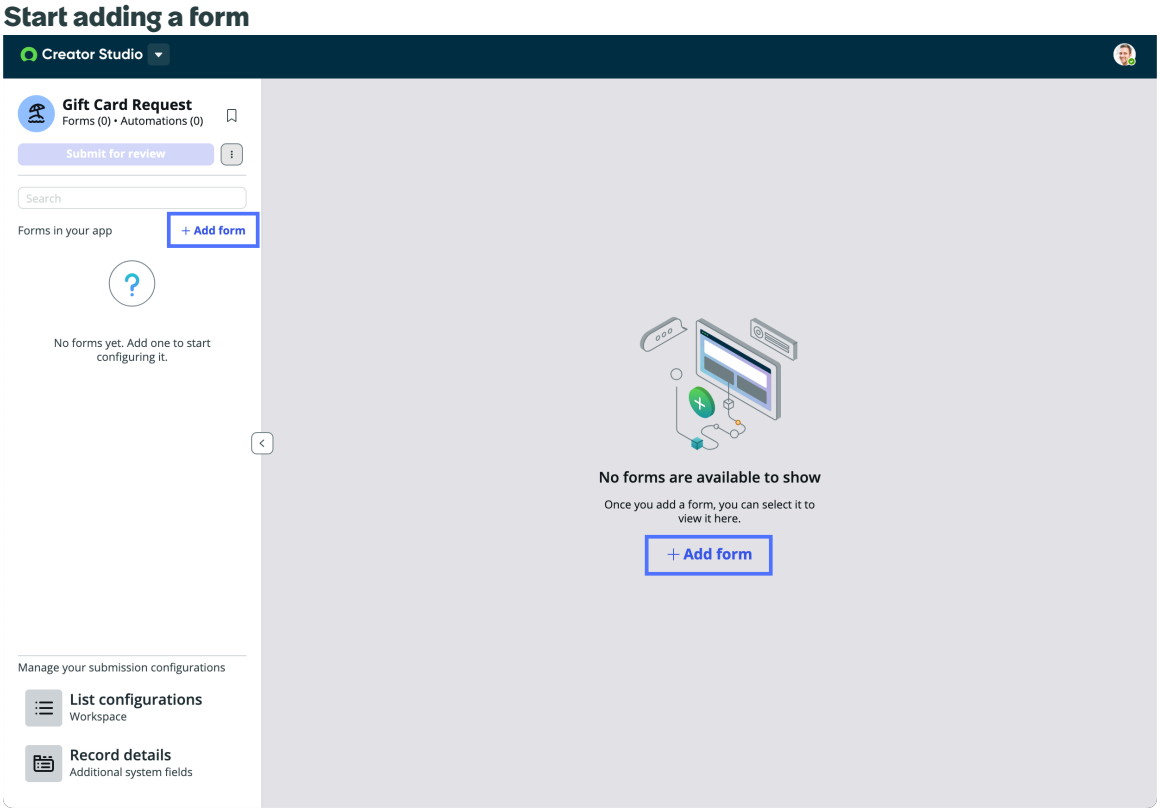
Adding a form enables people to make a request. Forms contain questions that people respond to when submitting a service desk request.

Before you begin
Role required: Creator Studio User

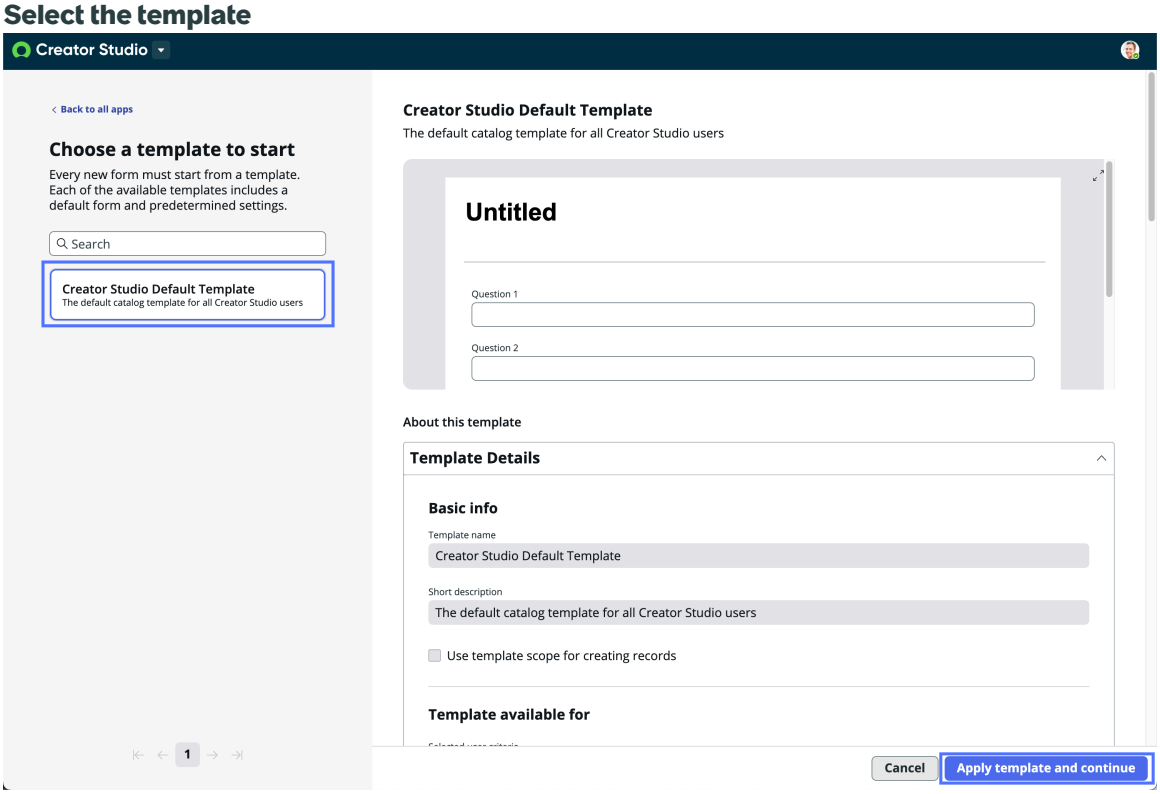
Procedure

1. Select **+Add form** within the app.
   You can select **+Add form** in either the navigation panel or the app canvas.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



2. Verify on the template page that the **Creator Studio Default Template** is selected.



3. Select the **Apply template and continue** button.
Next, let's customize the form by adding questions.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Tutorial part 3: Customize the form

Customize the form for the gift card request app by manually entering questions. When it's done, you can mark the form as ready to be published.

Before you begin

Role required: Creator Studio User

About this task

You can also use ServiceNow Otto for Creator on the **Build with AI** tab to customize a form. But for this tutorial, you will manually enter questions.

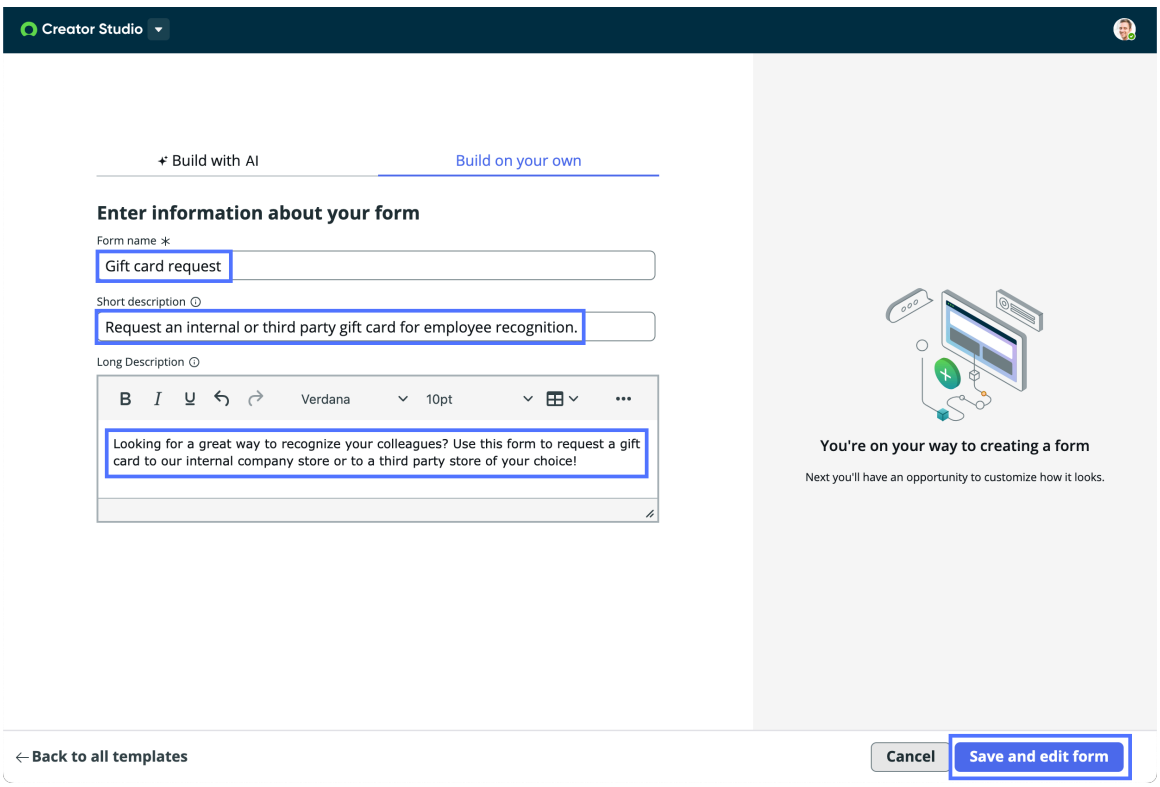
Procedure

1. Select the **Build on your own** tab.
2. Enter the following information for the form:

Form detail values

Field	Value
Form name	`Gift card request`
Short description	`Request an internal or third party gift card for employee recognition.`
Long description	`Looking for a great way to recognize your colleagues? Use this form to request a gift card to our internal company store or to a third party store of your choice!`

Enter form details

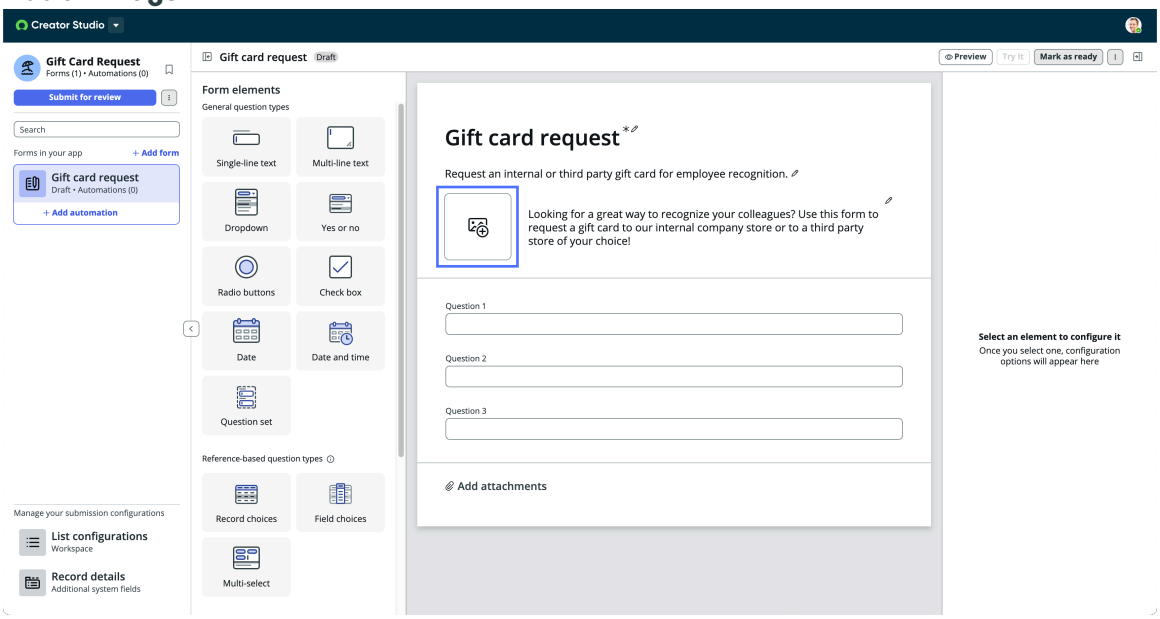


© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: This page enables you to specify details for your first form. You can also edit the form details, such as name and description, from inside the app after the form is generated.

3. Select the **Save and edit form** button.
   Next, we'll work on the contents of the form.
4. Add an icon to spiff up your form by selecting the icon and uploading an appropriate image. You can use any image that you like.

Add an image



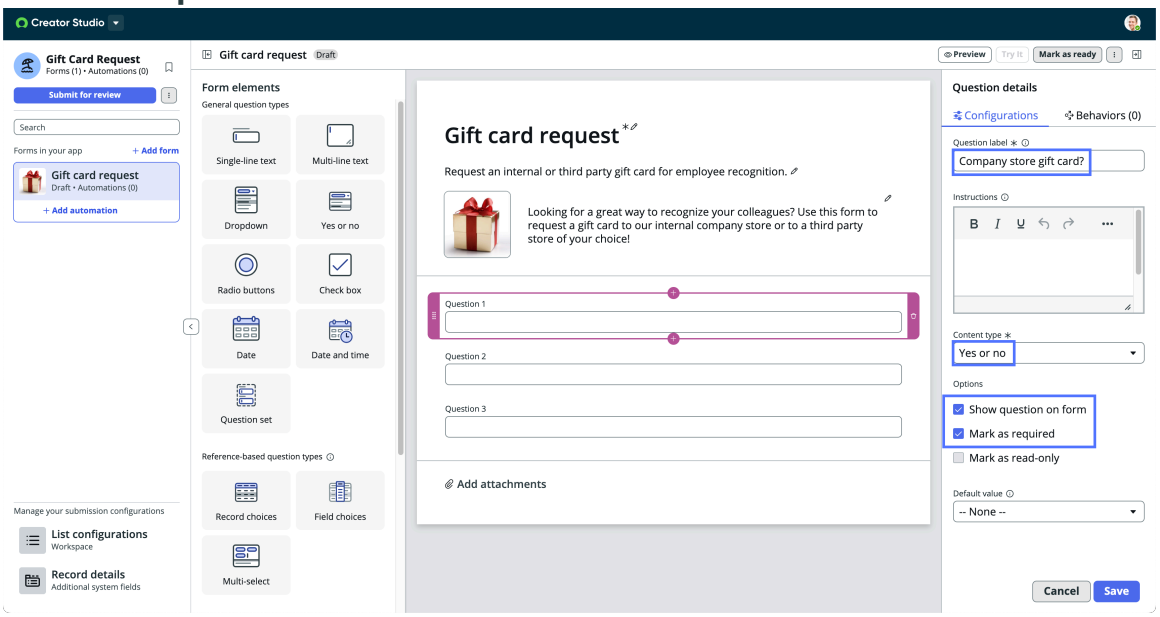
   a. Select the add image icon .
   b. Search for and choose your image to add to the form.
5. Create your first question that enables users to order the gift card.
   a. Select **Question 1**.
   b. Enter the following values in the Question details panel that appears:

Question 1 values

Field	Value
Question label	`Company store gift card?`
Content type	**Yes or no**
Show question on form	Select this option
Mark as required	Select this option

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add the first question



c. Select the **Save** button.

6. Create the second question to specify the amount of the gift card.

a. Select **Question 2**.

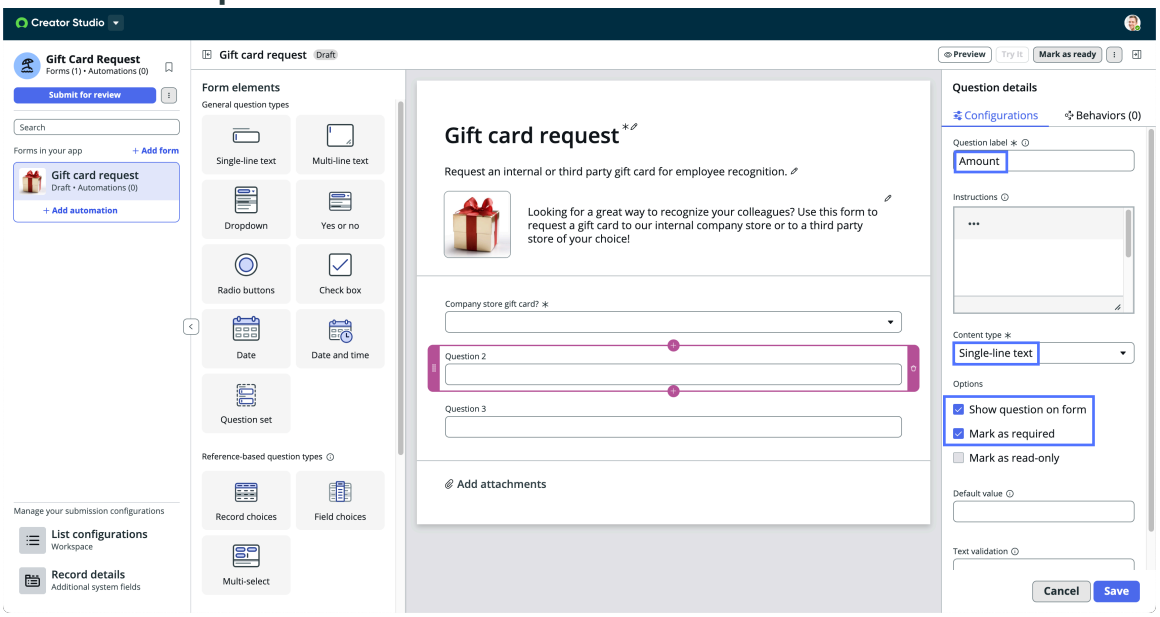
b. Enter the following values in the Question details panel that appears:

Question 2 values

Field	Value
Question label	`Amount`
Content type	Leave selected as **Single-line text**
Show question on form	Select this option
Mark as required	Select this option

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add the second question



c. Select the **Save** button.

7. Create the third question that defines who the gift card recipient will be.

a. Select **Question 3**.

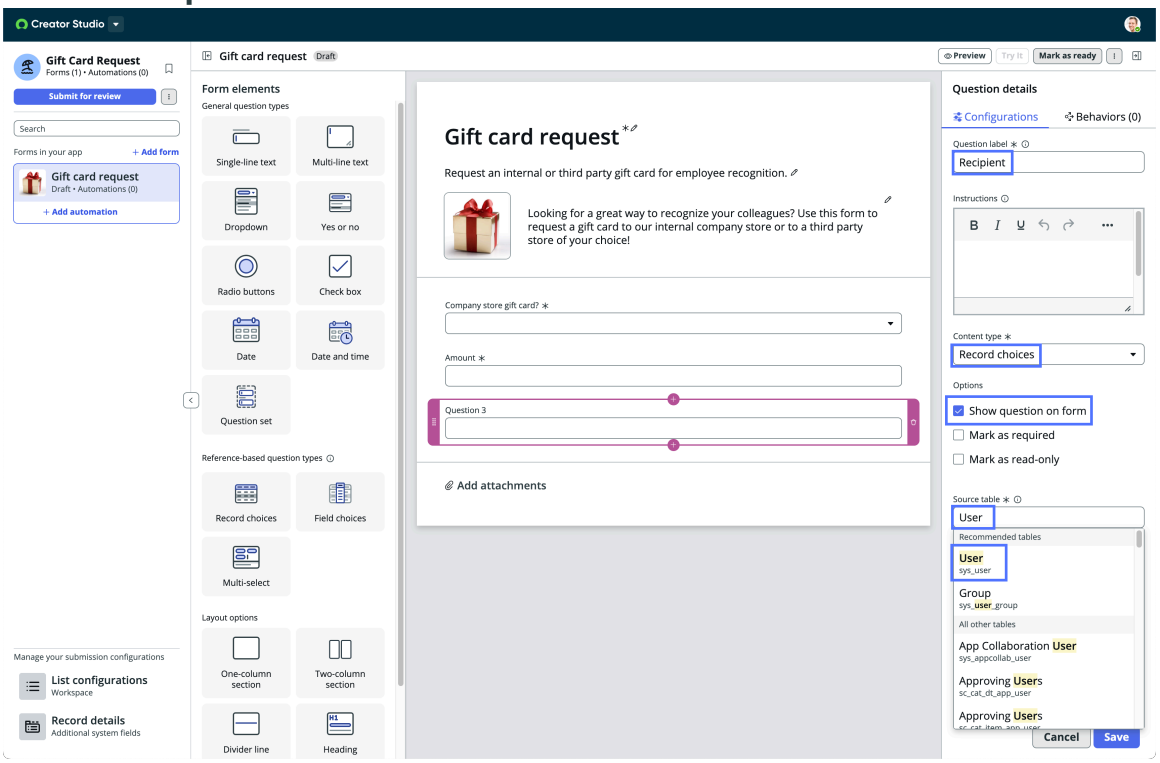
b. Enter the following values in the Question details panel that appears:

Question 3 values

Field	Value
Question label	`Recipient`
Content type	**Record choices**
Show question on form	Select this option
Source table	`User (sys_user)`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add the third question



c. Select the **Save** button.

8. Add a fourth question so that users can enter a justification for the gift card.

a. Drag the **Multi-line text** form element to add it to the form below the third question.

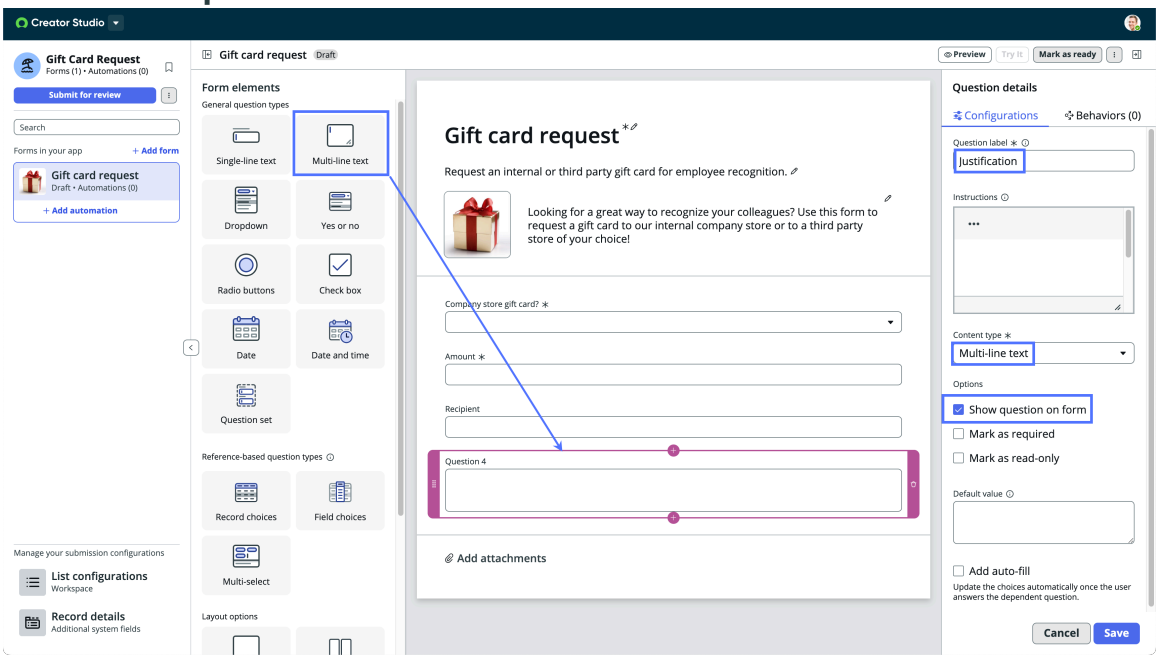
b. Enter the following values in the Question details panel that appears:

Question 4 values

Field	Value
Question label	`Justification`
Content type	Leave as **Multi-line text**
Show question on form	Select this option

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add the fourth question



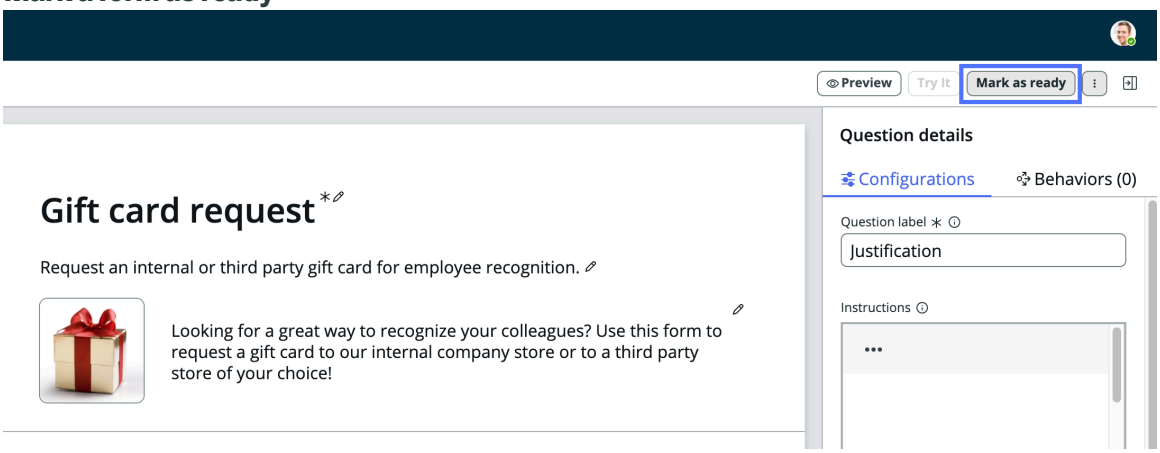
c. Select the **Save** button.

9. Select **Mark as ready** to publish the form.

When you finish editing a form, you mark it as ready, which puts it in the Published state. This version of the form then appears in its specified catalog of items.

Note: The published form appears only on the non-production instance where you're developing your app. You must request that your admin deploy the app to production for the form to appear in the catalog on the production instance.

Mark a form as ready

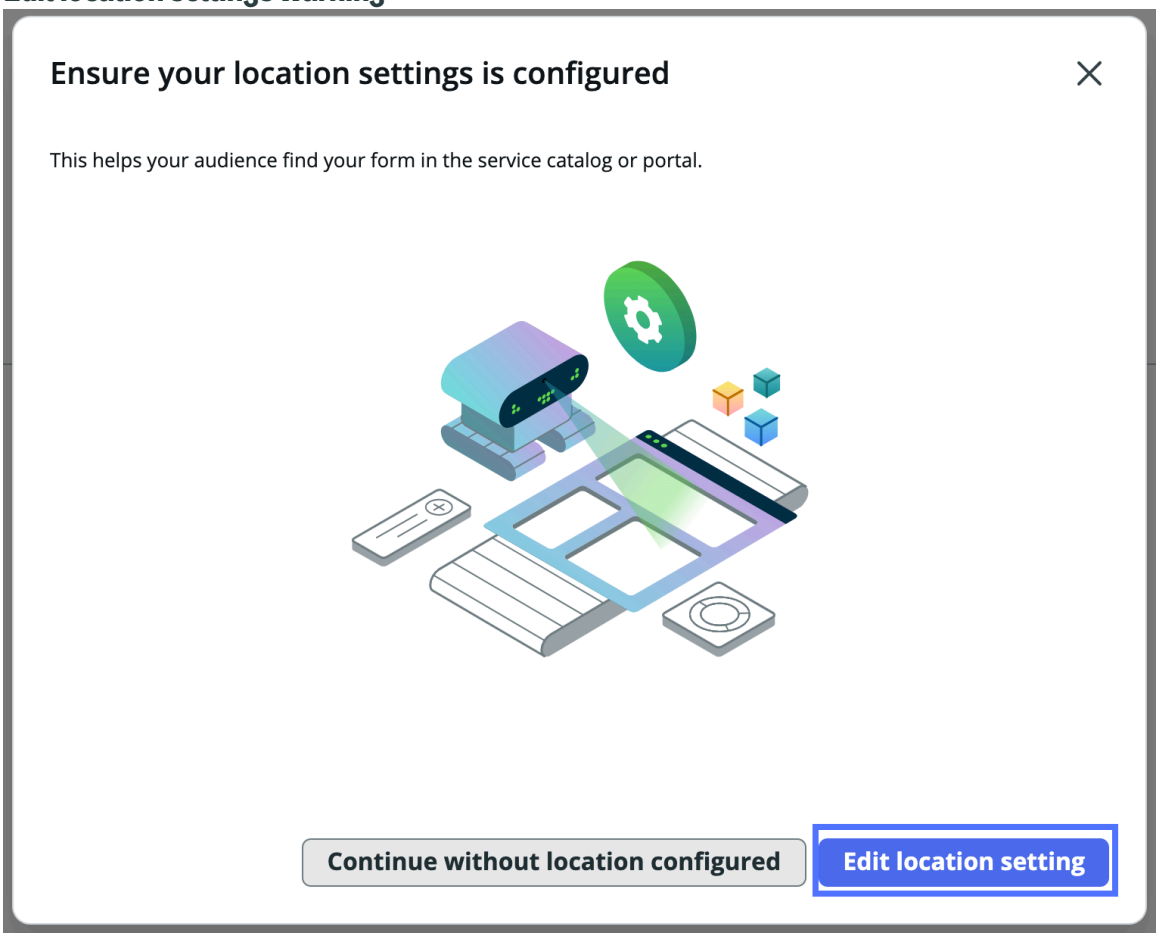


10. Specify where users can access the form in the Service Catalog and Employee Center.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. Select the **Edit location setting** button.

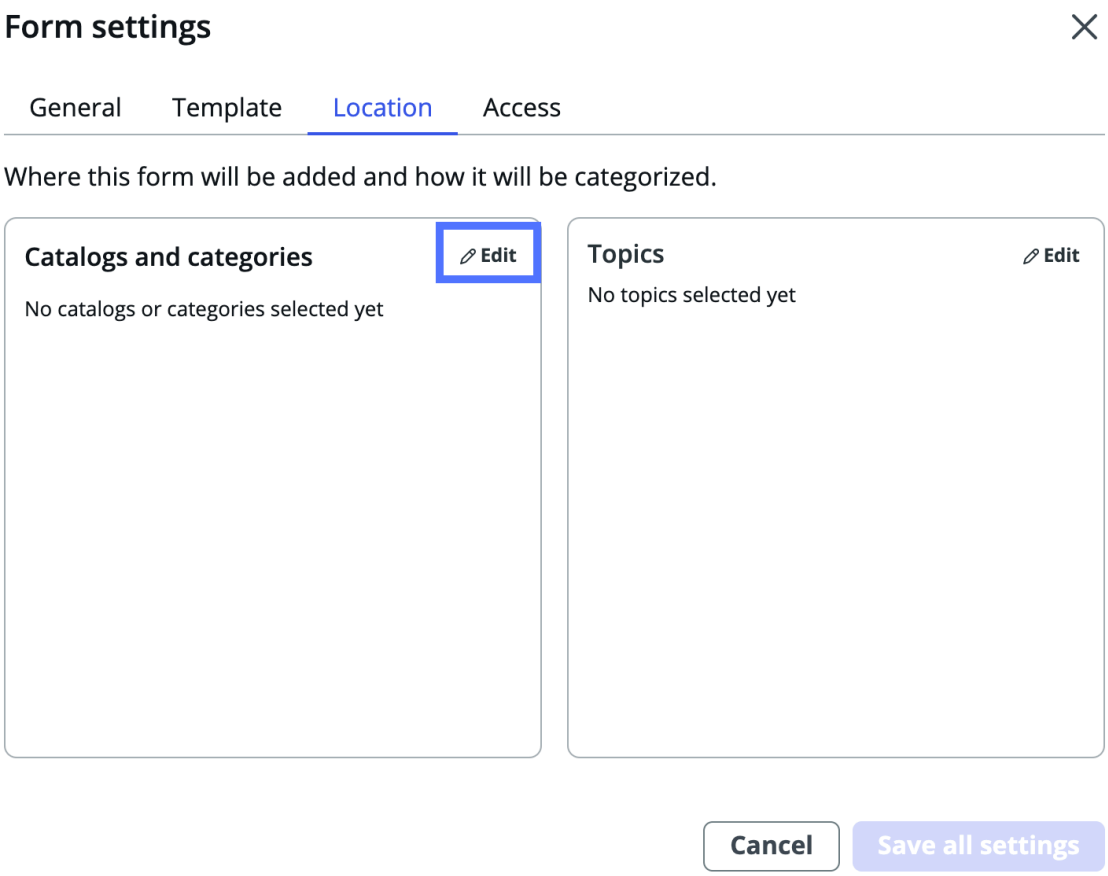
Edit location settings warning



b. Select the **Edit** button to add the form to a Service Catalog topic.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

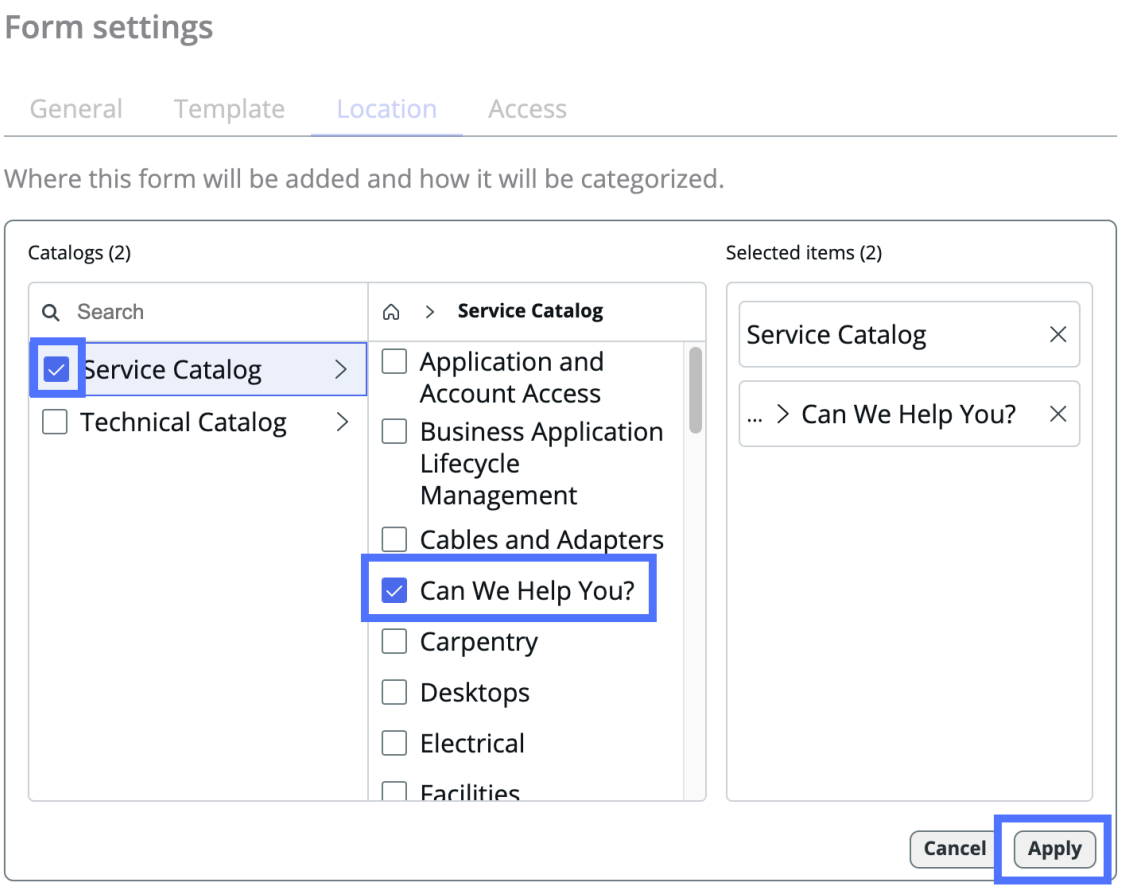
Form settings modal



c. Select the catalog that represents the business area the app will use.
For example, you can expand the *`Service Catalog`* and then select the **Can we help you?** category.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Form settings modal catalogs

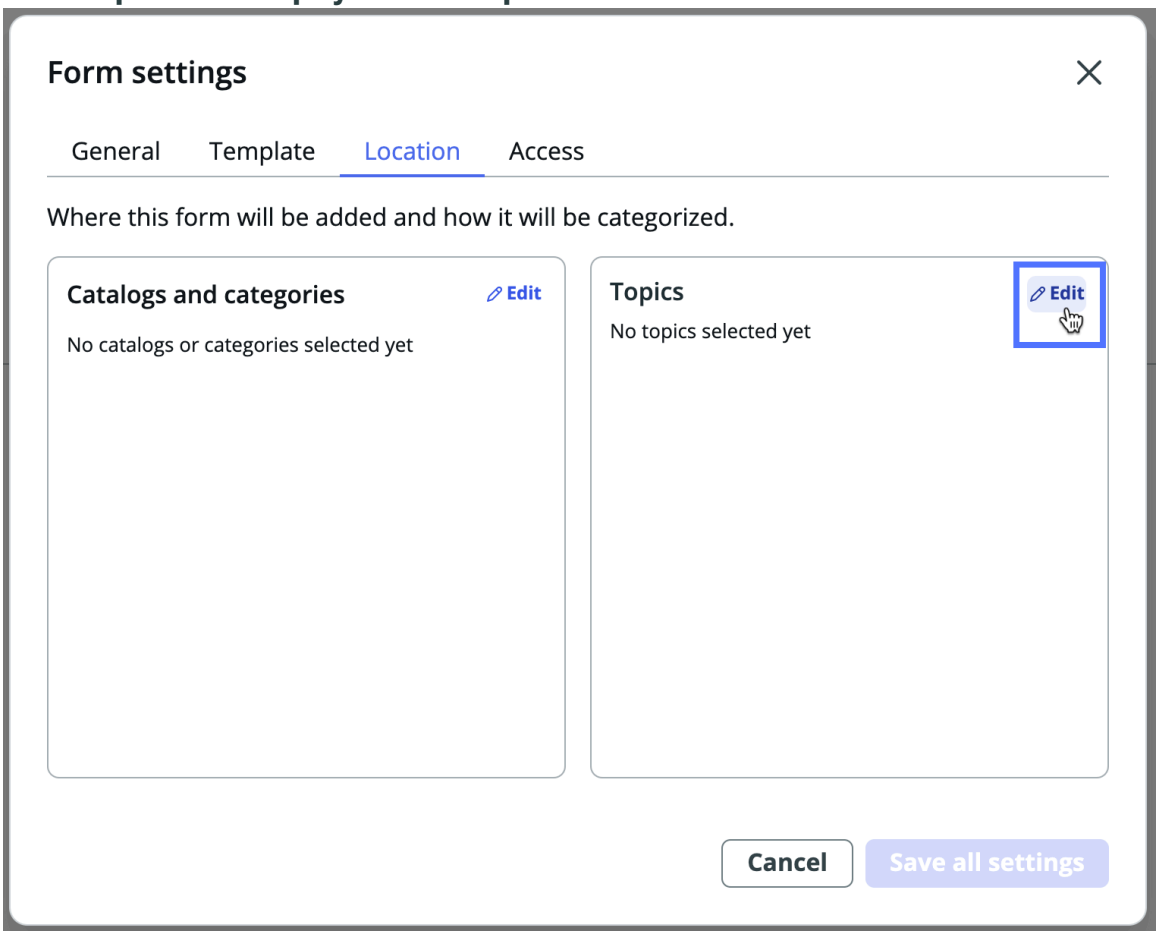


d. Select the **Apply** button to save your changes.

e. Select the **Edit** button to add the form to an Employee Center topic.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

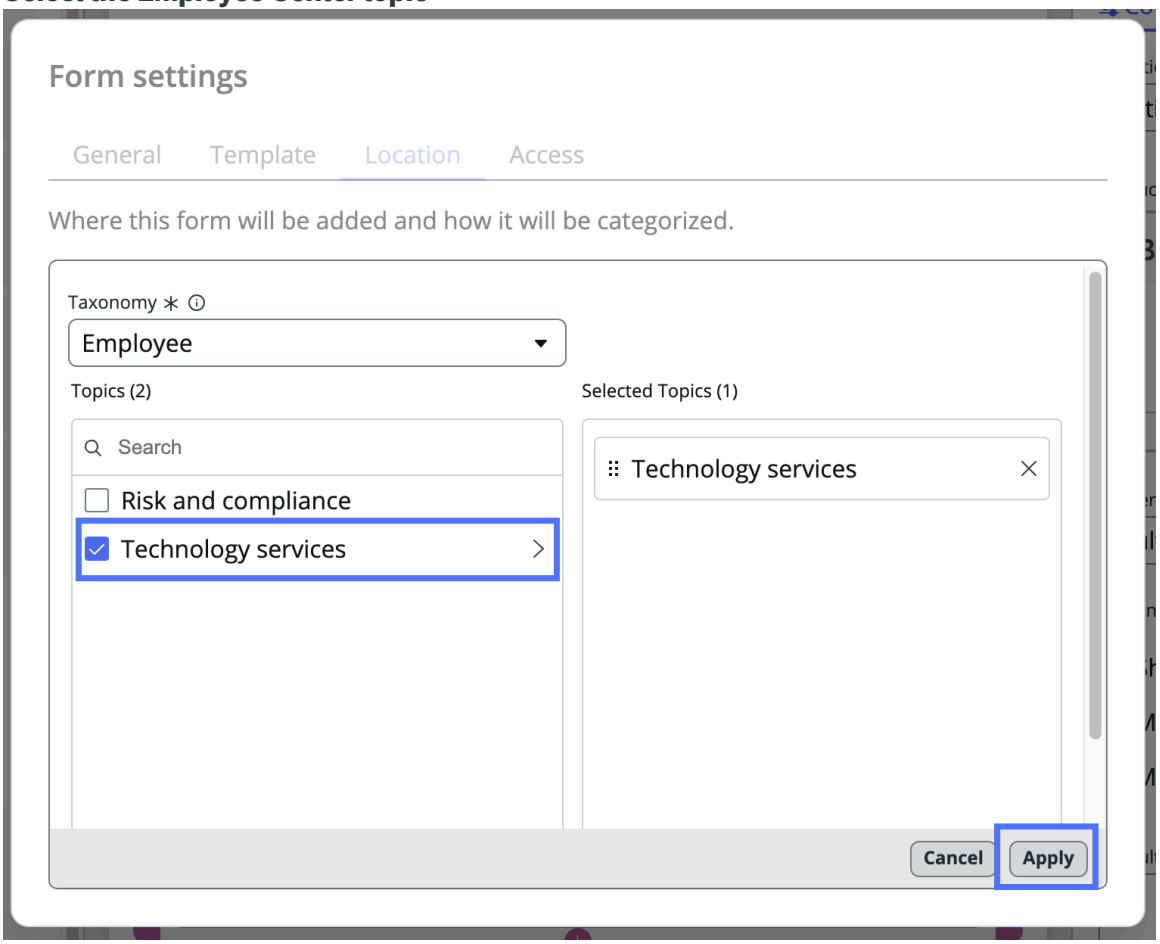
Select topics in the Employee Center topic



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

f. Choose the section of the Employee Center where the form will live.

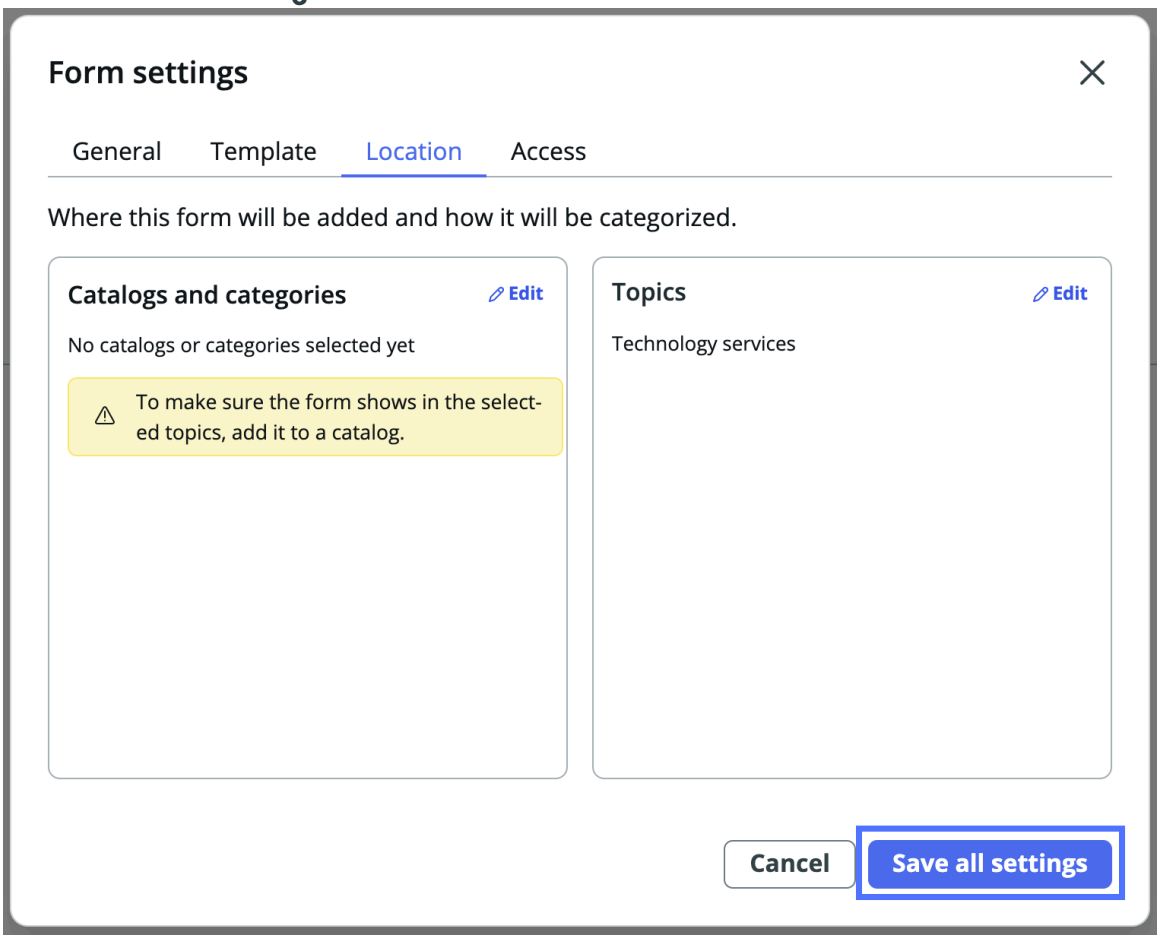
Select the Employee Center topic



11. Select the **Apply** button.
12. Select the **Save all settings** button to save your changes.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Save all location settings

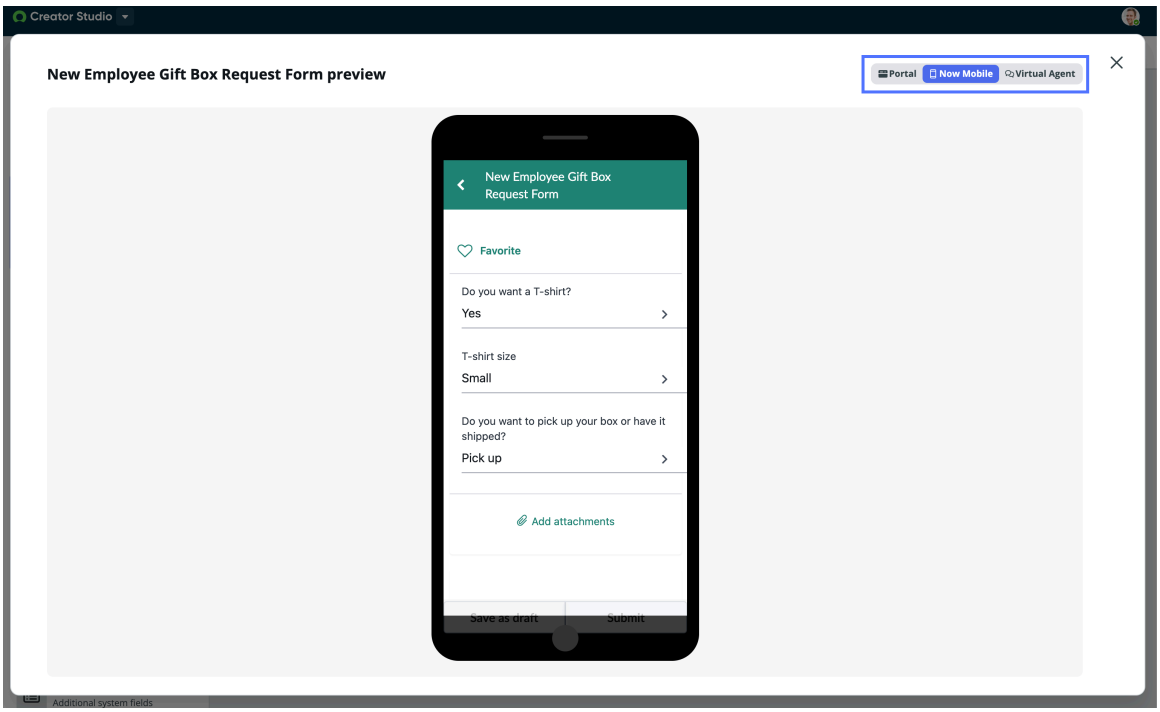


13. **Optional:** See how the form will look by selecting the **Preview** button.
    You can see a preview of the how the current version of the form will appear in various experiences by selecting **Portal** (such as Employee Portal), **Now Mobile**, or **Virtual Agent**. You

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

can fill in the form when previewing, but selecting the **Submit** button doesn't generate a task record.

Preview a form



Tutorial part 4: Add an automated playbook

Add a playbook that uses the form that you just created to automate approval for the gift card request app.

Before you begin

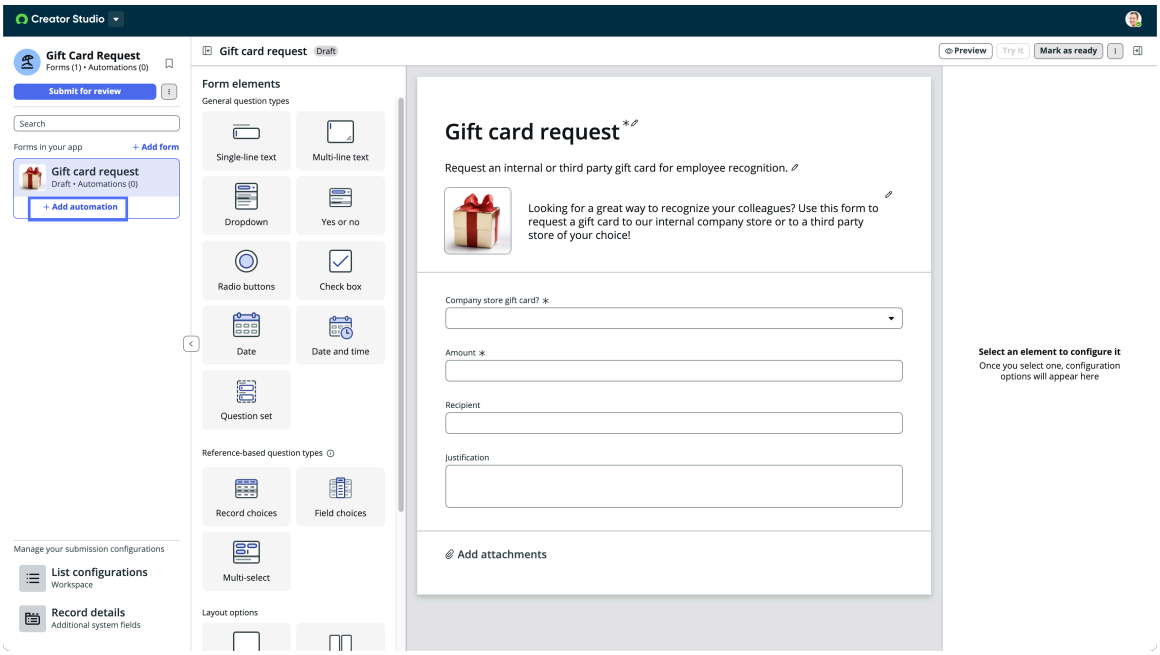
Role required: Creator Studio User

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Select **+ Add automation** for the Gift card request form.

Select to add an automation



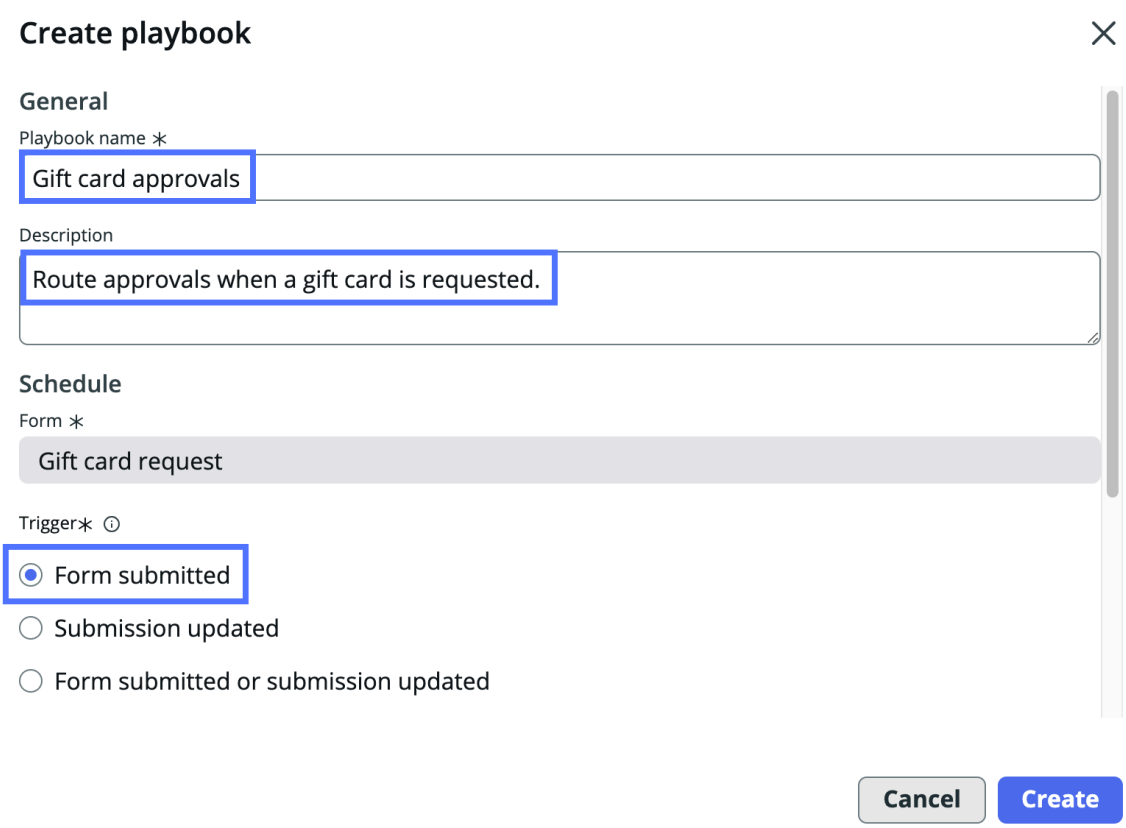
2. Enter the following values in the Create a playbook modal:

Create a playbook modal values

Field	Value
Playbook name	`Gift card approvals`
Description	`Route approvals when a gift card is requested.`
Form	Make sure the name of the form that you created (**Gift card request**) is selected.
Trigger	Leave the trigger as **Form submitted**.

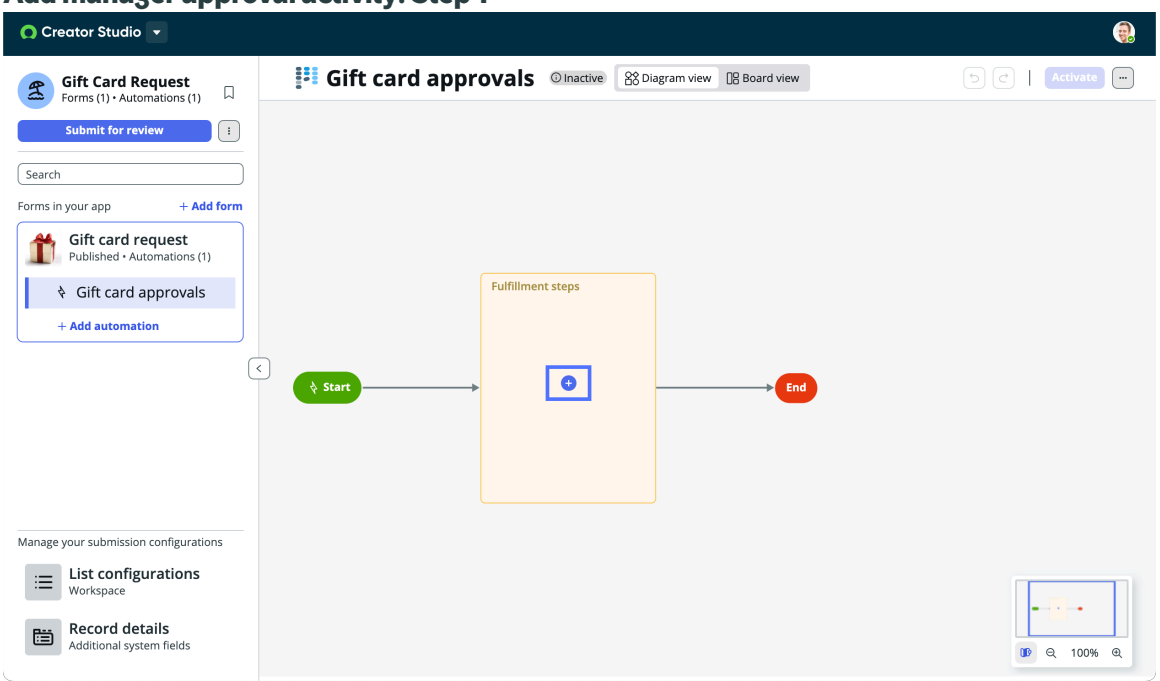
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Enter playbook details



3. Select the **Create** button.
4. Add the manager approval activity.
   a. Select the add icon ( ).

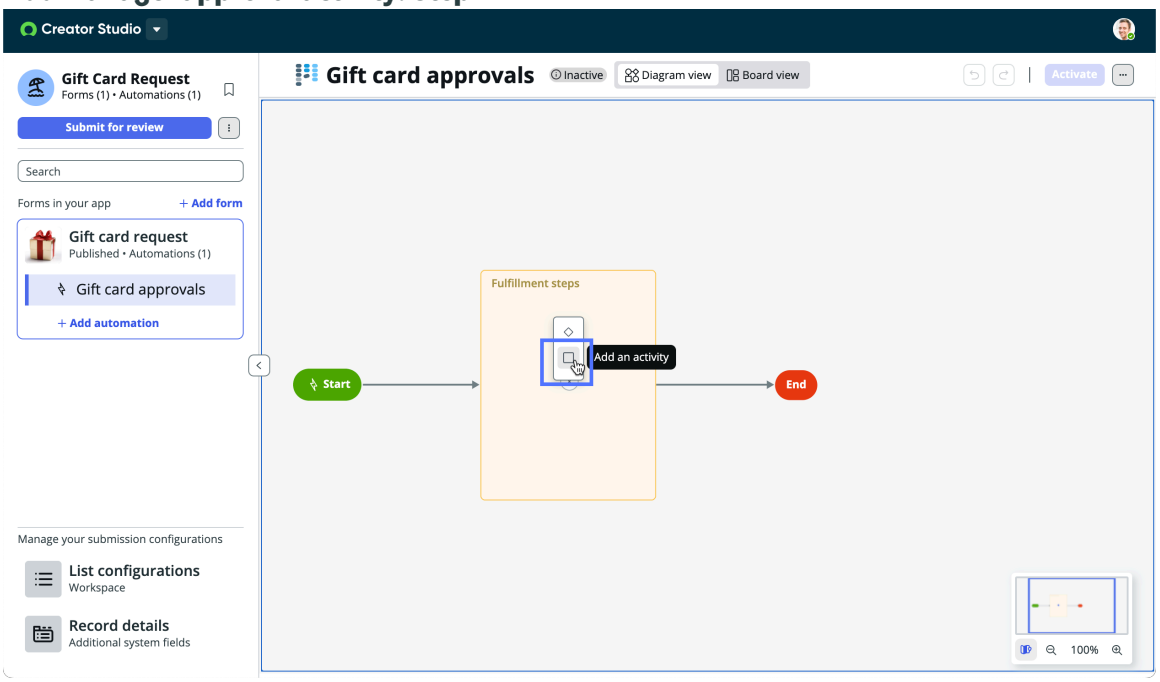
Add manager approval activity: Step 1



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

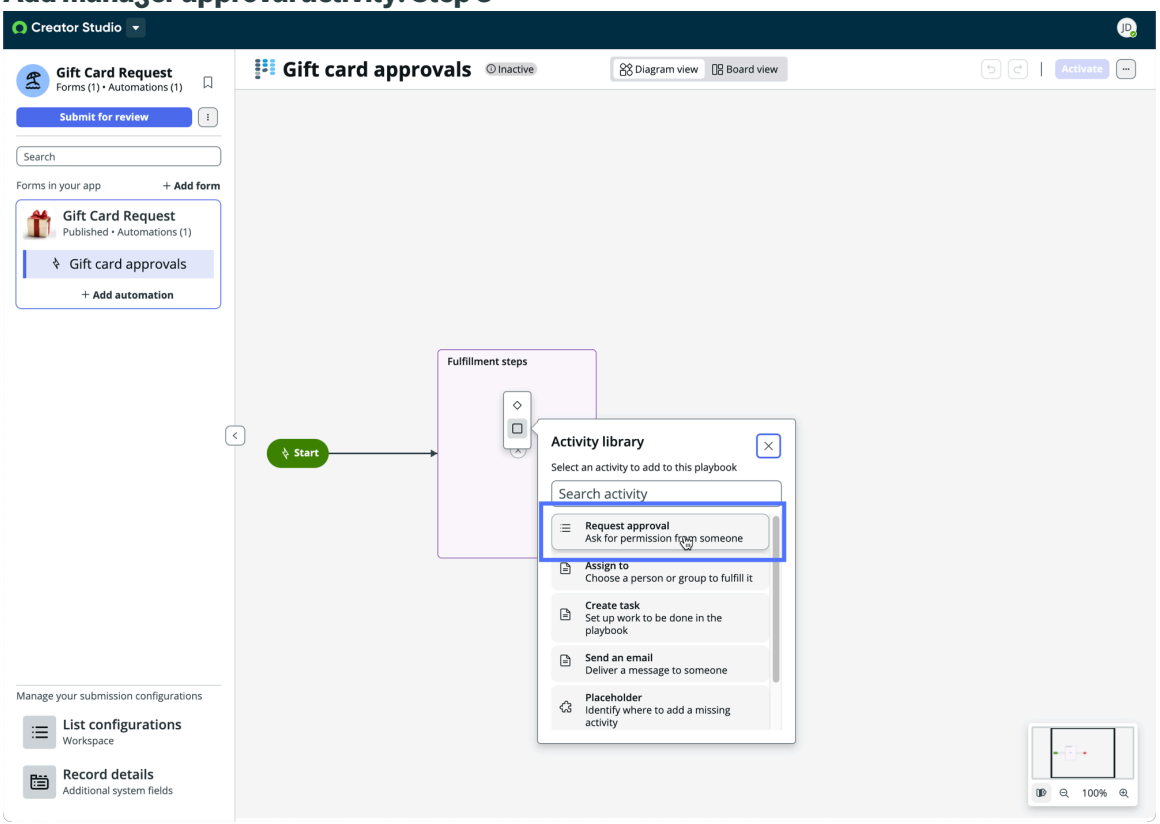
b. Select the Add an activity icon ☐.

Add manager approval activity: Step 2



c. Select the **Request approval** activity.

Add manager approval activity: Step 3



The Request approval properties panel appears, where you can define the activity.

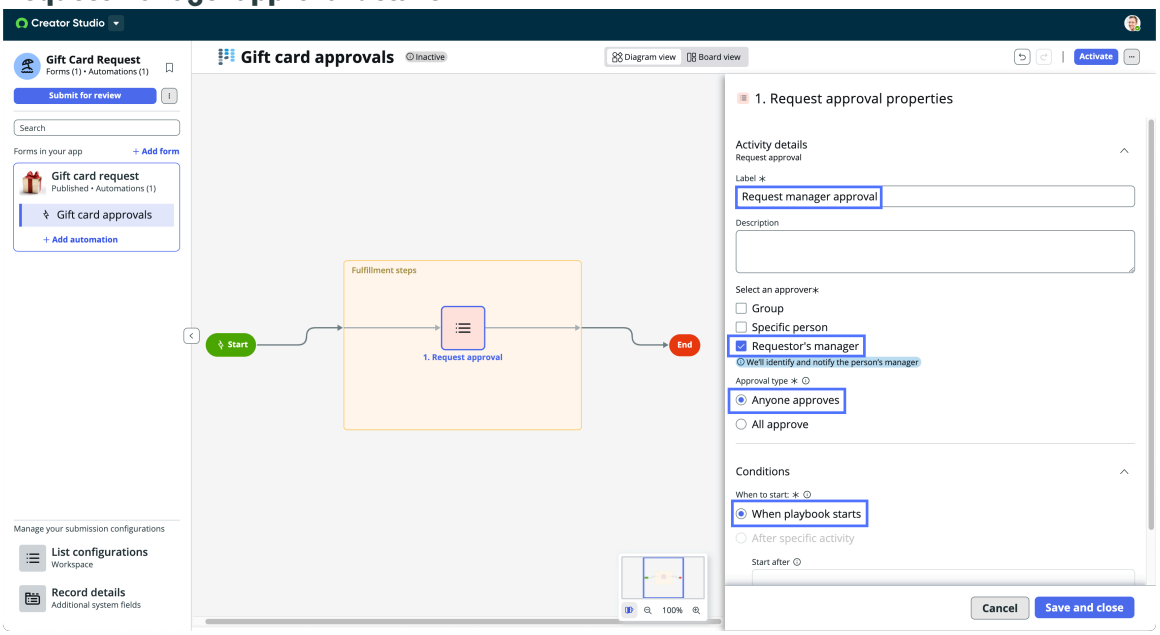
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

d. Enter the following activity details.

Manager approval activity detail values

Field	Value
Label	`Request manager approval`
Requester's manager	Select this option.
Approval type	Leave as **Anyone approves**.
Conditions	Select **When playbook starts**.

Request manager approval details



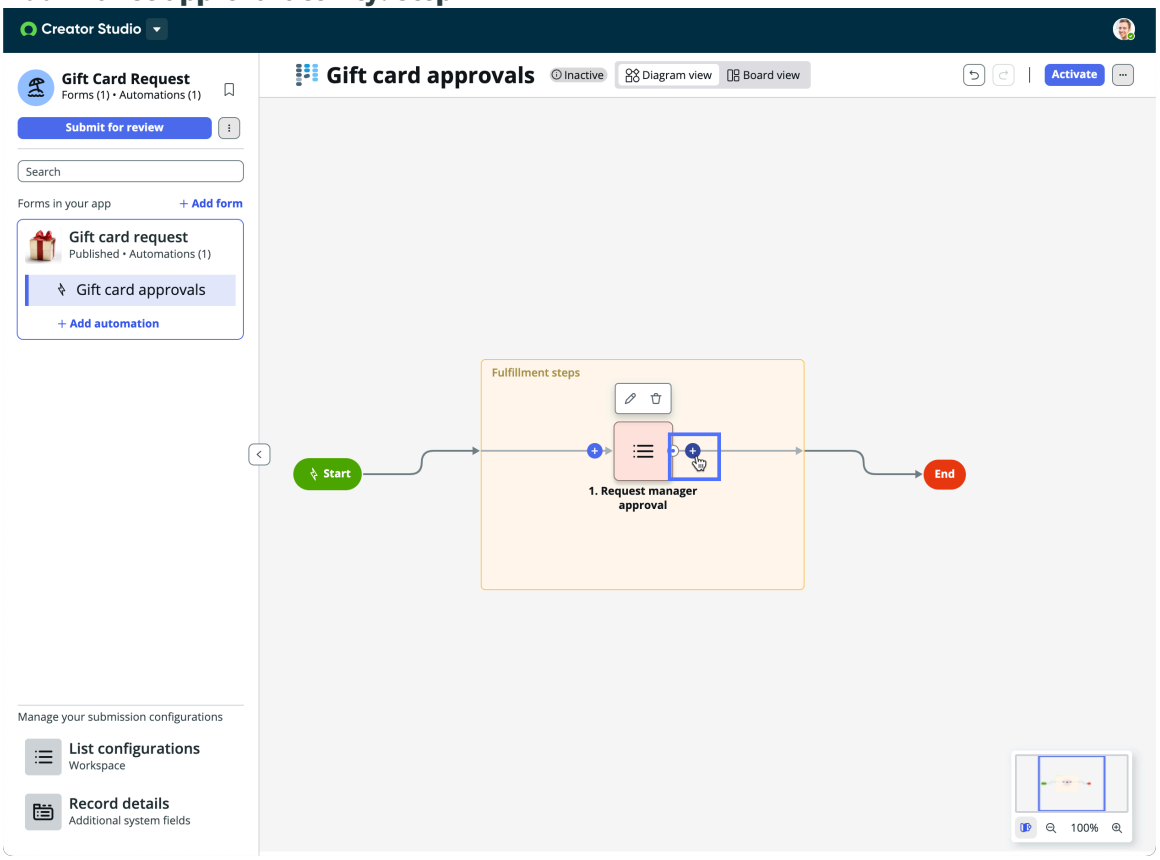
e. Select **Save and close**.

5. Add the finance approval activity.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. Select the add icon .

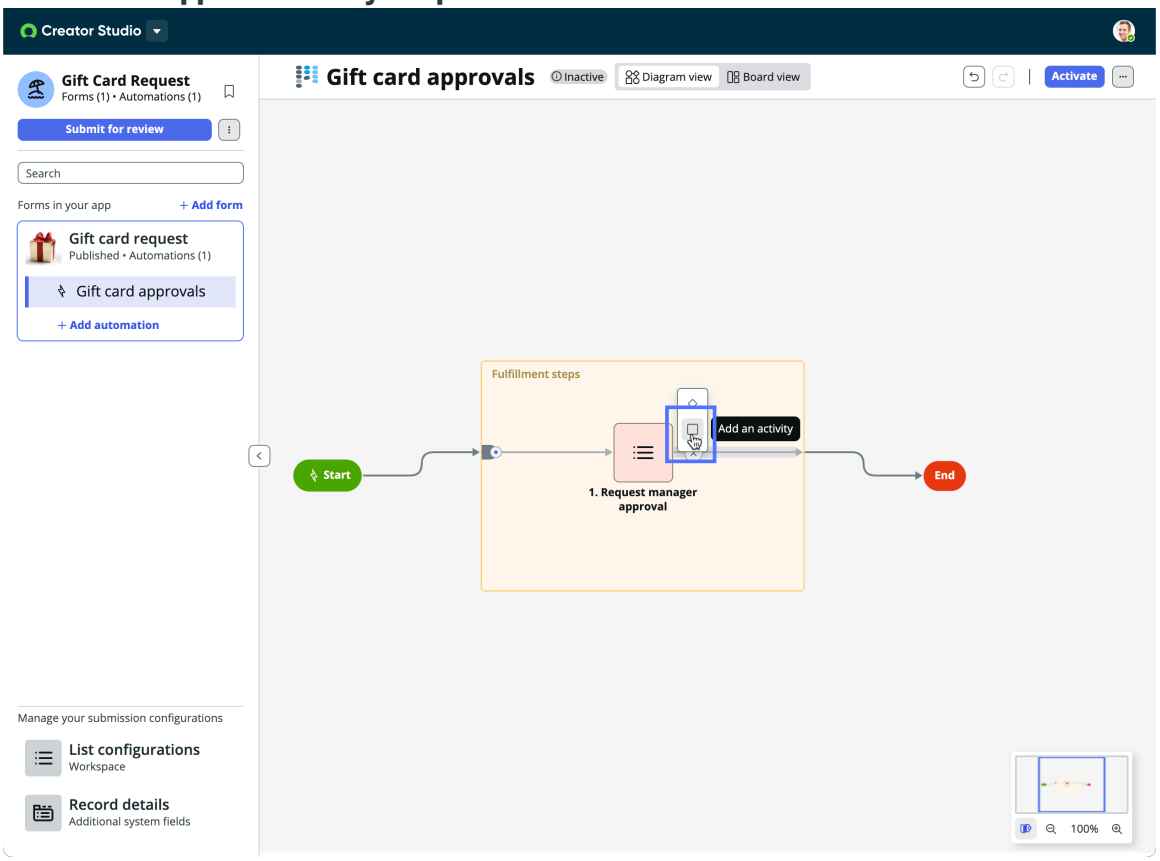
Add finance approval activity: Step 1



b. Select the add an activity icon .

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

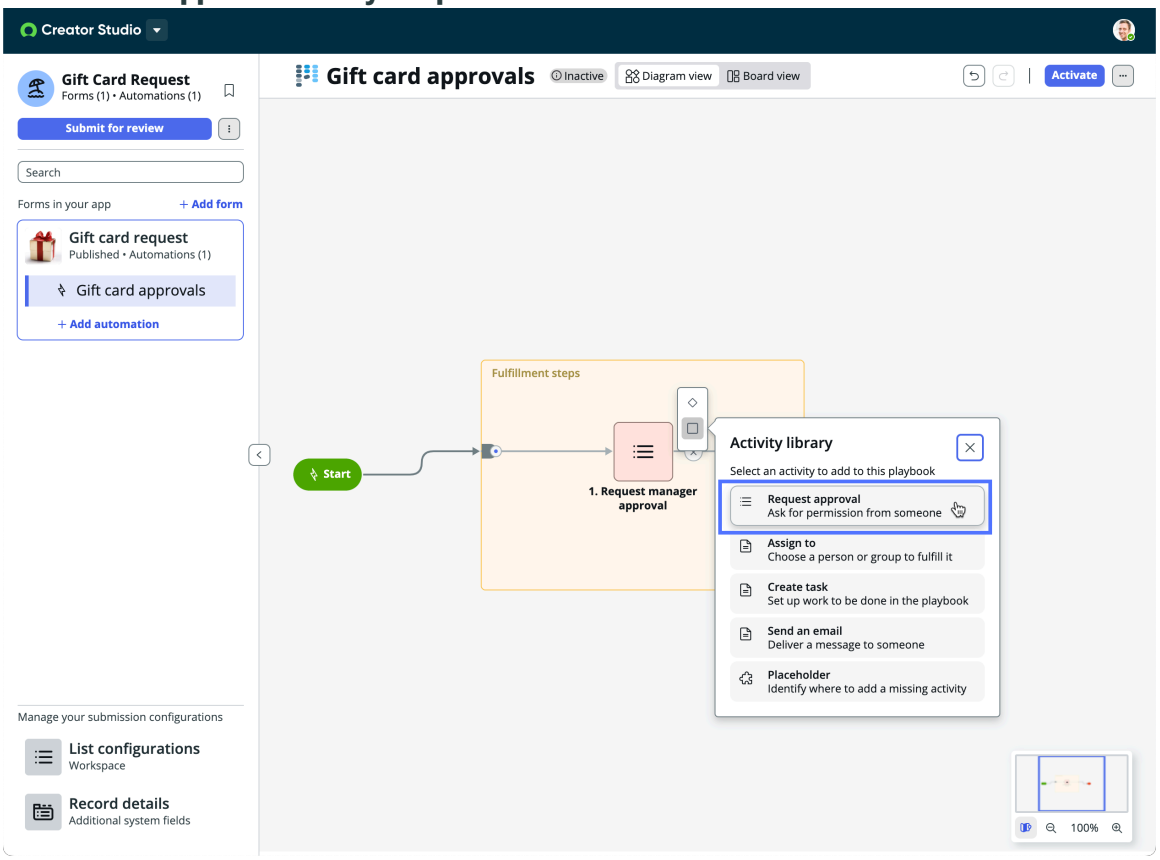
Add finance approval activity: Step 2



c. Select the **Request approval** activity.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add finance approval activity: Step 3



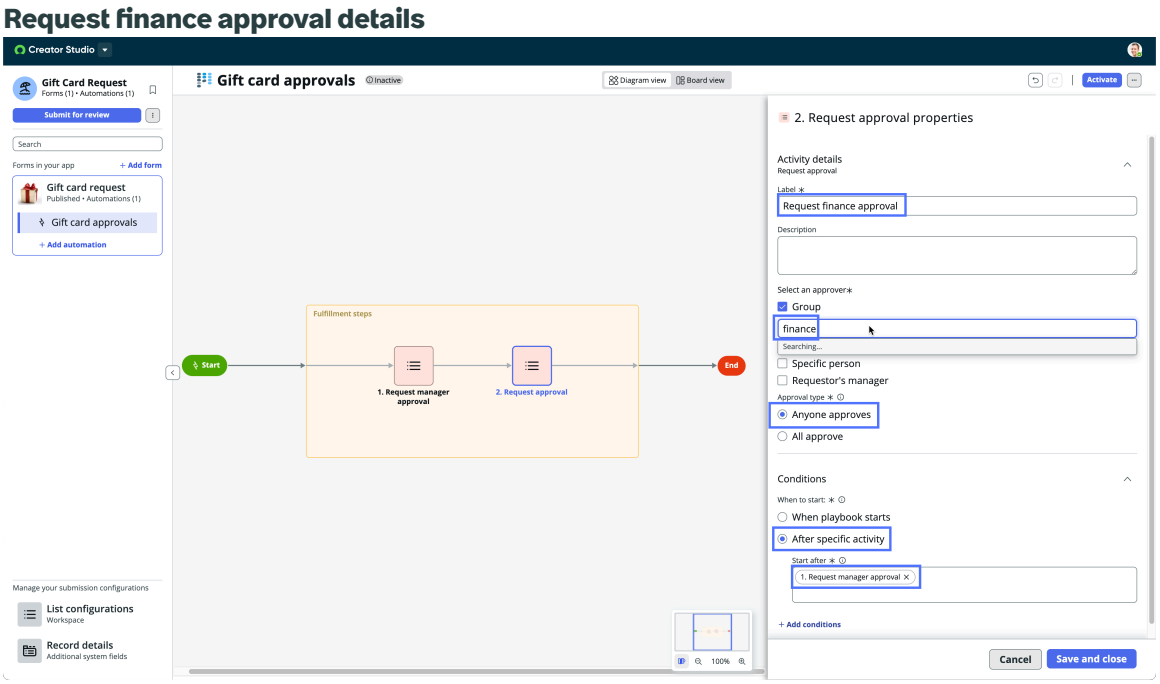
The Request approval properties panel appears, where you can define the activity.

d. Enter the following activity details.

Finance approval activity detail values

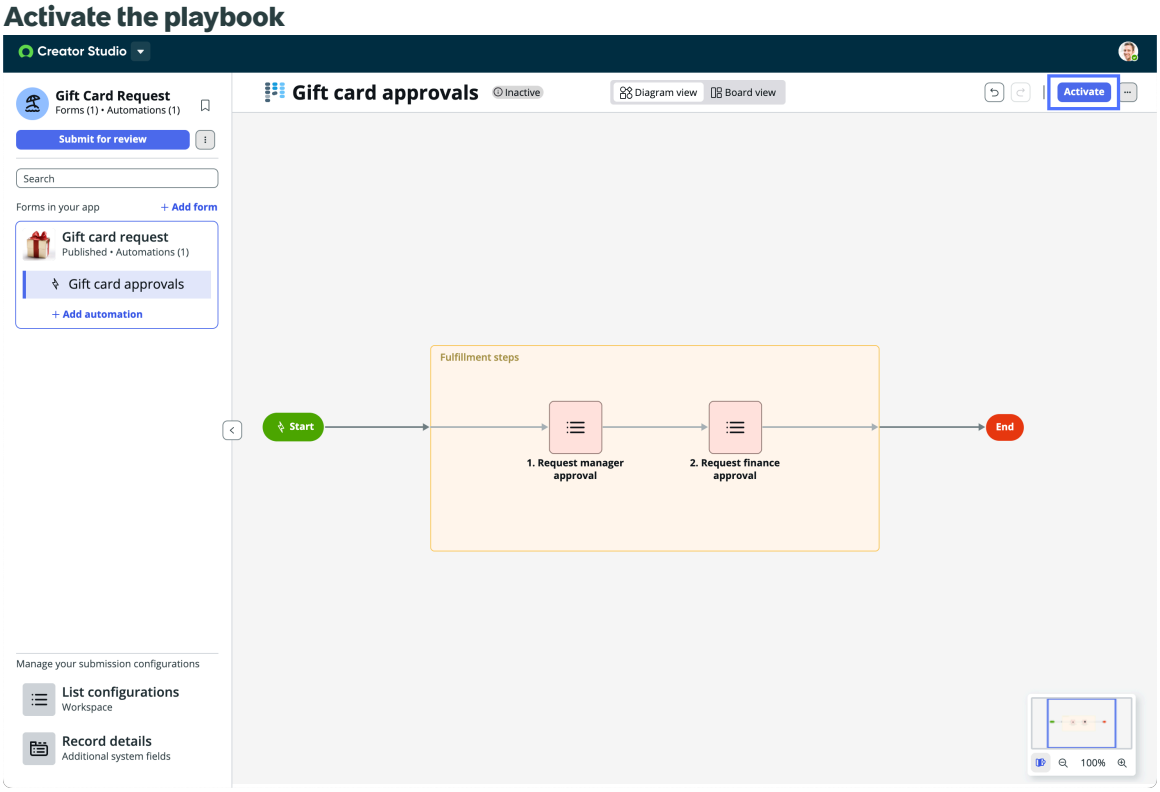
Field	Value
Label	Enter `Request finance approval`.
Group	Select this option.
Group	Enter `Finance group`.
Approval type	Leave as **Anyone approves**.
Conditions	Leave as **After specific activity** with **Request manager approval** selected.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



e. Select **Save and close**.

6. Activate the playbook so that it can run when the form is submitted by selecting the **Activate** button.



Result

Great! You've finished adding a playbook. Next, you can customize the workspace configuration where fulfillers work on submitted requests.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Tutorial part 5: Customize the fulfillment workspace configuration

Customize the lists and submitted records in the workspace where fulfillers can process gift card requests from the. In this list customization, you add a column to view the assignment group.

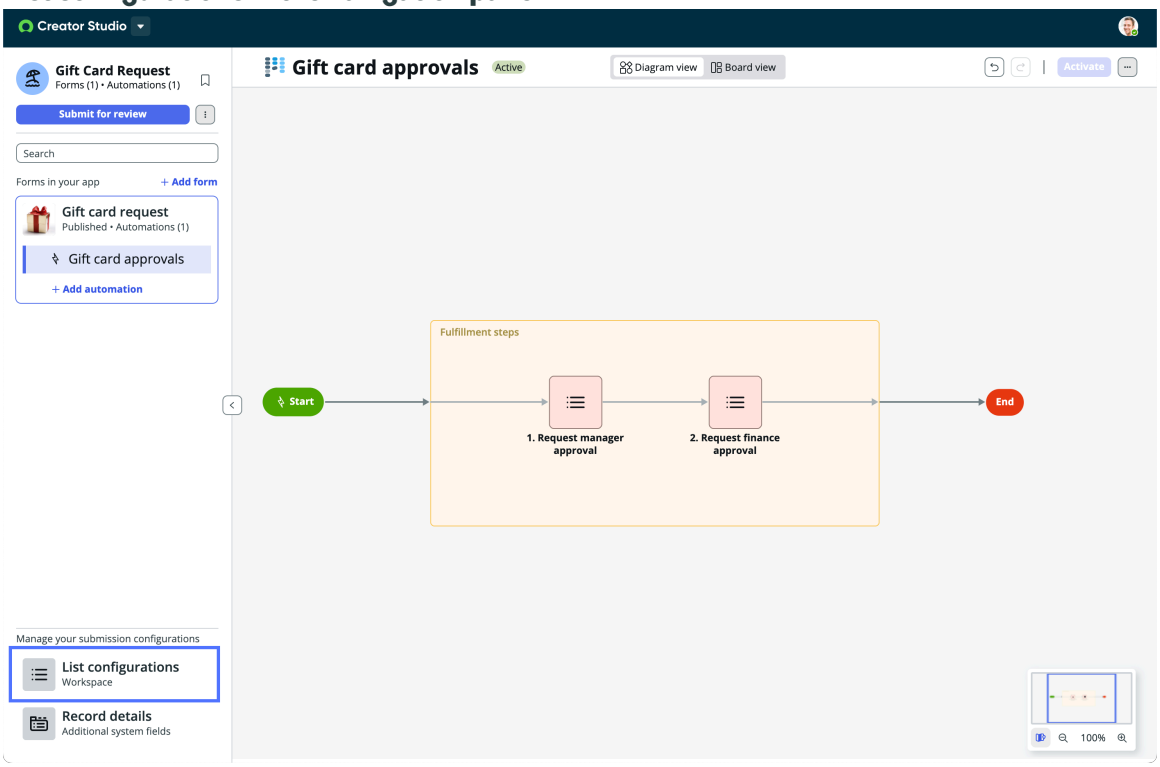
Before you begin

Role required: Creator Studio User

Procedure

1. Select the **List configurations** section in the navigation panel.

List configurations in the navigation panel

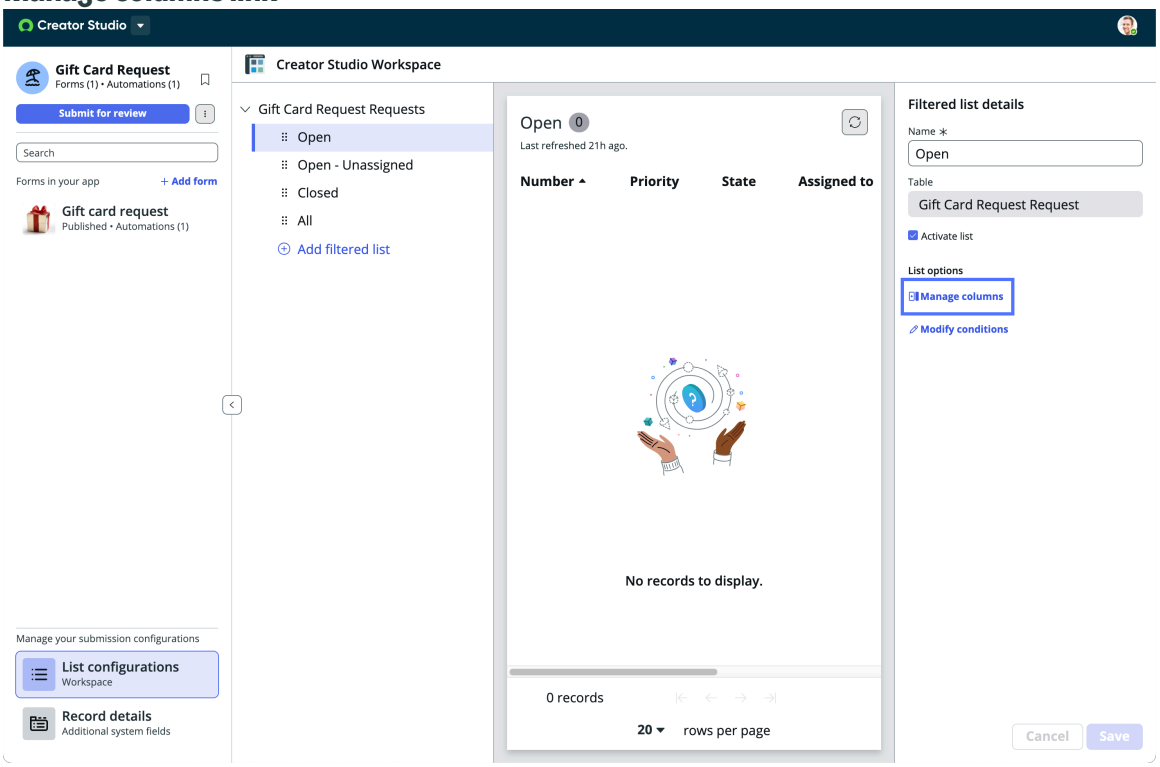


2. Add the assignment group to the workspace.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. Select the **Manage columns** link in the Filtered list details panel.

Manage columns link

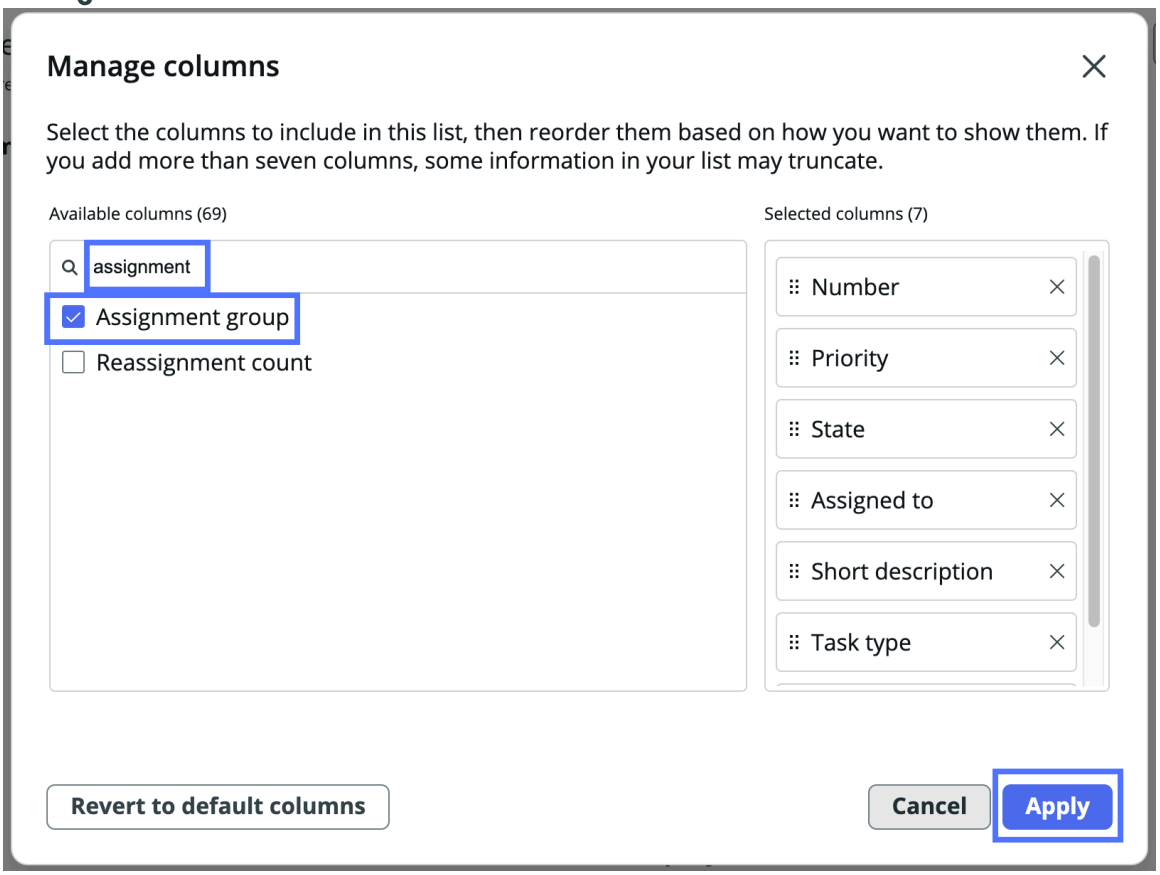


b. Search for `Assignment group` in the Available columns.

c. Select the **Assignment group**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

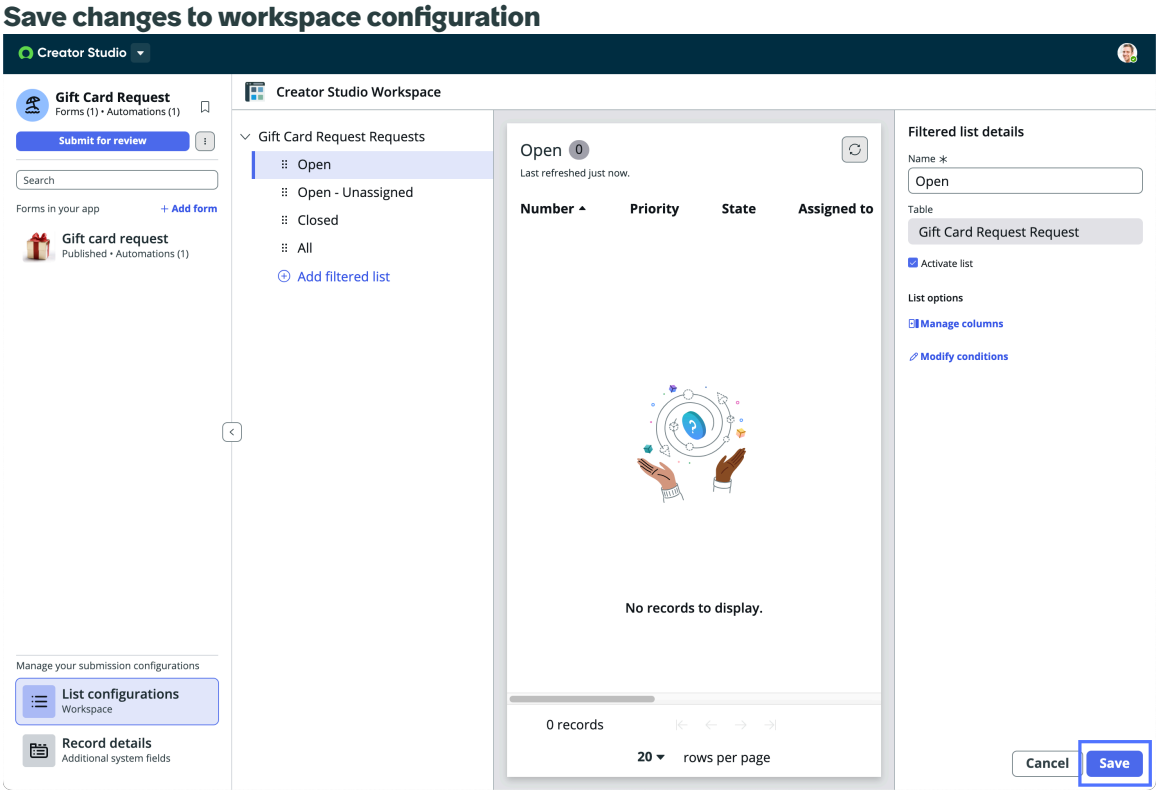
Manage the columns to add a column



d. Select the **Apply** button.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

e. Select the **Save** button in the Filtered list details panel.



Result

That's it, you've built an app! Now it's time to submit it for review and deployment.

Tutorial part 6: Test and submit the app for review

Test your app and then request that an admin review and deploy the app to production.

Before you begin

A pipeline for deployment must be configured before you can submit the gift card app for review.

Role required: Creator Studio User

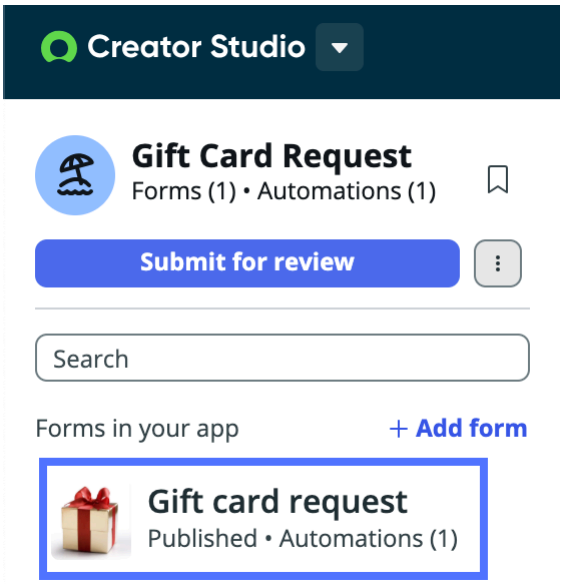
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Test how the app works to make sure that it's ready by selecting the **Try it** button.

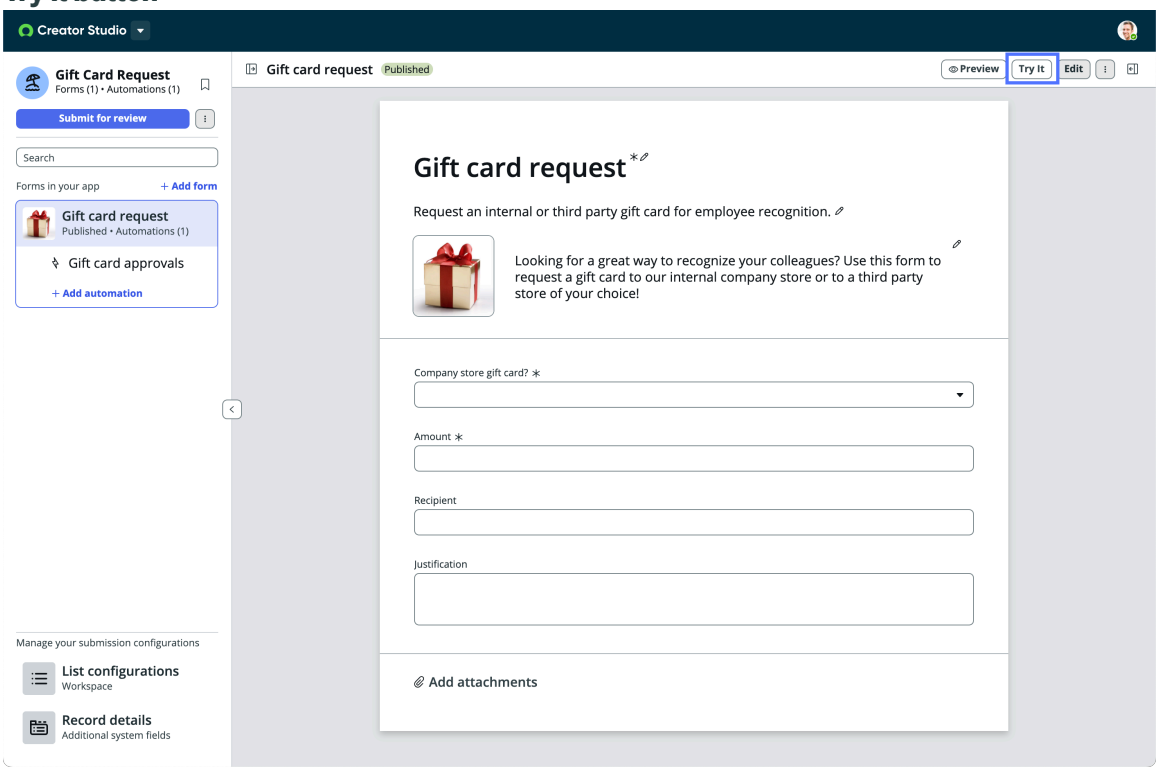
   a. Select the **Gift card request** form from the navigation panel.

Select the form to test it



   b. Select the **Try it** button.

Try it button



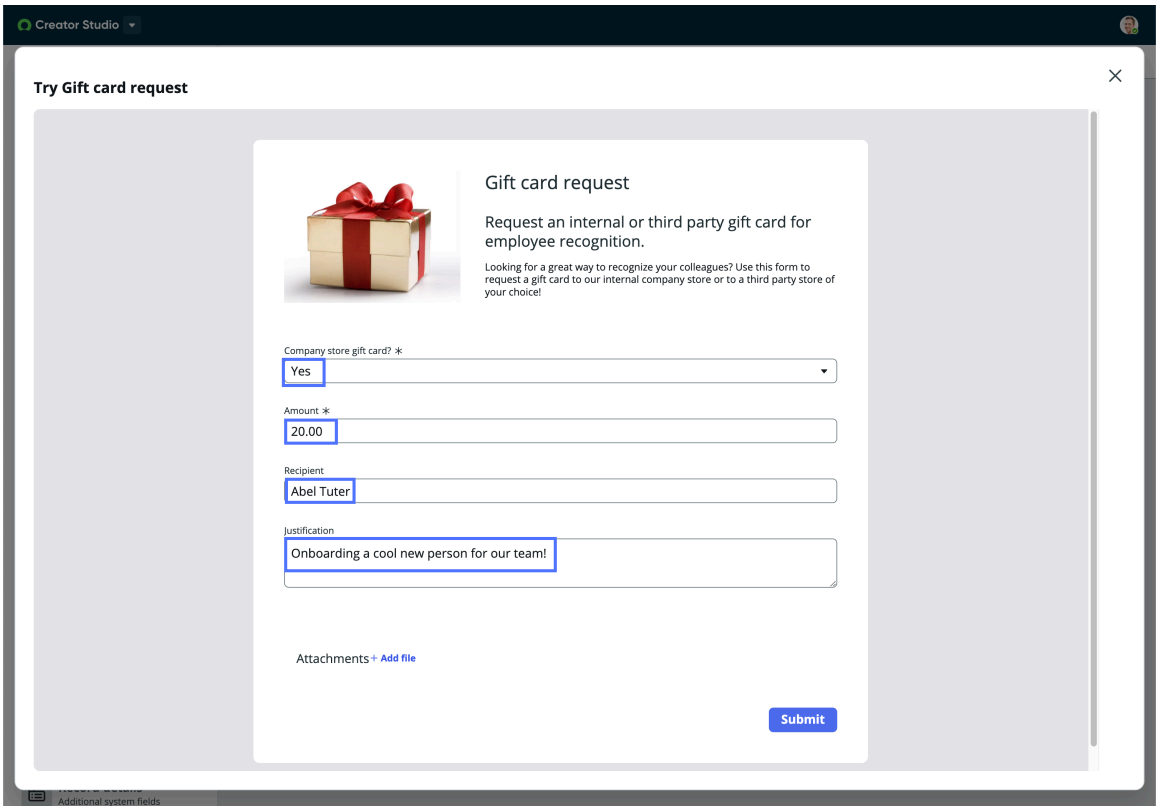
   c. Complete the form by entering the following values for each field.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Try it values for Gift card request form

Field	Value
Company gift store card?	`Yes`
Amount	`20.00`
Recipient	Choose a recipient from the list of people.
Justification	`Onboarding a cool new person for our team!`

Test the form



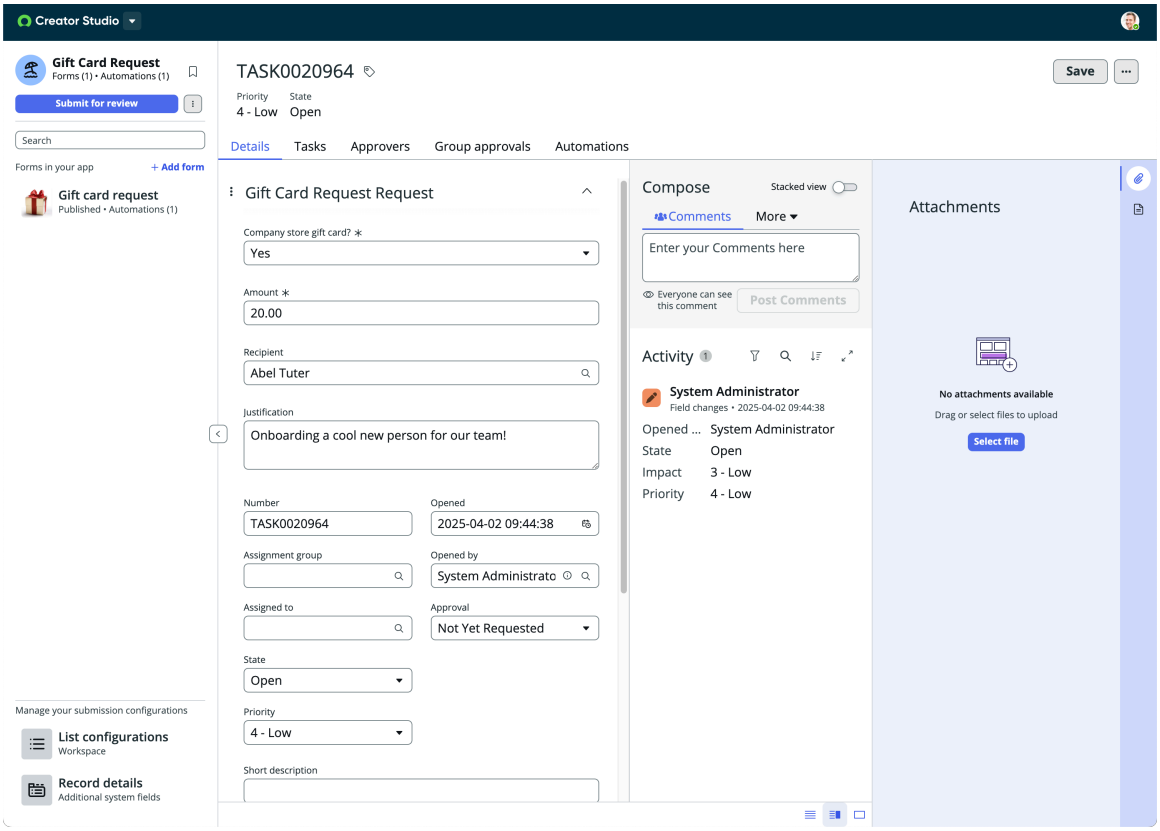
d. Select the **Submit** button when you're done.

After you submit your responses, the ServiceNow AI Platform runs any playbooks associated with the form.

The record that your submitted form creates appears in Creator Studio. You can view the results of the playbooks and interact with the record to see how it appears in the Request App Workspace.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

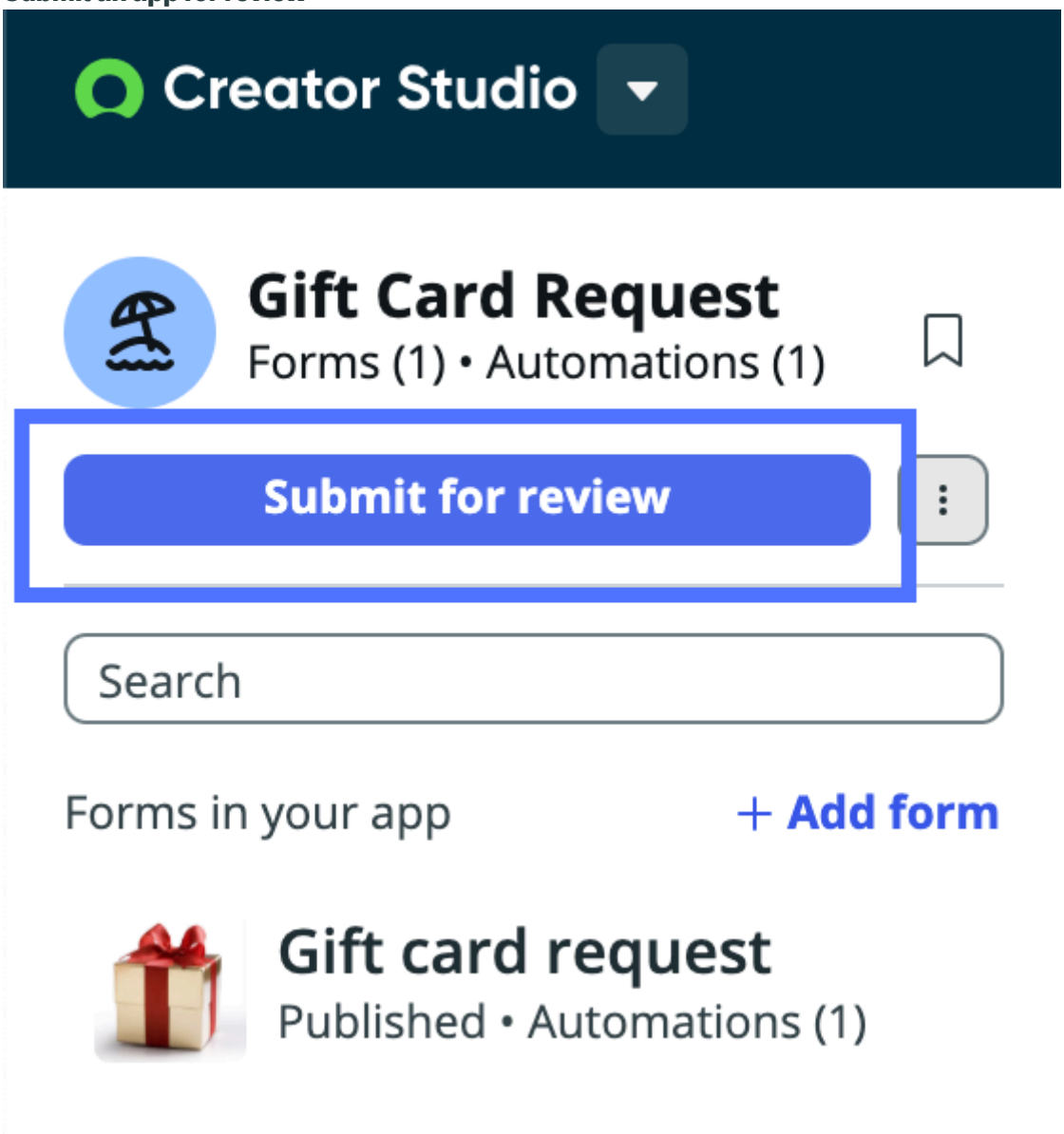
View the submitted record



2. Select the **Submit for review** button.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

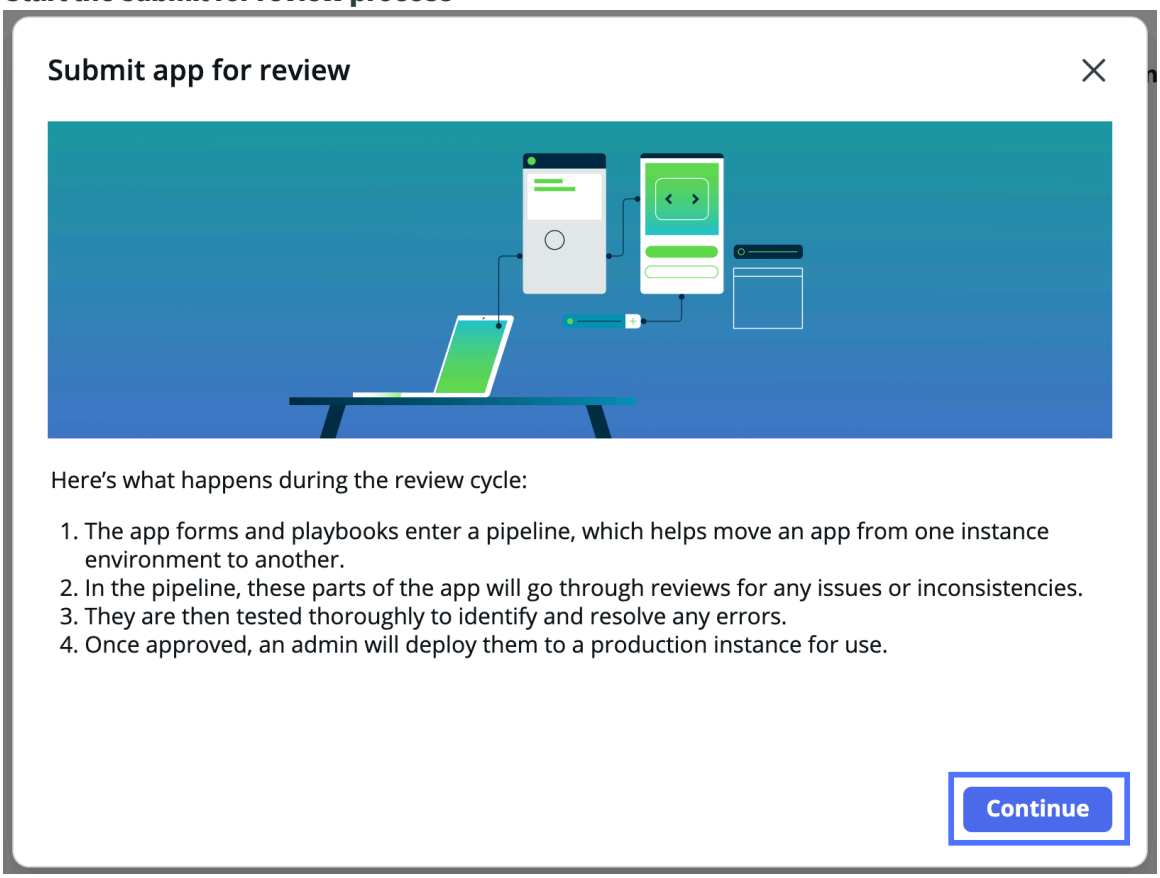
Submit an app for review



3. Select **Continue** on the Submit app for review modal.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

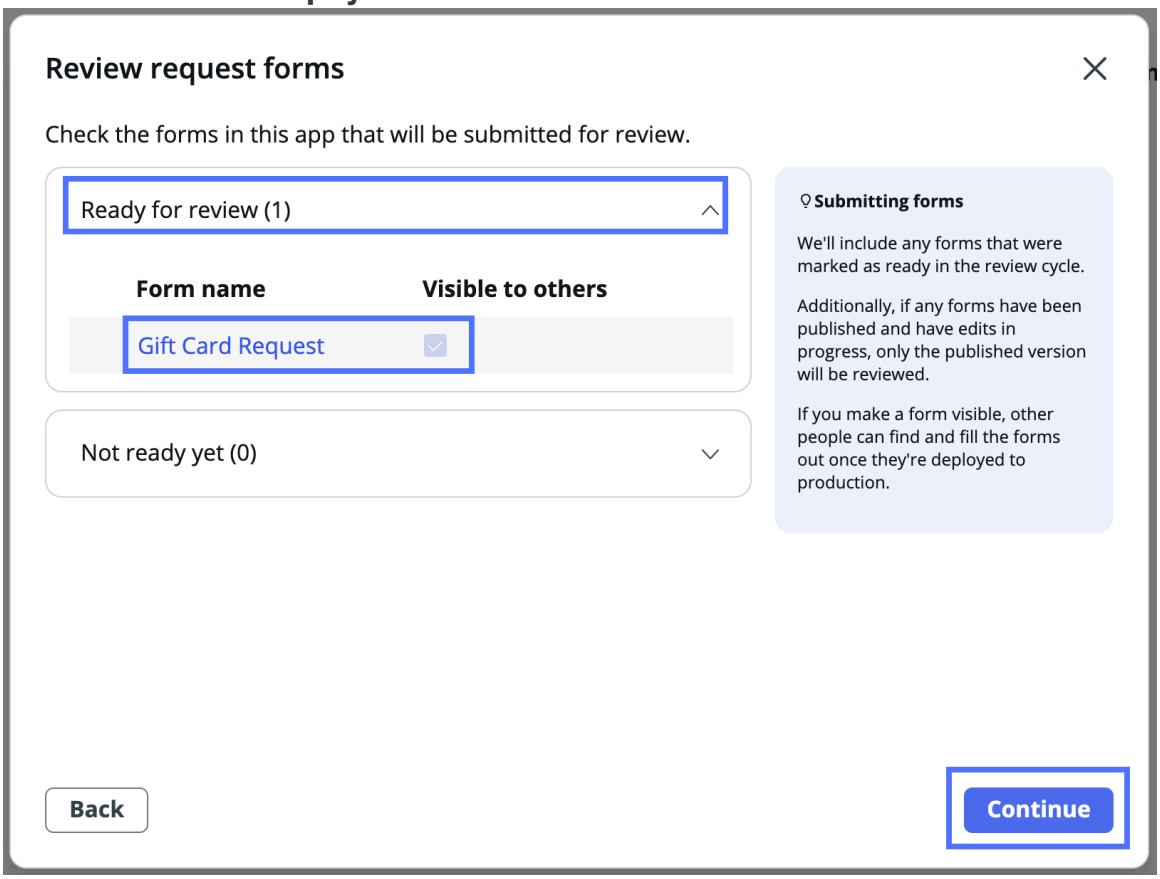
Start the submit for review process



4. Check the form to be reviewed for deployment in the Review request forms modal.
   a. Expand the **Ready for review** section.
   b. Ensure that the **Visible to others** option is selected for the **Gift Card Request** form.
   c. Select **Continue**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

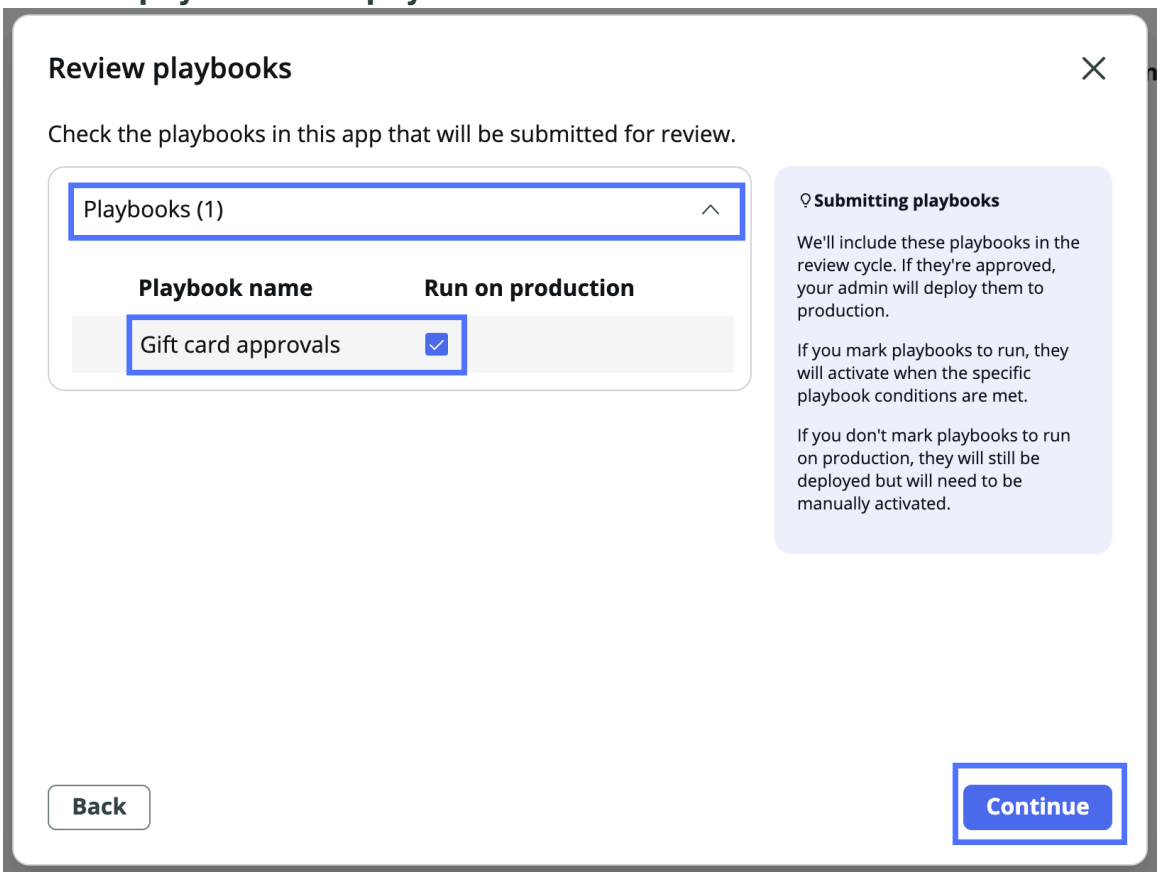
Check the form to be deployed



5. Check the form to be reviewed for deployment in the Review playbooks modal.
   a. Expand the **Playbooks** section.
   b. Ensure that the **Run on production** option is selected for the **Gift card approvals** form.
   c. Select **Continue**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

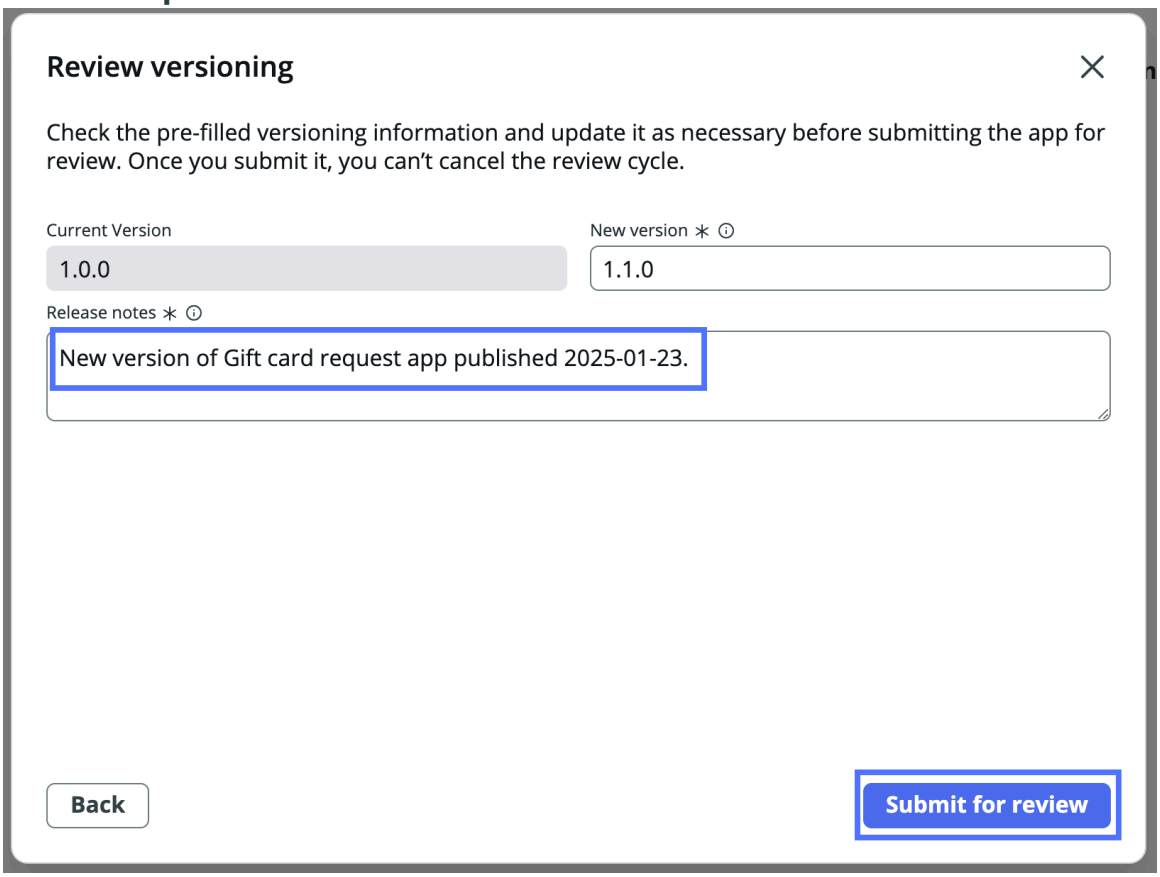
Check the playbook to be deployed



6. Confirm that the information in the Review versioning modal is correct, and add the **Release notes**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Check and update release version and submit for review



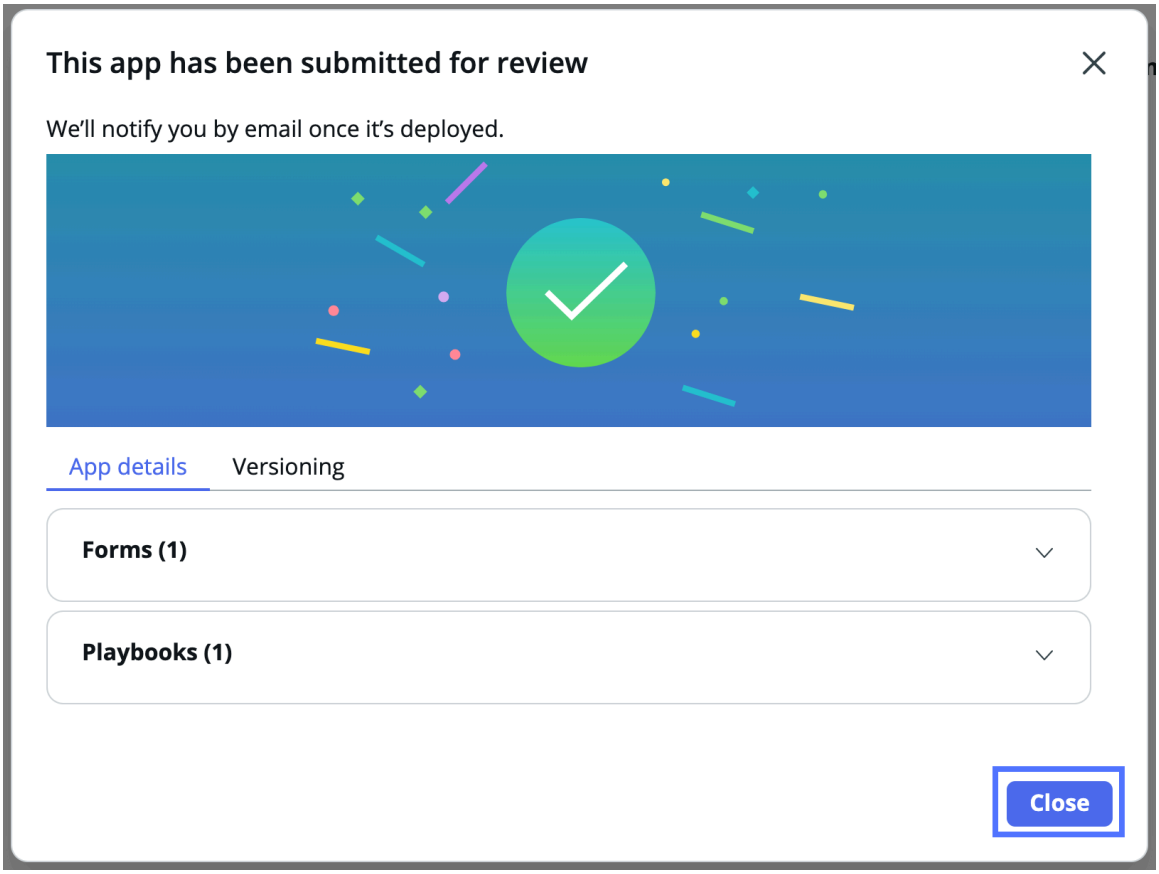
7. Select **Submit for review**.

Result

Congratulations, you have created your gift card request app! Now your admin must review and deploy it.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Confirmation of review submission



Tutorial part 7: Admin deploys the app

Impersonate an admin to deploy the gift card request app that you created.

Before you begin

If you can't impersonate an admin, ask your local admin to deploy the app for you.

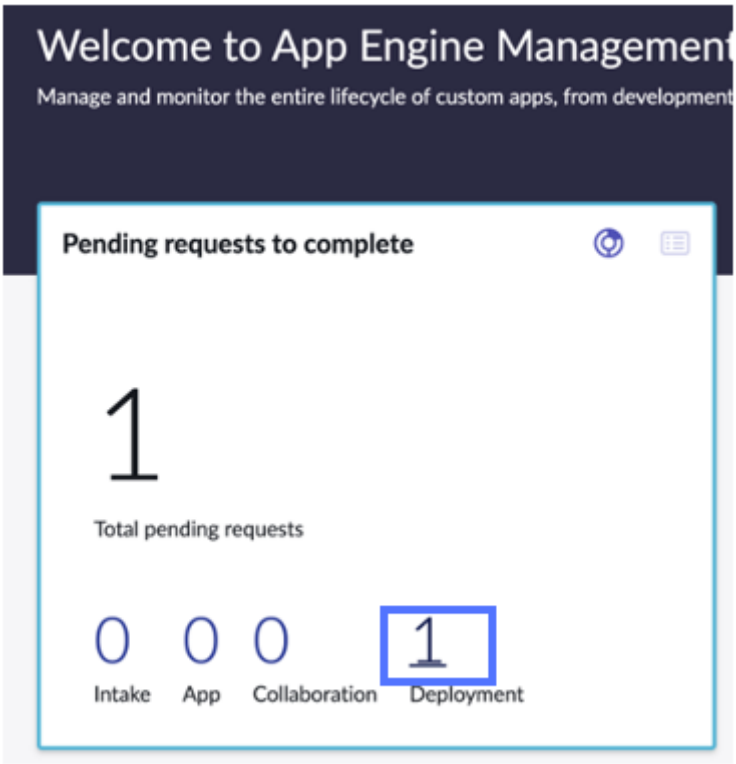
Role required: admin

Procedure

1. In the ServiceNow AI Platform application banner, select your profile icon and impersonate an admin.
2. Navigate to **All** > **App Engine** > **App Engine Management Center**.
3. Select the number **1** in the Pending requests to complete section.

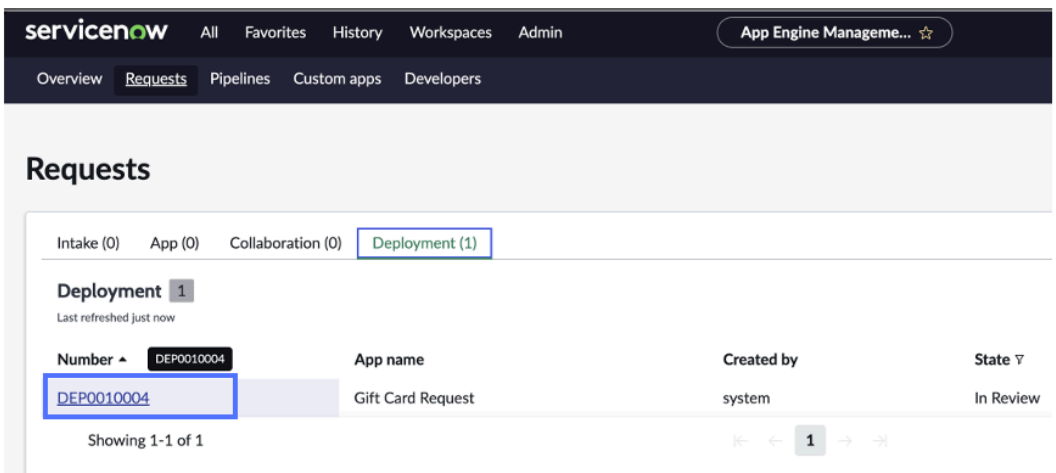
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Apps ready for deployment



4. Open the deployment record by selecting its **Number**.

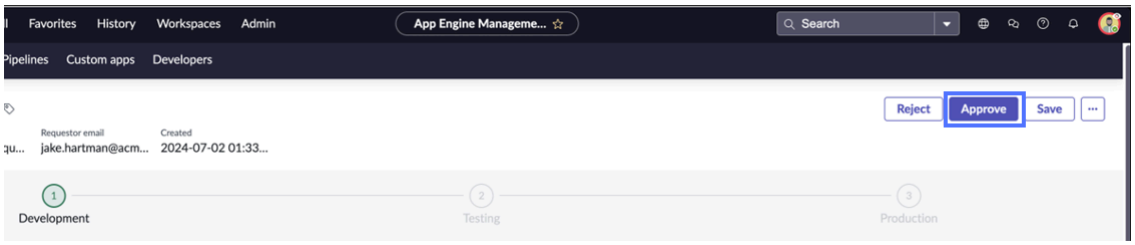
List of apps to deploy



5. Select the **Approve** button.

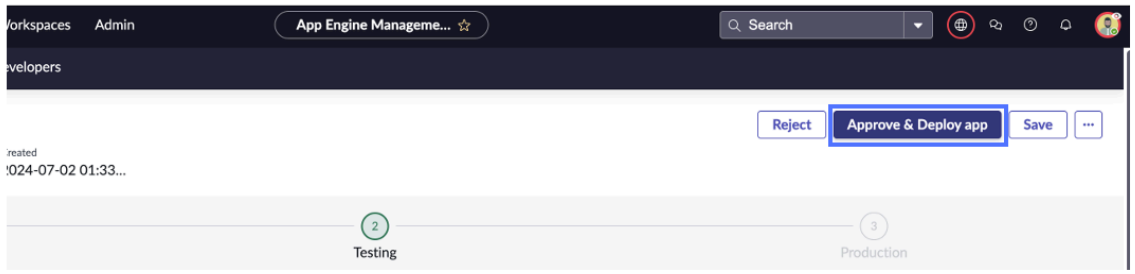
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Approve button for deployment



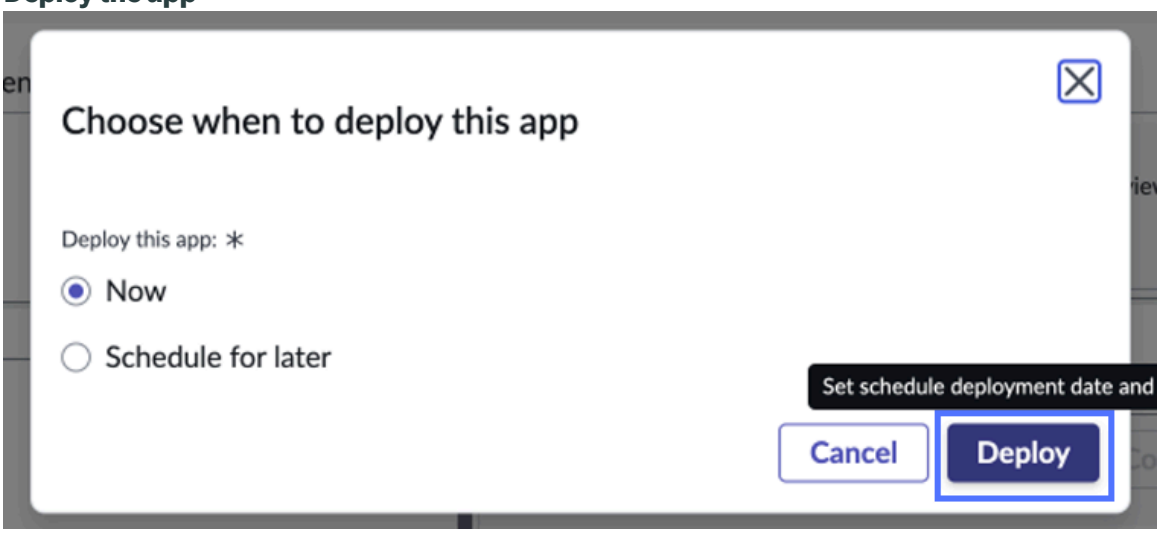
6. Select **Save**.
7. Reload the page.
8. Select the **Approve and deploy app** button.

Approve and deploy button for deployment



9. Select the **Deploy** button.

Deploy the app

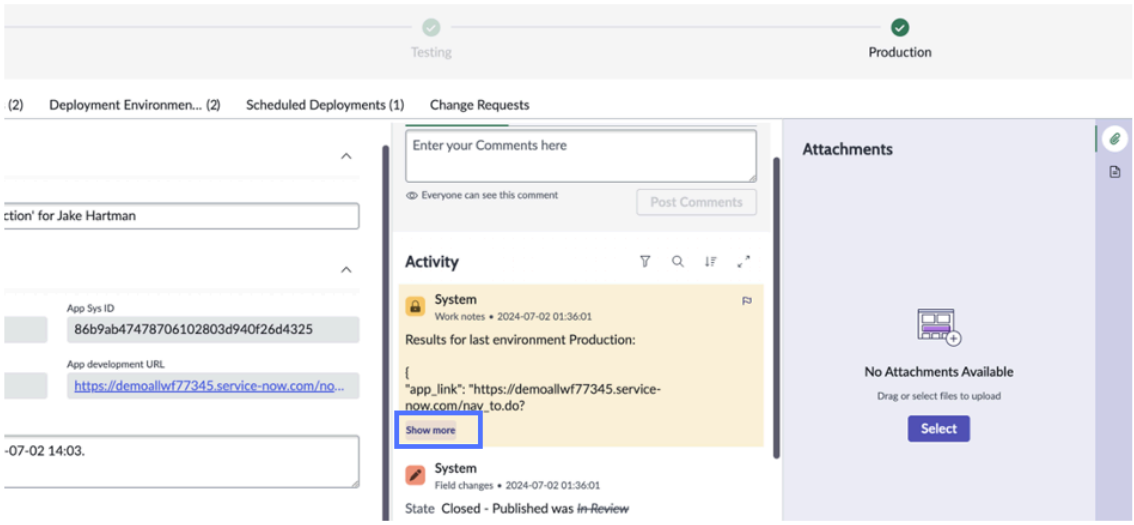


Result

The app has been deployed to production. Select **Show more** in the activity section if you want to view details for the app.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Verify the deployment details



You can end the admin impersonation now.

Tutorial part 8: Use the app to request a gift card

Use the deployed gift card request app that you just created to request a gift card.

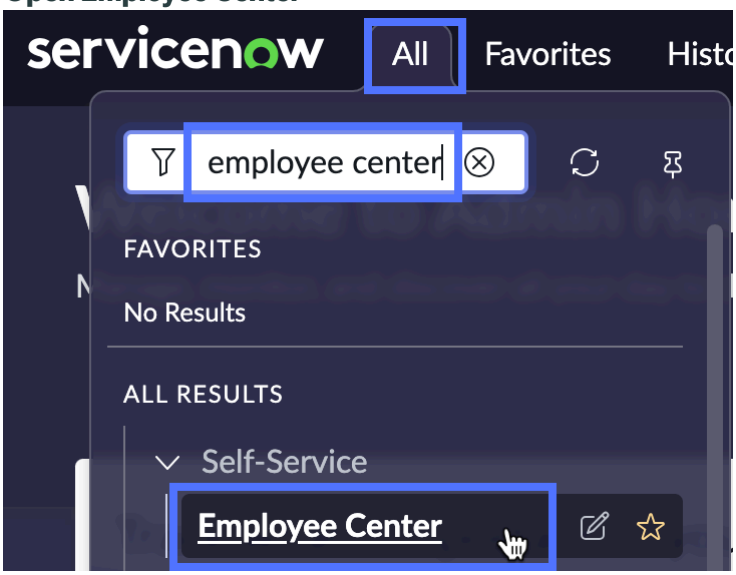
Before you begin

Role required: ServiceNow AI Platform user

Procedure

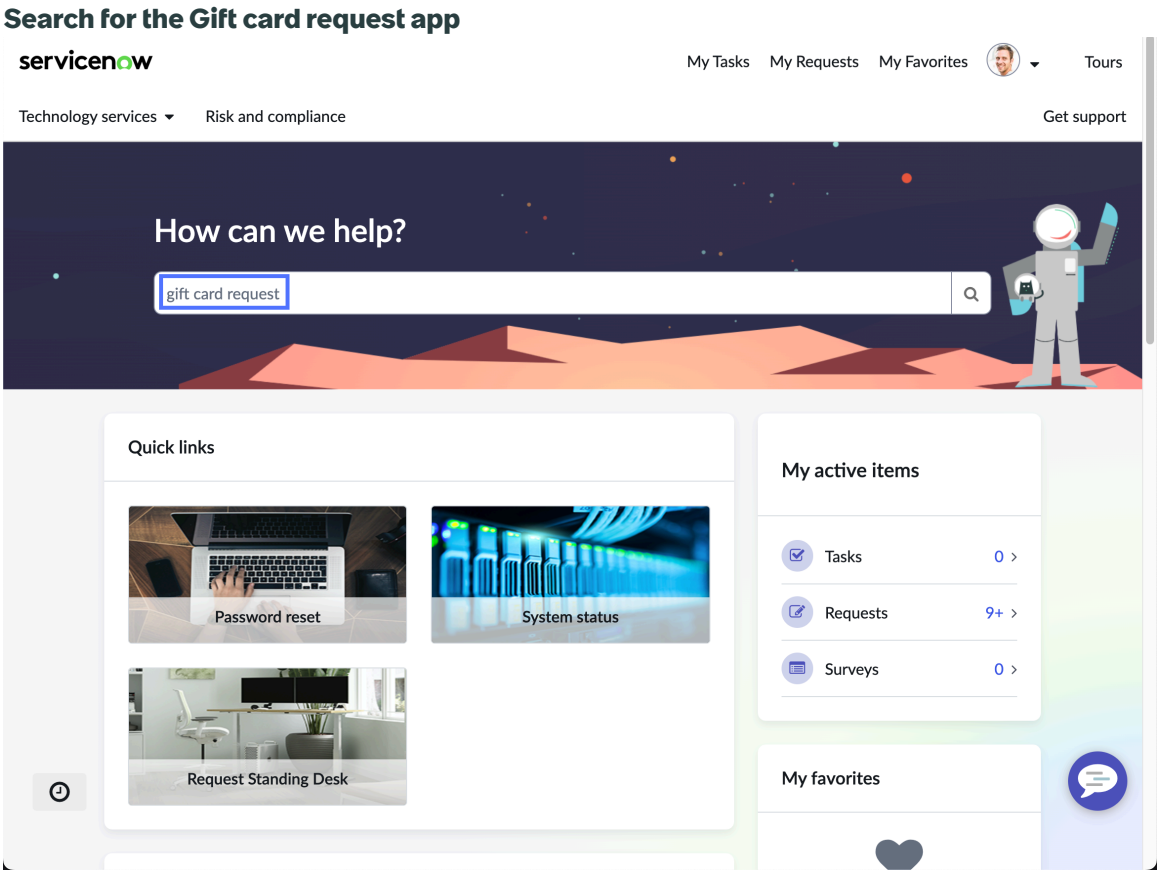
1. Navigate to **All** > **Self-Service** > **Employee Center**.

Open Employee Center

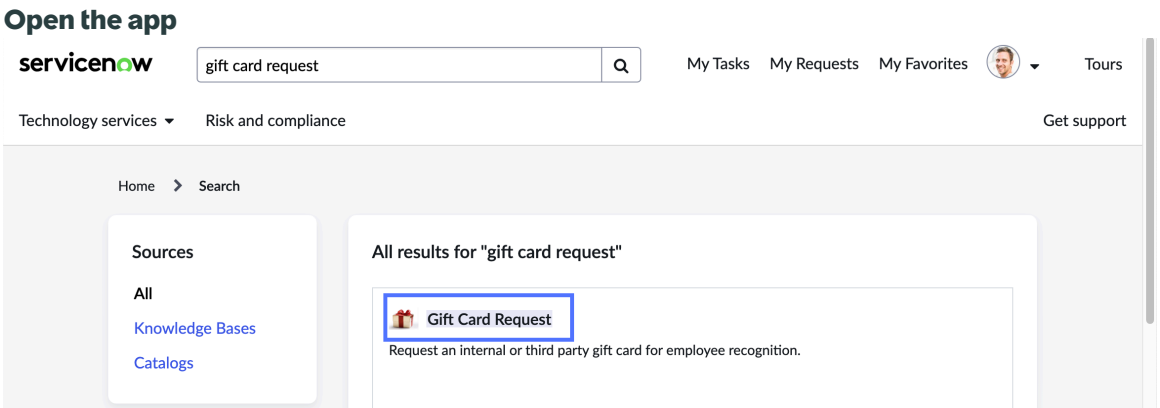


2. Search for `Gift card request` in the search bar.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

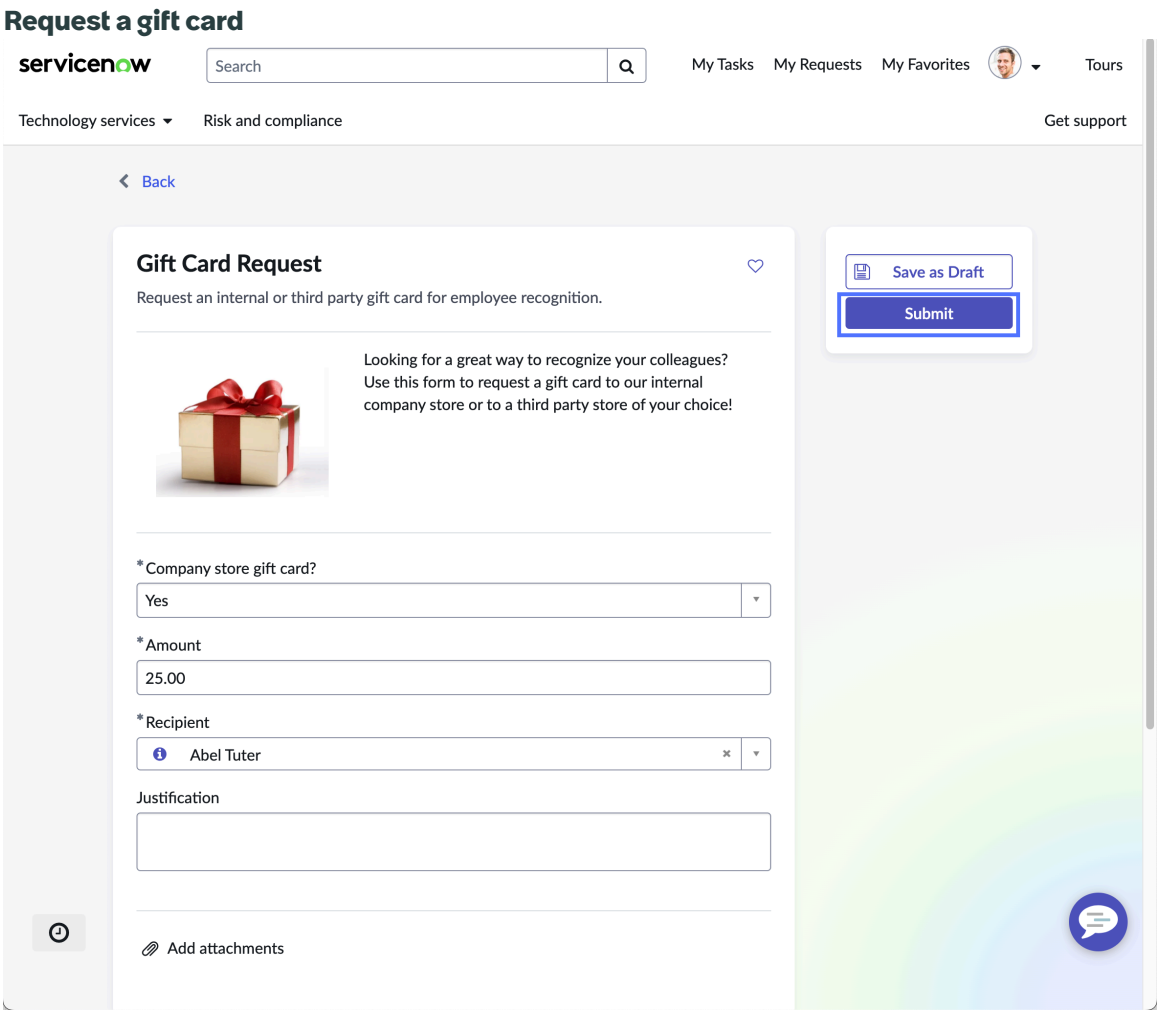


3. Select the **Gift card request** app that appears in the search results.



4. Fill out the gift card request form to request a gift card.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



5. Select the **Submit** button.

Tutorial part 9: Verify the gift card request

Verify that your app is working by checking the Request App Workspace to view the gift card request that you submitted.

Before you begin

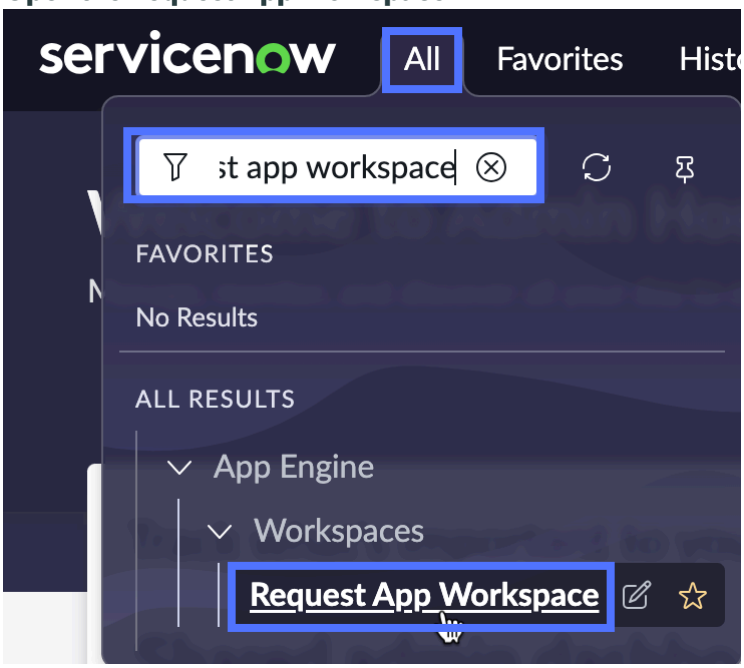
Role required: admin or the app's agent role, for example, x_snc_app_name.agent

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

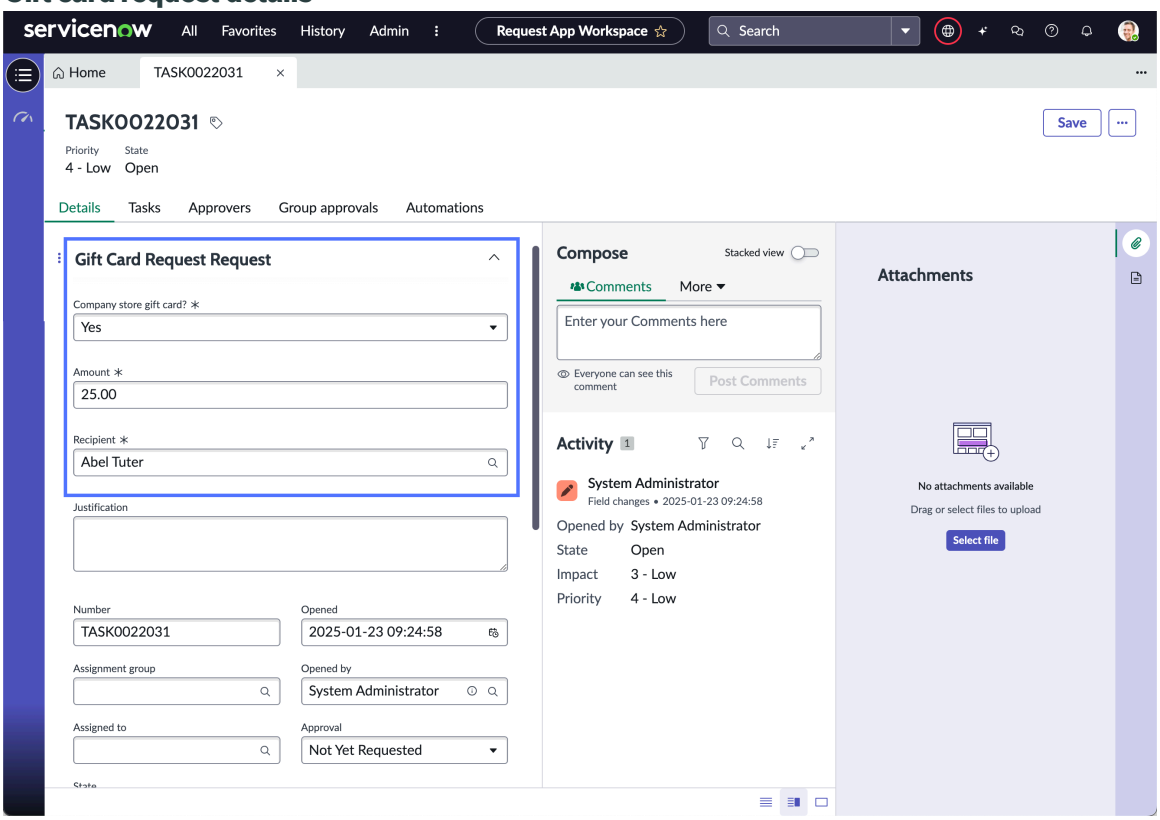
1. Navigate to **All** > **Workspaces** > **Request App Workspace**.

Open the Request App Workspace



2. Find the gift card request that you submitted and select its **Number** to open it.
3. View the gift card request details.

Gift card request details



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Result
Hooray! You have completed the gift card request app tutorial.

Building apps with Creator Studio

By now, your admin has set up Creator Studio, given you permission to build an app, and you've got a great idea for a request-fulfillment app! How should you think about using Creator Studio to create your app?

What apps are in ServiceNow

A ServiceNow app is a package that performs a specific task for a specified group of users. Think of an app as a container with a set of rules around who can access and edit it. For example, ServiceNow apps can include an API, a table, a workspace, a form, a flow, or any combination of those things.

A Creator Studio app uses forms and automation to enable users to request something and move the request through the approval process. Creator Studio also enables you to the see the fulfiller's view of the Request App Workspace for the app.

Create forms to gather request information

When finished, your app will look like a catalog of items that people can request. To create your app, you'll:

1. Select one of the available catalog templates.
2. Customize a form to be included in the catalog using a WYSIWYG editor.

   A form describes one item in the catalog and provides space for questions that a requester will answer to specify details about what they want, such as the model number of a laptop. If your catalog needs multiple items, you can create multiple forms.

   Note: You can create a form using AI. To learn more, see Generating forms with AI.

Automate workflows and simplify tasks with playbooks

Streamlining your request-fulfillment process goes beyond just collecting information. Creator Studio empowers you to automate tasks with powerful playbooks. Think of them as pre-programmed workflows that kick in after a form is submitted.

Here are some things that playbooks can do:

- **Effortless approval routing:** Automatically route requests to the right person for approval, eliminating manual steps and bottlenecks. Imagine a laptop request automatically going to the IT manager for a thumbs-up.
- **Intuitive playbook builder:** No coding required! Create custom playbooks using an easy-to-use editor.

The playbooks you create use the information that people supply when they fill out your forms. So, the more details you ask for, the smoother the process runs.

Workspaces give fulfillers a place to address requests

Once deployed, your app will sit in its own workspace category, which makes it easy for fulfillers to answer requests. You can customize the workspace or use it as-is.

For sys admin eyes only: Every app built in Creator Studio adds a record in the Request App Config table, which contains the table that the app uses. Your admin can change the table for an

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

app, see Administering an app's associated table for more information. The name of the table follows the format of scope_request, for example, x_snc_02_03_request.

Find existing apps in Creator Studio

Perhaps you've been given permission to work on an app someone else created, or you want to add new items to the catalog in your app. The Creator Studio home page acts as your central hub, listing all the apps people in your company have built using Creator Studio.

Before you begin

Summary: In this topic, you'll learn how to:

- Find an app in Creator Studio.
- Open an existing app.
- Perform actions on an app without opening it.

Get permission

To work on an app, your system administrator has to assign you the role of sn_creatorstudio.user or sn_creatorstudio.restricted_user.

About this task

Techy detail: All the apps listed on the Creator Studio home page have an entry in the Request App Config table (sn_creatorstudio_request_app_config).

You can use bookmarks for apps too

In addition to the searching detailed in this topic, you can use bookmarks to save and quickly find the app you're looking for. Check out Work with bookmarked apps in Creator Studio for details.

Find the app you want to work on

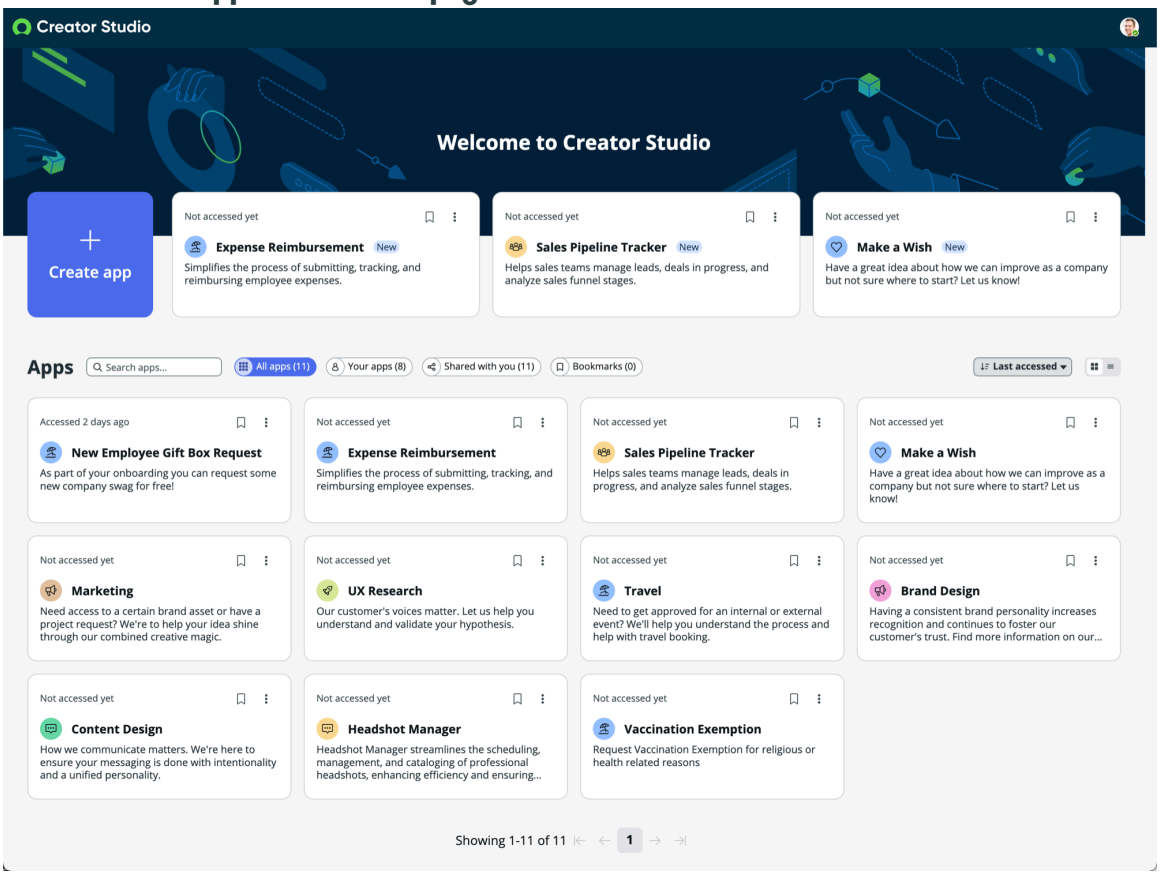
There could be a lot of apps listed on the Creator Studio home page. Here's how to find the one you want to work on.

Procedure

1. Go to **All** > **App Engine** > **Creator Studio**.
   The Creator Studio home page appears, which displays the apps most recently created by everyone in your company using Creator Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

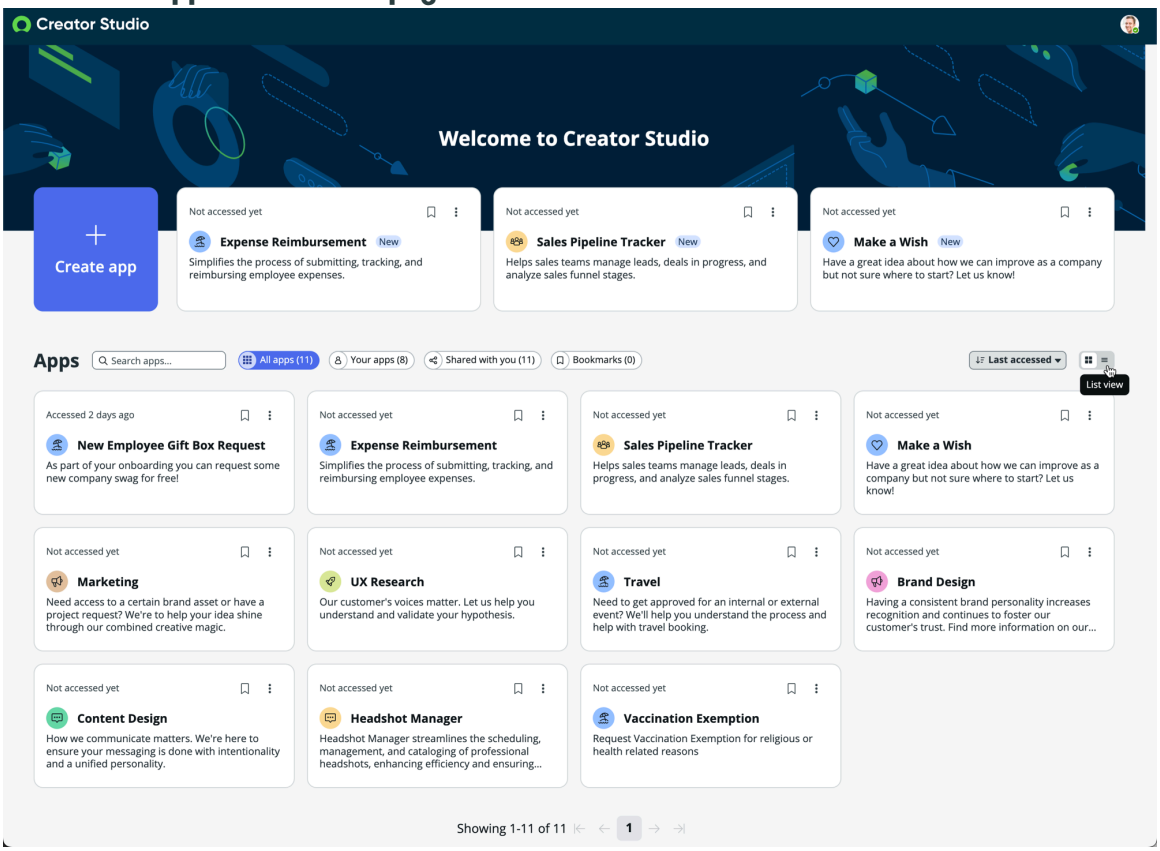
View available apps on the home page



2. **Optional:** Would you prefer to see the apps in a list instead of displayed as cards? Select the view icon and switch to **List View**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

List view of apps on the home page



3. If you know the name of the app you're looking for, start typing it the search bar.
   You must enter at least three characters for Creator Studio to filter the apps on your search criteria.

   Note: If you enter text in the search bar, all of the pills (such as **Your app** and **Bookmarks**) are also filtered based on the search term.

4. **Optional:** Narrow down the list of apps displayed by selecting one of the following:
   - **All apps**: Displays all the apps everyone in your company created using Creator Studio.
   - **Your apps**: Displays only the apps that you created. (You own the apps that you create.)
   - **Shared with you**: Displays only the apps you didn't create, but the owner said you could work on.

     For more information about working collaboratively on an app, see Delegated development and deployment.

5. **Optional:** Sort the list of displayed apps by selecting the drop-down sorting menu and then selecting one of the following options:

   - **Last accessed**: Lists the displayed apps based on the last time you worked on them; the most recently accessed app appears first in the list.
   - **Newest**: Lists the apps in the order of their creation date, newest to oldest.
   - **Oldest**: Lists the apps in the order of their creation date, oldest to newest.
   - **A-Z** or **Z-A**: Lists the displayed apps alphabetically in the order you choose.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: If you've entered text in the search bar, sorting respects the filtered search criteria.

6. **Optional:** 5. Select an app to open it in Creator Studio or select the more actions icon and select one of the following:
   - **Open in new tab**: Open the app in a different browser tab.
   - **Open in ServiceNow Studio**: Open the app in ServiceNow Studio to make more advanced edits.

     Note: You must have access to ServiceNow Studio for this option to appear. For more information on ServiceNow Studio, see Using ServiceNow Studio.

   - **Manage collaborators**: If you're the app owner, you can invite others to work on the app with you. For more information, see Manage collaborators for an app in Creator Studio.
   - **Copy link**: Send collaborators a link to the app so they can find it.
   - **Settings**: Update the app's image and other settings. For more information, see Edit an app's settings in Creator Studio.
7. **Optional:** Select **Request access** to open an app that you don't have permission to work on. To open an app, you must own it, or the owner must give you permission. For more information about requesting access, see Ask to work on an app in Creator Studio.

Result

Congrats: Congratulations! You've found and opened an app or performed an action on it. Next, we'll see how to work on it.

Work with bookmarked apps in Creator Studio

Bookmark an app to quickly find it in Creator Studio, as well as ServiceNow Studio.

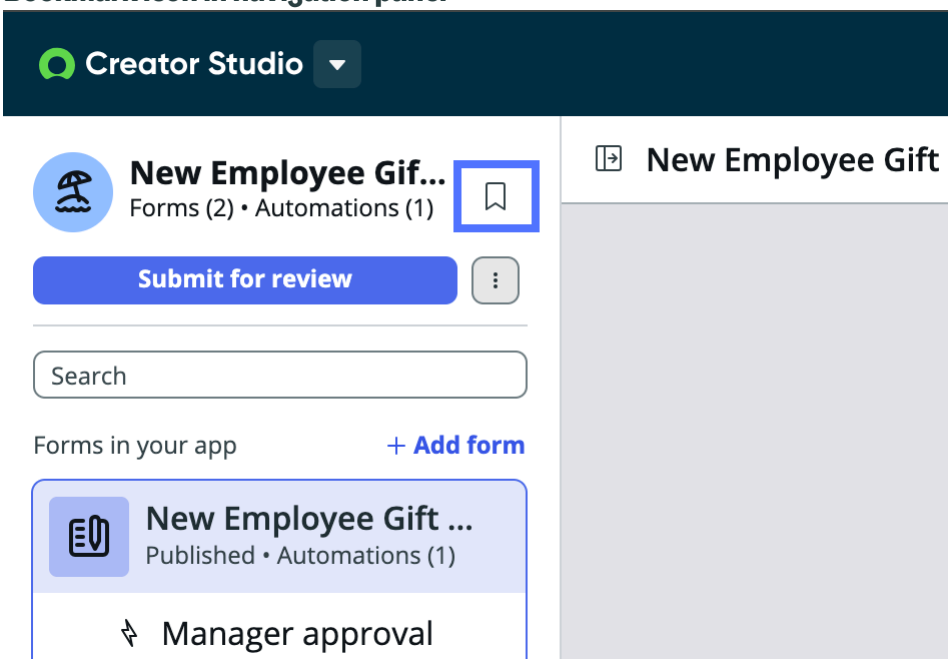
About this task

All apps that are made and bookmarked in Creator Studio are bookmarked and available to work on in ServiceNow Studio. However, apps that are made in ServiceNow Studio are not available to work on in Creator Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: If you already have the app open, you select the bookmark icon in the navigation panel.

Bookmark icon in navigation panel



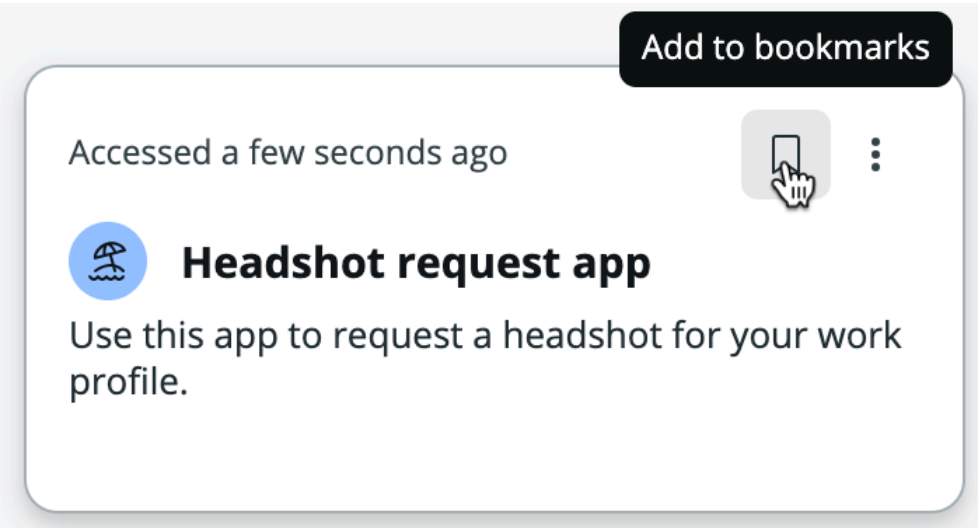
Before you begin

To work on an app, your system administrator has to assign you the role of sn_creatorstudio.user or sn_creatorstudio.restricted_user.

Procedure

1. Let's head over to **All** > **App Engine** > **Creator Studio**.
   All the apps built in Creator Studio appear on the home page.
2. Find the app that you want to bookmark.
   For details, check out Find existing apps in Creator Studio.
3. Select the bookmark icon for the app that you want to bookmark.

Select the bookmark icon

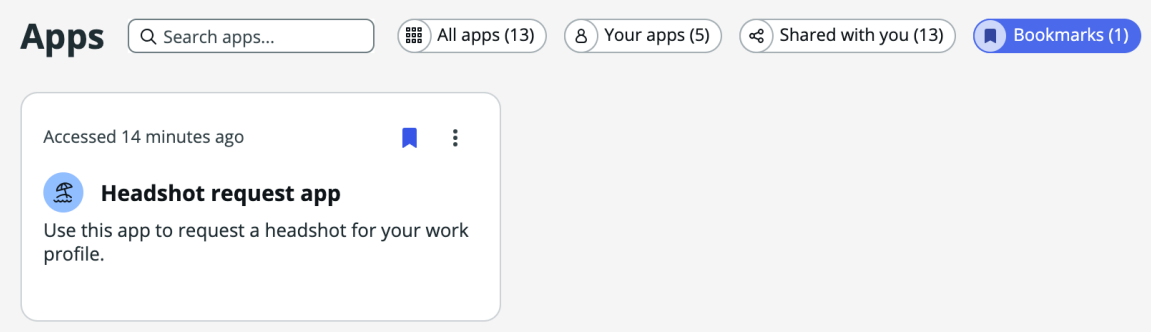


© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Result

Find your bookmarked app by selecting the **Bookmarks** pill on the Creator Studio home page.

Bookmarked apps on the home page



Note: If you've entered text in the search bar, your bookmarked app won't appear unless it matches the search criteria. To see all bookmarked apps, make sure you've cleared the search bar.

Change your development experience

Choose a different development experience if you want a more robust environment using the experience switcher in Creator Studio.

Before you begin

You must have permission to work in other development environments, such as ServiceNow Studio or ServiceNow IDE, to use the experience switcher.

Role required: admin, delegated developer, or any additional role that your admin granted access to

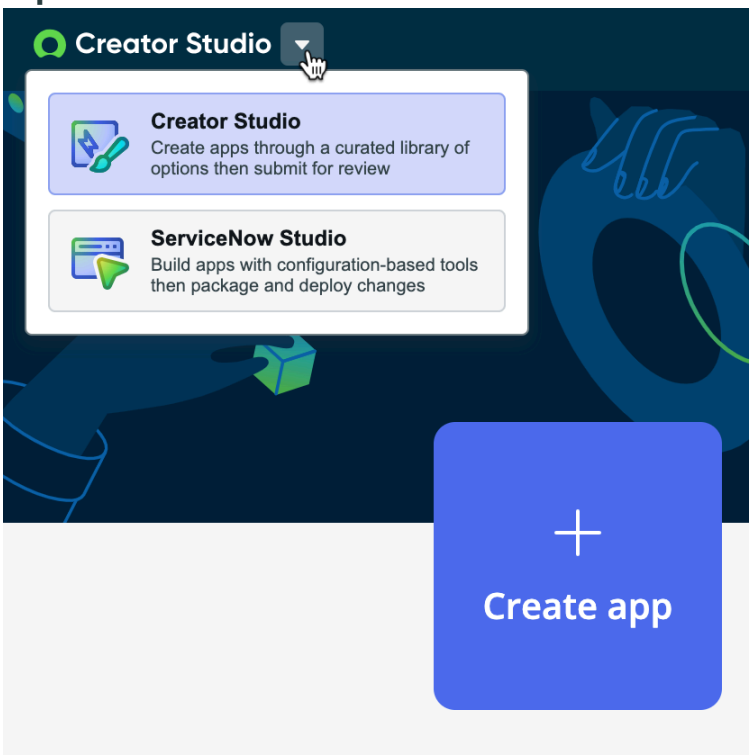
Note: Even if you have the delegated developer role, if you also have the Creator Studio User or Creator Studio Restricted User role, you won't see the experience switcher.

Procedure

1. Let's head over to **All** > **App Engine** > **Creator Studio**.
   All the apps built in Creator Studio appear on the home page.
2. Display the experience switcher by selecting the arrow icon next to the Creator Studio logo in the banner at the top of any page in Creator Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Experience switcher in Creator Studio



3. Select the development experience you want to switch to.
   For example, select ServiceNow Studio.

   Note: The development experiences that you can select from depend on your permissions. Contact your administrator if you need assistance.

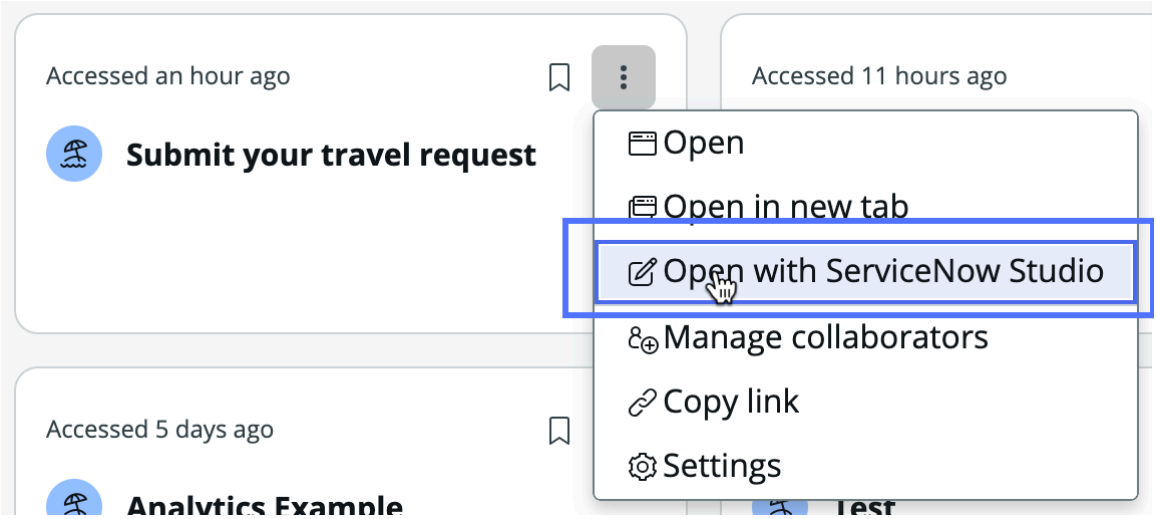
Result
The development experience you selected opens in the same browser tab.

Alternatively, you can open a specific app in ServiceNow Studio in the following ways:

- On the home page by selecting the more actions icon ... on the app's tile and choosing to **Open in ServiceNow Studio**.
- After you open the app by selecting the more actions icon ⋮ in the navigation panel and then select **Open in ServiceNow Studio**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Open in ServiceNow Studio link



App creation in Creator Studio

Have you ever dreamed of creating your own app? With Creator Studio, you might be closer than you think! The following links will send you to the sections in this guide that will walk you through the steps.

Do you have permission to create an app?

If you don't have permission to build apps yourself, just ask an admin. They can give you permission, or you can give them an app name and a short description, and they can start the app for you. Then, you can jump in later and edit it.

Don't build your app from scratch!

Creating an application from scratch can take some time. That's why Creator Studio does all the basic work for you. Just pick a form template (think starter kit) and provide some basic info (like a name) or use AI to create a form. Later, we'll show you how to customize your app.

Edit your app's look and feel

Want your app to stand out? By editing the settings, you can give it a unique name, a head-turning image, and other great features.

Create an app in Creator Studio

Get started creating an app by specifying basic info for the app that you're building in Creator Studio.

Before you begin

If you don't have permission to create an app, you can request that a teammate create one for you. For more information, see Ask an admin to create an app for you in Creator Studio.

You can also watch a short video on how to create an app.

https://player.vimeo.com/video/1123376394?h=27cf6583d7&badge=0&autopause=0&player_id=0&app_id=58479

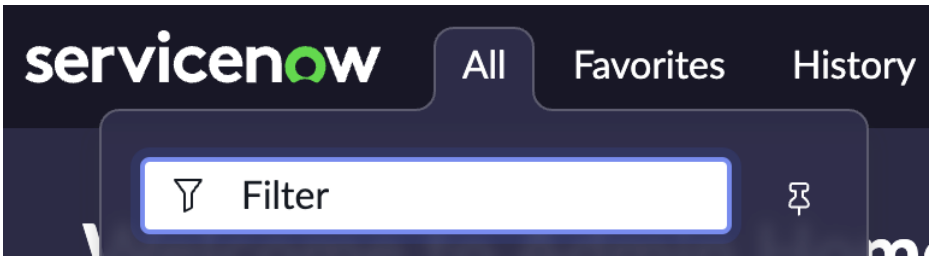
Your administrator must add you to the Creator Studio Users group.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Let's open Creator Studio. On your instance, select the **All** tab.

All tab

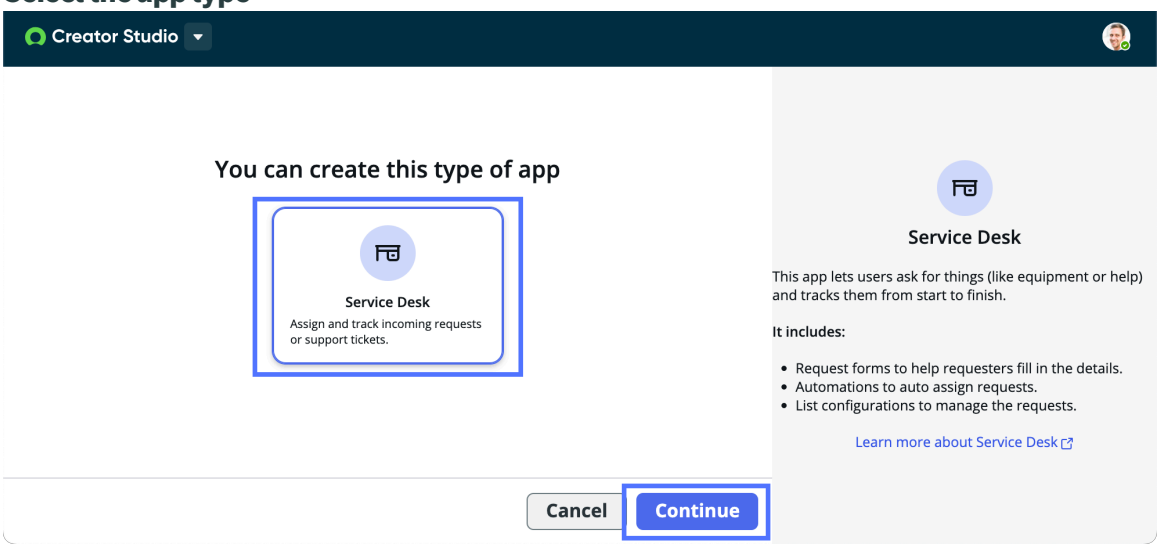


This tab lets you see all the apps installed on your instance.

2. In the white text field, enter `Creator Studio`.
The list of apps you see underneath the text box now pertains only to App Engine.
3. In the list, select **Creator Studio**.
That's how you open Creator Studio! You've landed on its home page. Under **Apps**, you see all the apps that people already created.
4. Select **Create app** to begin your journey.
   - If you're a system administrator, you can read more about this topic in Application collaboration.
   - If you want to know how to request an admin to create the app for you, check out Ask an admin to create an app for you in Creator Studio.
5. Select the type of app you want to build, such as **Service Desk**.
Service Desk apps enable users to submit requests, report issues, and access support related to services within your company.

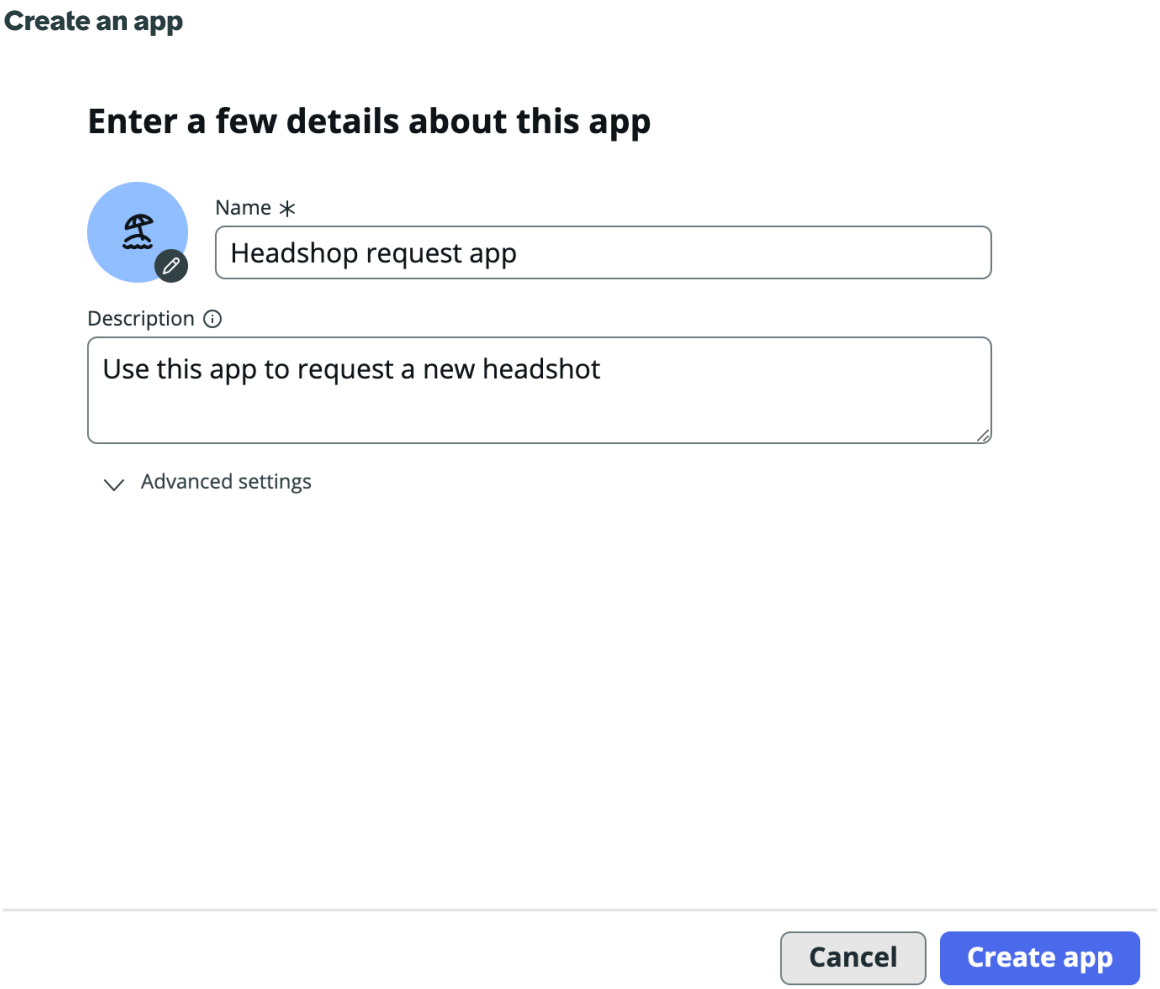
Note: Your admin may have hidden this page.

Select the app type



6. Select the **Continue** button.
7. Fill in the fields on the Create app page that appears.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



a. Give your app a **Name**, which should be descriptive and intuitive.

b. Describe what your app does in the **Description**.

For example, `This app enables a person to request office equipment.`

c. Optional: Select **Advanced settings** and confirm that your app's name and tables are unique by specifying the **Scope**.
For more advanced information about scopes, see Application scope.

d. Select **Create app**.

Result

Congrats: Hooray, you've created your app! Next, we will use a template to add a form that people will use to fill out requests in your app. Check out Add a form to an app in Creator Studio for how to do that.

To learn all the things you can do as the app owner when building the app, see Application collaboration.

For sys admin eyes only: Every app built in Creator Studio adds a record in the Request App Config table, which contains the table that the app uses. Your admin can change the table for an

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

app, see Administering an app's associated table for more information. The name of the table follows the format of scope_request, for example, x_snc_02_03_request.

Ask an admin to create an app for you in Creator Studio

Don't have permission to build an app? Ask your admin to get you started by creating a basic app for you. Just give them a name and tell them what you want the app to do. Once they start it, you'll take over and modify it.

Before you begin

To request an app to be created for you, the admin needs to assign you the role of sn_creatorstudio.restricted_user.

About this task

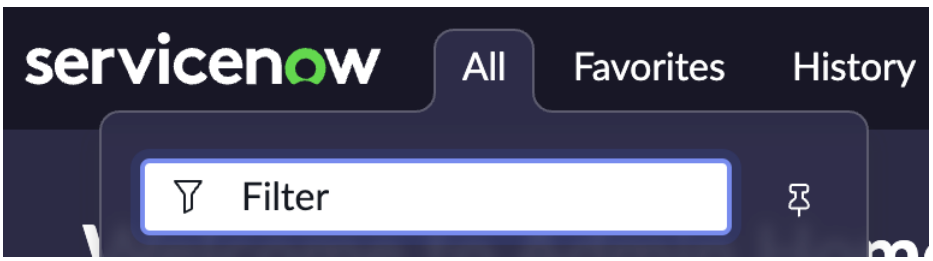
When you ask an admin to create an app for you, you automatically become the owner of it. For more information, see Collaborating with others to build apps in Creator Studio.

Here's how to ask an admin to create an app for you.

Procedure

1. Let's open Creator Studio. On your instance, select the **All** tab.

   All tab

   This tab lets you see all the apps installed on your instance.

2. In the white text field, enter `App Engine`.
   The list of apps displayed underneath the text box now pertains only to App Engine.
3. In the list, select **Creator Studio**.
   That's how you open Creator Studio! You've landed on its home page. Under **Apps**, you see all the apps that people already created.
4. Select **Create app** to get started with your request.
5. Fill in the fields on the modal that pops up.

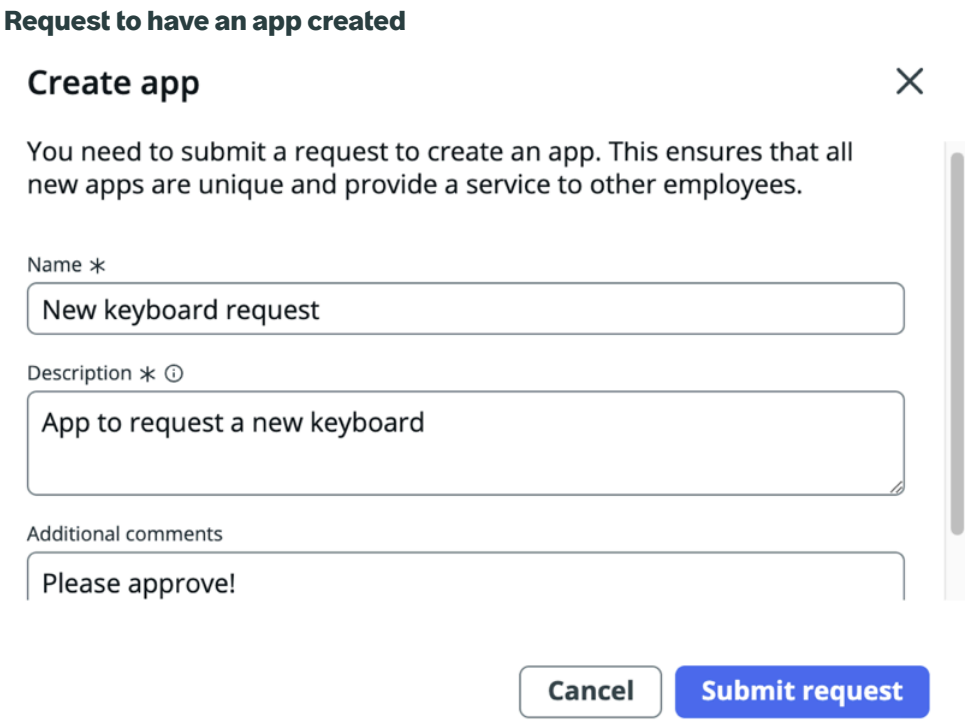
   a. Give your app a **Name**, which should be descriptive and intuitive.

   b. Describe what your app will do in the **Description**.

   For example, `App to request office equipment`

   c. Enter any additional comments that you want the admin to know when approving your request.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



6. Select the **Submit request** button.

Result

After you ask your admin to create the app, they'll review and approve the request. They might need more details about the app from you. You'll get an email notification once they approve and create it. At that point, you're the owner of the app. You can find your app in Creator Studio and start building out the app's forms, automation, and workspace category.

To learn all the things you can do as the app owner when building the app, see Application collaboration.

For admins:

- Every app built in Creator Studio adds a record in the Request App Config table. The table's name follows the format of scope_request, for example, x_snc_02_03_request. However, your admin can change the table for an app after you create it. For more details, check out Administering an app's associated table.
- You'll review and approve app requests in the Application Tasks table of the Request App Administration app.

Edit an app's settings in Creator Studio

Maybe you want to jazz up your app a bit, for example by giving it a new name, tweaking the description, or swapping in a new image. All it takes is just a few clicks in the settings! Let's dive into how you can do it, even if you're not a tech whiz.

Before you begin

To edit an app's settings, you must be given permission to work on the app.

About this task

There are some things you can't change after you've created the app, like its table or scope. If you want to change things up big time, you'll need to start from scratch with a new app.

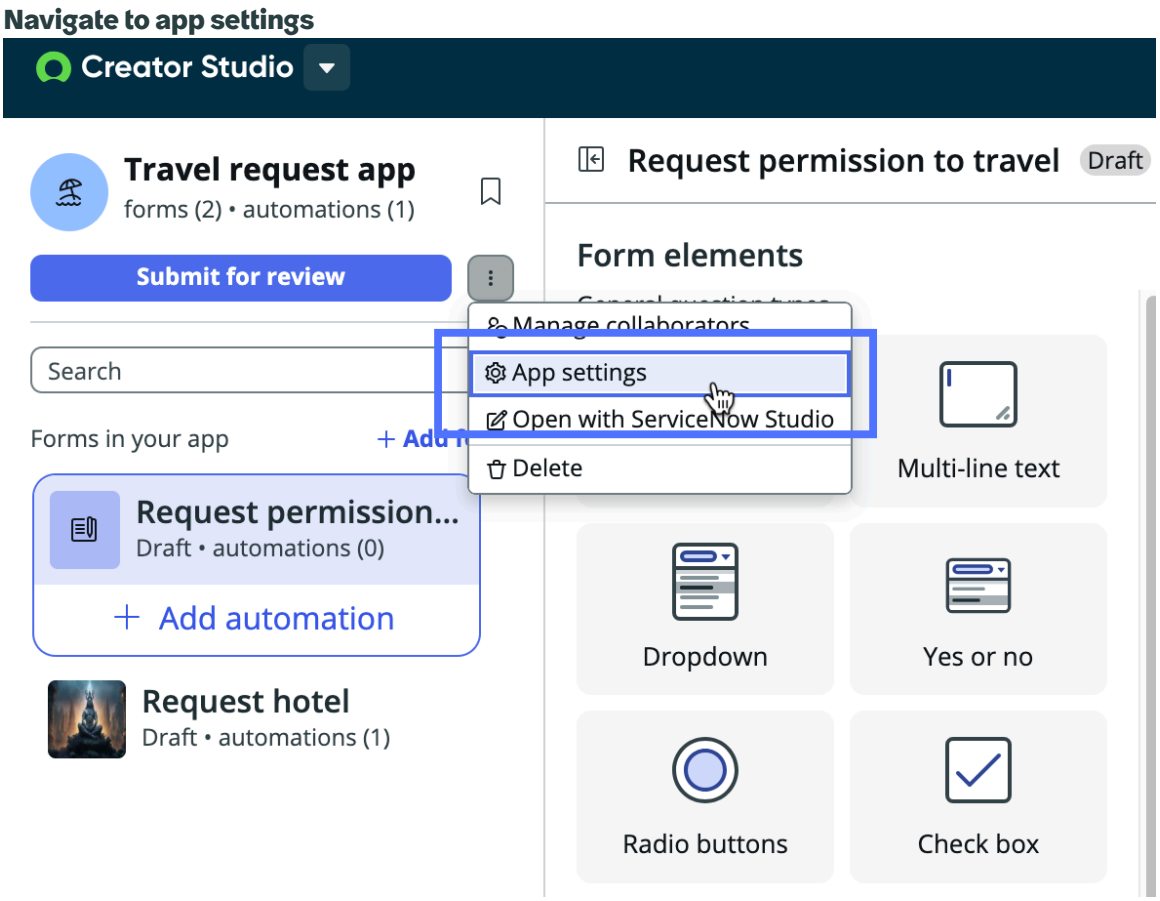
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Head over to the Creator Studio home page by going to **All** > **App Engine** > **Creator Studio**.
   You'll see a list of all your apps right there on the home page.
2. Select the app you want to work on.
   It'll open up so you can start making changes.
3. Open the App settings modal.

   a. Select the more actions icon ⋮ in the navigation panel.

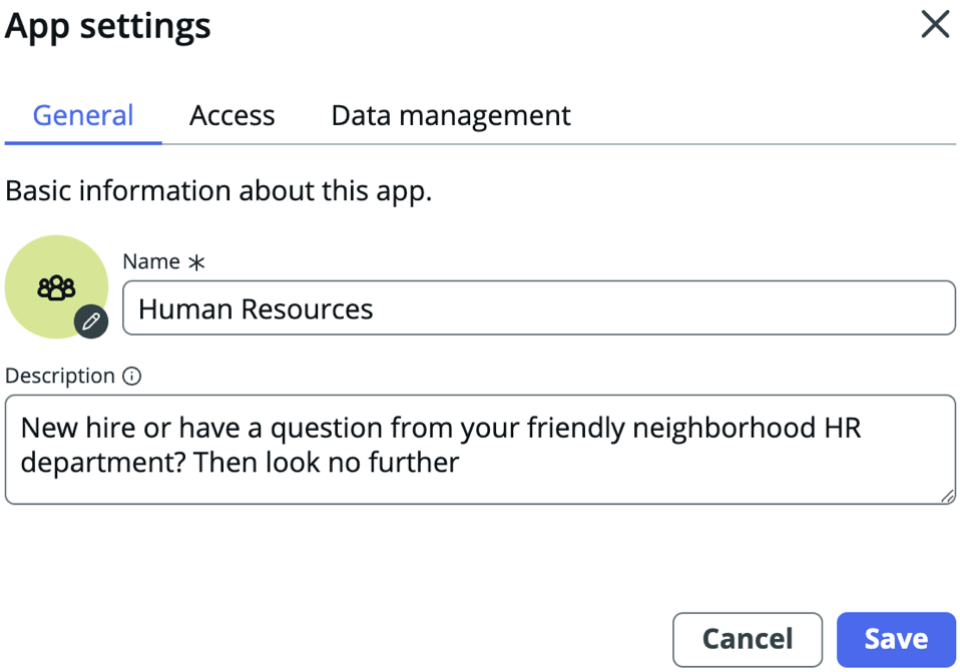
   b. Select **App settings**.



4. Now it's time to make those changes on the **General** tab, including updating the app's image.
   You can't make changes to security settings on the **Access** tab after the app has been created.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

App settings



5. Once you're happy with everything, select the **Save** button.

Result

And there you have it: With just a few simple steps, you gave your app a whole new vibe. So go ahead, get creative, and make it yours!

Delete an app in Creator Studio

Delete an app to remove it from Creator Studio.

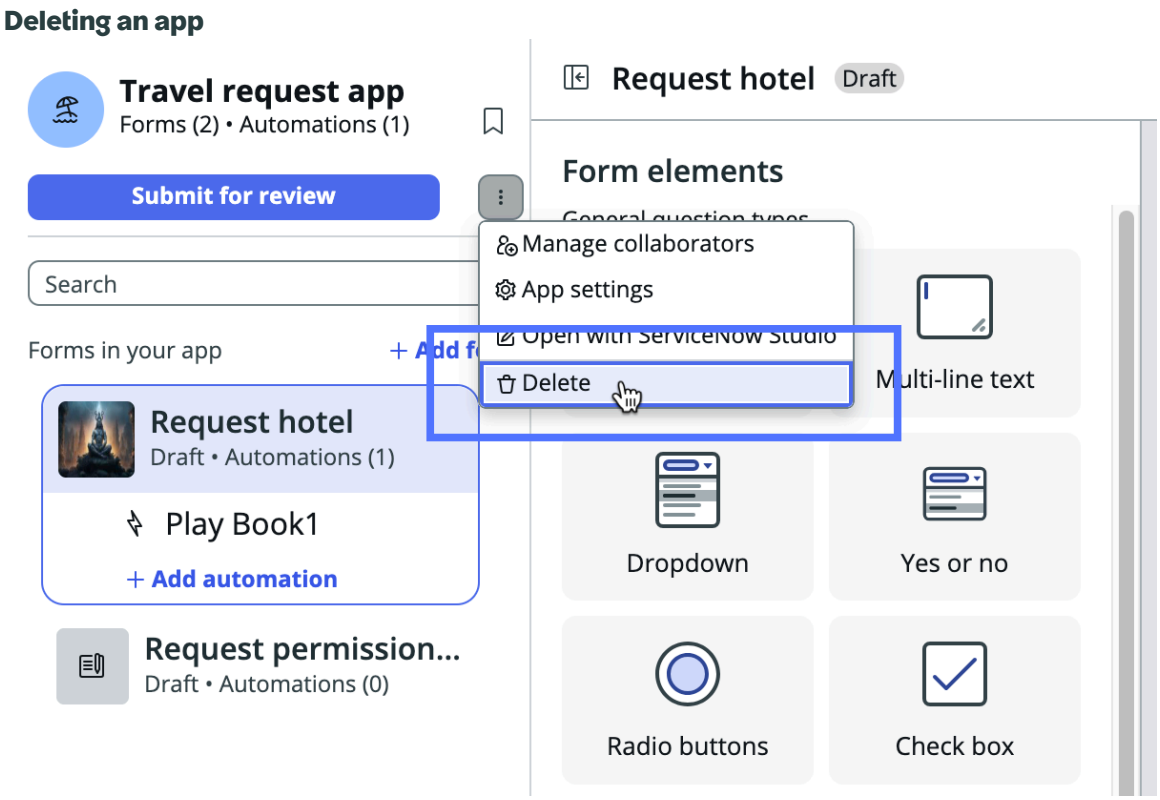
Before you begin

To delete an app, you must be given permission to delete apps. For more information, see Customized app collaboration permissions in Creator Studio.

Procedure

1. Go to **All** > **App Engine** > **Creator Studio** to see all the apps on the Creator Studio home page.
2. Open the app you want to delete.
3. Select the more actions icon ⋮ in the navigation panel.
4. Select **Delete** from the menu that appears.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



5. Type `Delete` in the modal that appears.
6. Select the **Delete** button.

Result
The app is removed from Creator Studio, but if it was deployed, the deployed app is still available.

Collaborating with others to build apps in Creator Studio

Sometimes you need help with building out your app, and that's OK! And sometimes other people need your help building their apps, which is great! This point is where collaboration comes into play.

You can collaborate on apps you don't own by requesting permission to edit them, and invite other people you work with to collaborate on your apps with you.

Collaboration, also referred to as delegated development, builds on the existing delegated development feature set in the ServiceNow AI Platform. It enables developers to invite other developers into apps so they can co-create and develop the app together. Depending on your permissions, you can invite others to collaborate on an app with you, or request to join someone else's app.

Note: You must have an App Engine Enterprise license to take full advantage of collaboration.

The ServiceNow AI Platform Collaboration app is automatically installed with Creator Studio. For more information about the Collaboration app, see Application collaboration.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Collaboration descriptors: Owners and Editors

So, you realize you either need help with your app, or you want to help someone else build their app. What's next? You need the appropriate collaboration role.

There are two standard types of collaborators when you co-develop an app with other people: Owners and Editors. These two roles are called collaboration descriptors. The collaboration descriptor that someone is assigned determines if they can assign, manage, and monitor delegated development permissions. For example, people who are Owners can do more than people with the Editor collaboration type.

If needed, your admin can define custom collaboration descriptors to select when managing collaborators, either in the Creator Studio Guided Setup or in the Collaboration app. For more information on custom descriptors, see Configure Creator Studio using Guided Setup and Create collaboration descriptors to assign permissions.

Default collaboration descriptors

Descriptor	Description
Owner	Owner of the application. • If you created the app, you're automatically the owner. • Owners can manage other collaborators for the app. • Owners have the delete app permission, which enables them to edit app settings, such as name, description, and icon. • Owners automatically get the delegated_developer role for the app.
Editor	• Editors can invite collaborators. • Editors have limited ability to edit the app.

Collaboration development permissions

You've designated Owners and Editors in your app, now what? You must invite other collaborators to work in your app.

Collaboration permissions enable you to control who's building apps in Creator Studio. You assign permissions to developers (or users who deploy applications) so that they can develop and deploy applications.

If you invite someone to collaborate on an app and they don't have the Delegated developer (delegated_developer) role, an App Engine admin must approve the collaboration request. For more information, see Delegated development and deployment.

When you add a user or group, a collaboration task is generated behind the scenes, and an approval flow kicks off. If you have App Engine Management Center (AEMC) installed, your admin can review and approve/deny these collaboration request tasks there. If you don't have AEMC installed, admins can navigate to **All** > *App Engine* > **Collaboration** > **Collaboration Tasks**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The collaboration task that goes to your admin provides information on which application a developer is being added to, and what permissions are granted. Approvers sometimes need to review these task records before they add developers to the application.

What can owners and editors do?

If you're in the Creator Studio users group, you can see all users and groups collaborating on an app, as well as their collaboration descriptors.

If you can't see the users and groups, you may have a different permission with more restrictions. But fear not! You can always contact your admin if you have any questions about who's working on an app.

The following table provides a list of general defaults for what owners and editors can do in Creator Studio.

Note: For the full list of default Owner and Editor collaborator collaboration type permissions, see Customized app collaboration permissions in Creator Studio.

Collaboration on Creator Studio features

Creator Studio feature	Owner	Editor
Invite collaborators	Yes	Yes
Manage collaborators	Yes	No
Edit app settings, such as name, description, and icon	Yes	No
Create, edit, and delete forms	Yes	Yes
Create, edit, and delete automation	Yes	Yes
Manage the fulfiller workspace list configurations and records	Yes	Yes
Submit app for deployment	Yes	No
Delete app	Yes	No

Collaboration roles and instances on different versions

As admins implement Creator Studio, they may have it installed on a non-production instance while their production instance is on a previous version of the ServiceNow AI Platform that doesn't have Creator Studio. This mis-match of instance versions affects the Collaboration Approval Workflow, which specifies the non-production instance as the source and the production instance as the controller. If the controller doesn't have the version of the collaboration plugin that supports Creator Studio, collaboration is unsupported.

To ensure that users can use the Collaboration Approval Workflow regardless of instance versions, admins must assign the catalog _builder_editor role to Creator Studio user groups.

Ask to work on an app in Creator Studio

Want to work on an app but don't have access to it? No problem. We'll show you how to ask for access.

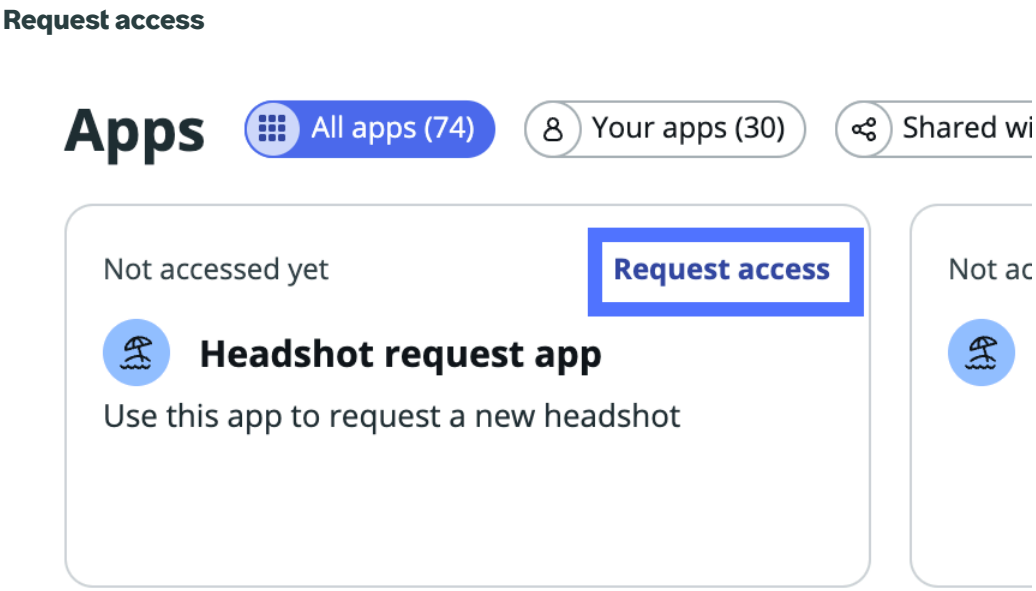
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin
First things first, you need the right access to get involved. Your system administrator must give you the role of sn_creatorstudio.user or sn_creatorstudio.restricted_user.

About this task
Getting permission to work on an app delegates you to its development, which means you can start pitching in and making things happen. For more information, see Delegated development and deployment.

Note: Admins don't see the **Request access** option because their permissions enable them to access all apps.

Procedure
1. Head over to Creator Studio by going to **All** > **App Engine** > **Creator Studio**.
   You'll see a list of all the apps built in Creator Studio right there on the home page.
2. Look for the app you want to help with and select its **Request access** link.



3. **Optional:** Use the **Enter an optional message** field if you want to leave a little note explaining why you're interested in joining the app. Maybe you've got some great ideas, or just want to learn the process.
4. Once you're ready, select the **Request access** button.

Result
With any luck, the App Engine admin will approve your request, and you'll be in! Then, it's time to roll up your sleeves and start working on the app.

Note: Users with a delegated developer role are automatically approved for collaboration.

What to do next

Important: After your request to join an app gets approved, you must sign out of the ServiceNow AI Platform and then sign back in to access the app.

If your organization uses App Engine Management Center (AEMC), App Engine admins approve collaboration requests there.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

If your organization doesn't have AEMC, admins approve requests to join apps in the Collaboration Tasks table. For more information, see Approve a collaboration request.

Manage collaborators for an app in Creator Studio

Add or remove people who can work on an app, or change their permissions for the app using the **Manage collaborators** option.

Before you begin

Your system administrator needs to assign you the Manage collaborators permission. Admins can also manage collaborators.

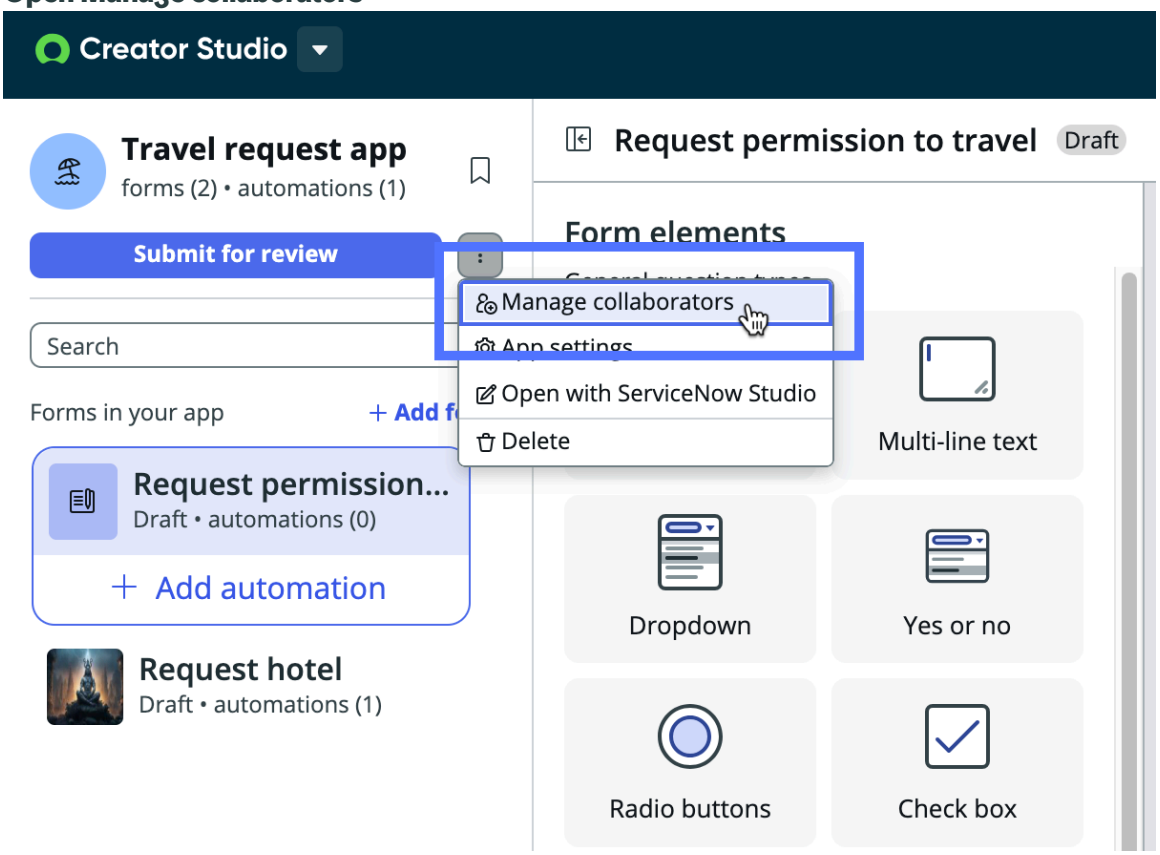
About this task

You can invite collaborators or manage collaborators depending on your permissions for each app. For example, if you have the Invite collaborators permission, you can't change the collaborator type of existing collaborators, such as Editor or Owner.

Procedure

1. Go to **All** > **App Engine** > **Creator Studio** to see all the apps on the Creator Studio home page.
2. Find the app you want to manage collaborators for.
3. Access the Collaborators with others modal in one of two ways:
   - From the Creator Studio home page, select the app's more actions icon and select **Manage collaborators** from the menu that appears.
   - Open the app, select the more actions icon in the navigation panel, and select **Manage collaborators** from the menu that appears.

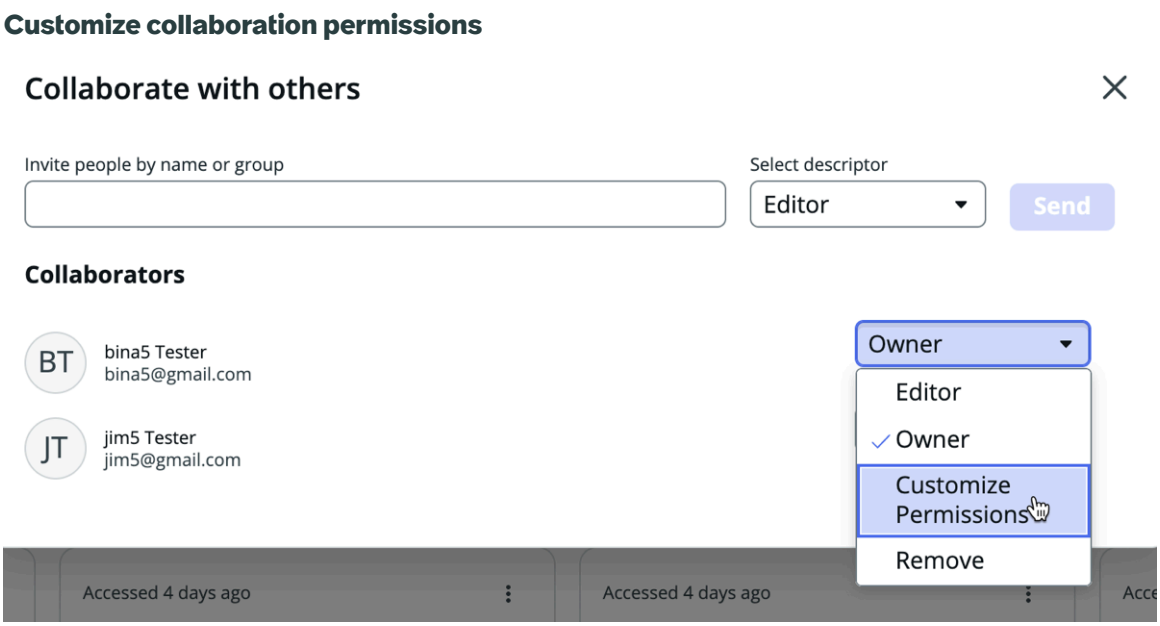
Open Manage collaborators



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

4. Change the permission of existing collaborators in the Collaborator section using the **Select descriptor** field.
   - **Owner** changes the user or group to be an owner for the app.
   - **Editor** changes the user or group to be an editor for the app.
   - **Customize permissions** enables you to select specific things that users can do with the app. For more information, see Customized app collaboration permissions in Creator Studio.
   - **Remove** revokes the user or group's access to work on the app.

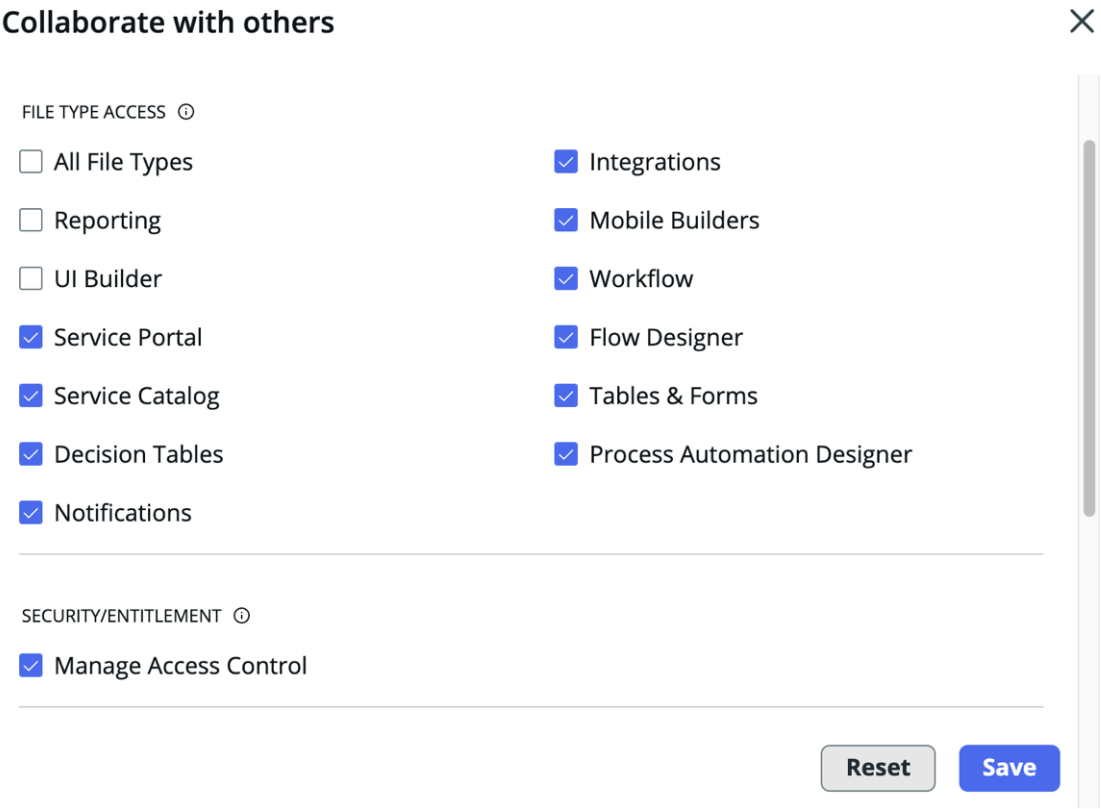
   Note: The collaboration descriptor must be in the global scope for it to be used by others.



5. Customize collaboration permissions by selecting **Customize permissions** and selecting the permissions the user should have.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Customize collaboration permissions



6. **Optional:** Add any additional collaborators.
   a. Start typing the people or groups you want to invite in the **Invite people by name or group** field.
   b. Specify the collaboration level in the **Select descriptor** field.
   c. Select the **Send** button to start the approval process.
   You can specify as many users and groups as needed.

Result
Unless you customized permissions, which requires the **Save** button, your changes are automatically saved when you close the Collaborate with others modal.

What to do next
Your App Engine admin must then approve the changes to collaborators.

Working with forms in Creator Studio

To add items or services to your catalog, you must create a different form for each thing being requested.

Summary: After reading this topic, you'll understand:

- How form templates get you started creating your own forms.
- How to customize the layout and questions on your forms.
- How to make a form (AKA catalog item) appear in your app.
- How to make a form stop appearing in your app.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

What is a form?

A form is a list of questions that people answer to make a request. They might ask for a keyboard, a laptop, or permission to take a vacation. The answers to those questions provide the fulfiller with all the information that they need to fulfill the request. For example, if someone asks for a laptop, the fulfiller might need to know the size, the model number, the operating system, and so forth.

In addition to questions, forms can contain images, headings, and text descriptions to explain the questions.

You'll need a different form for every type of item or request that a requester can ask for. Why? Because the questions you must ask for a keyboard differ from the questions that you must ask to give a requester permission to take a vacation. So, there's one form for each type of request that people can make. You can think of a form as representing a catalog item.

Your app will likely have multiple forms, meaning multiple things the requester can ask for. We call a collection of forms a catalog of items.

Key terms:

Item

Something a requester can ask for. Each item requires its own form.

Catalog

A collection of items that represent all the items a requester can ask for.

Some examples of catalogs and items

The following table shows three catalogs. The final column shows the names of the items in each catalog. If you have three items, you'll need three forms.

Example catalogs and their items

Service catalog	Example app	Sample catalog items
HR self-service	Human Resources global people portal app	• Benefits coverage question • Time off request • Change of address needed
Branding resources	Marketing and branding app	• PowerPoint templates • Approved Zoom backgrounds • Logo files
IT fulfillment	Hardware request app	• Mouse device • Headset • Replacement charging cable

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Let Creator Studio give you a head start

When you choose a template to start creating your form, Creator Studio builds a form with questions you might like to use. The questions and their layout make up the form.

Key term:

Form template

A set of questions arranged on a form provided by Creator Studio.

The form template that Creator Studio provides is called the Creator Studio Default Template. It contains three questions to get you started. You can customize, add, and subtract questions on your form.

Note: You can create a form using AI. To learn more, see Generating forms with AI.

Your administrator can delete the default template, create additional form templates, and restrict who can fill out the forms. If you're an administrator, you can read more about this topic in Creating catalog templates for use in Creator Studio apps.

To store the forms that requesters fill out, Creator Studio creates a table by default. One row represents one filled-out form.

Note: Your application can contain as many forms as you need.

When you create a form, Creator Studio uses the app's request table to hold the submitted answers for each form. All requests from all of an app's forms are stored in the same table.

Types of questions you can ask

You get to pick the questions to ask requesters. Creator Studio provides two types of questions you can ask:

- **General questions** are where you give requesters an empty box to write a short answer (think name or email) or a list of choices to pick from (like a dropdown menu or those little circles you fill in).
- **Reference-based questions** are where you provide a list of choices that you define in a table.

You can add questions to forms that have been preconfigured by your admin, and can't be edited. Adding a **Question set** element to a form enables you to select from the curated options.

You can make forms change based on how users answer questions in several ways:

- Forms can automatically update how questions appear (or are hidden) based on how users answer a them. We call this dynamic behavior. For example, if a user says they want a T-shirt for an event they're attending, you can make a **T-shirt size** field required.
- Forms can automatically populate answers based on answers to record choice questions. For example, a form can enable someone to select a user for the **Choose the gift box recipient** field, which retrieves its values from the User table. An additional **Preferred name** field can be automatically populated from the **First name** field of the User table selection, but then edited if someone prefers a nickname.

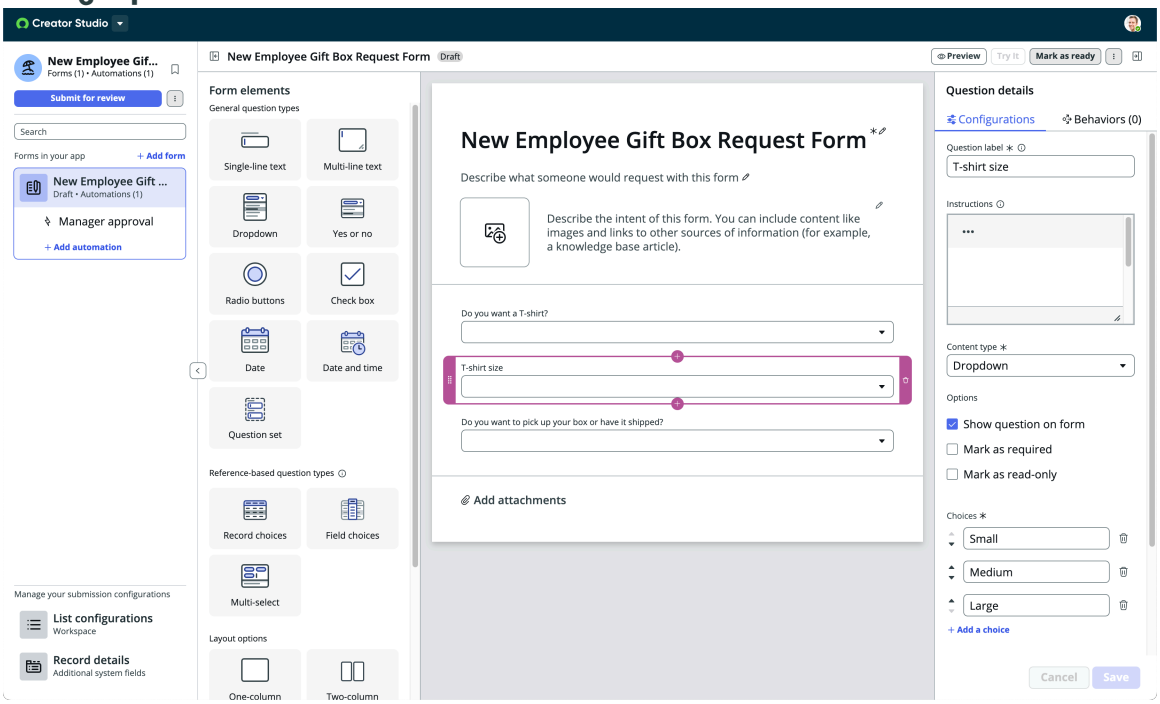
Laying out the questions

Creator Studio provides an initial layout for its default form. You can use columns, images, divider lines, and more to customize the layout of the questions on the forms. There are two procedures that you can use to customize a form:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

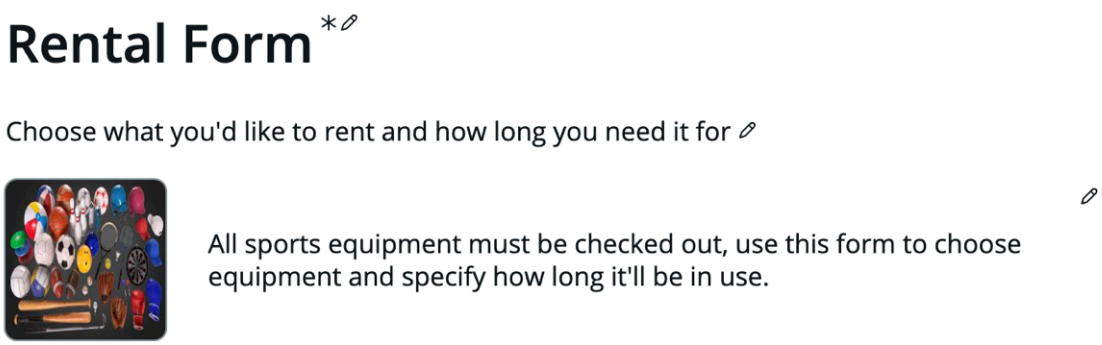
- Changing the questions – Change the number of questions, what they're asking, and adjust where questions appear on a form. Find information on changing questions in Customize your form for an app in Creator Studio.

Editing a question on a form



- Changing the look and feel of the form – Add pictures, write some text to explain the questions, and even move them around to make them flow smoothly. You can read about that in Change the layout of an app's record in Creator Studio.

Edit the look and feel of the app's tile



Where can I see forms?

After deployment, your app lives as forms in the Service Catalog and categories you specified when creating the forms.

Users can access those forms directly in a Service Catalog, as well as Service Portal and an Employee Center.

If you associate the app's form with one or more topics, the form will appear in the relevant, dynamically created topic pages in Employee Center. Find out more about topics in Associate a catalog item with a taxonomy topic in Employee Center, and more about taxonomy, which is a categorization method, in Unified Taxonomy for Employee Center.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: If you want to check out what a form looks like and test submitting responses to it, you can try it out after it's been published. When you try a form, you submit answers, which trigger any associated automations. For more information, see Test one of your forms for an app.

Making forms available for an app

Forms have two states:

- Draft
- Published

Keeping forms in a Draft state lets you work on them until you're finished. When you're pleased with the form, publish it by selecting the **Mark as ready** button. Only then will it appear in your application (assuming your app has been deployed).

A published form appears as an item in your catalog of offerings. For example, your site might enable requesters to ask for computers, mice, monitors, keyboards, technical support, and so forth. Each item requires a different form, and appears as a different catalog item in your app.

Key terms: Don't confuse deploying your app with publishing a form.

- Publishing a form means that the form will appear as a catalog item only after your app is deployed. That's why the button label is **Mark as ready**.
- Deploying an app is what system admins do to make your app available for people to use.

Add a form to an app in Creator Studio

Add a form to the app you just created. Forms contain questions that people respond to when submitting a service desk request.

Before you begin

Role required: You must be an admin or your administrator must add you to the Creator Studio Users group.

Procedure

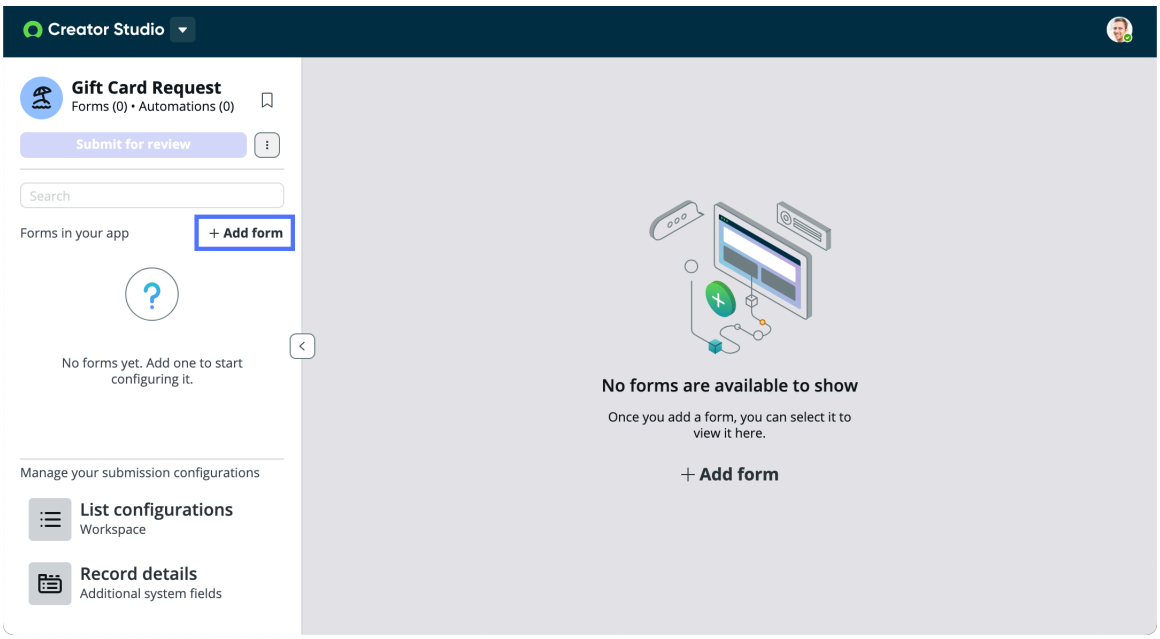
1. Let's head over to **All** > **App Engine** > **Creator Studio**.
   All the apps built in Creator Studio appear on the home page.

   Note: If you just created an app, this topic picks up right after you created an app, so you can skip to step 3.

2. Open the application that you want to add a form to.
3. Select **+ Add form** to get started.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Select to add a form



4. Select a catalog template from the list, which your admin can customize.

   You may not see a catalog template that is exactly what you want. So, choose one that's the closest. If none look close, select the **Creator Studio Default Template** option, if your admin hasn't removed it. It's your best bet.

   Tip: Feel free to click through the list of catalog templates. Previews will help you pick the one you want.

   Don't worry if you don't see other catalog templates to choose from! Your admins may not have created any custom catalog templates for you yet.

5. Select **Apply template and continue**.
6. Choose how you want to build the form:
   - To use AI, stay on the **Build with AI** tab and enter directions for the form you want. To learn more, see Generating forms with AI.

     Note: You can select **Try an example** to see some examples of the available directions. To see a quick mock up of what the form will look like, select the **Generate form preview** button. If you want a different form, change the provided direction and select **Regenerate preview** for a quick look.

   - To manually build the form, select the **Build on your own** tab and enter the information for the form:

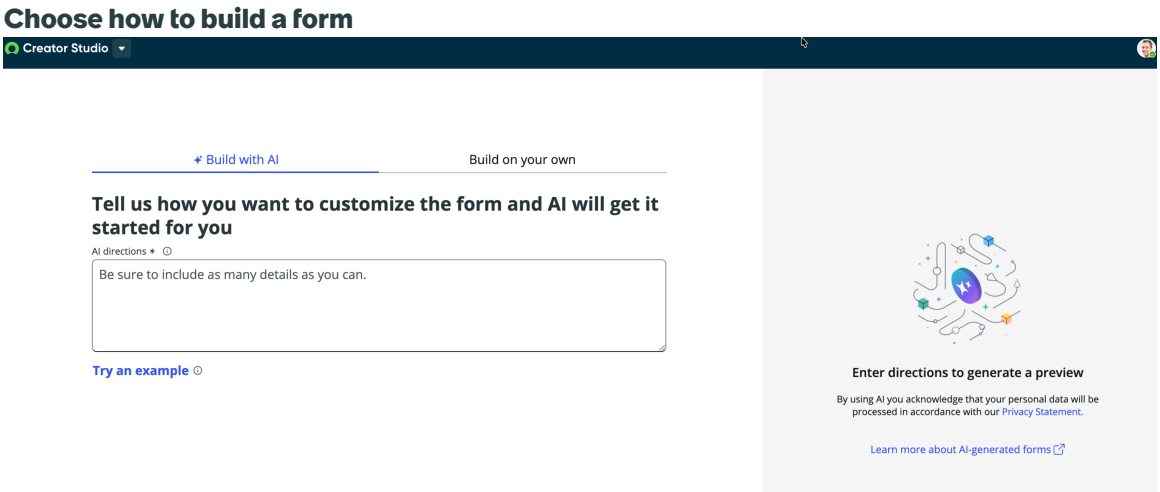
Form details

Field	Description
Form name	A descriptive name for the form.
Short description	Some information on what the form is used for.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Long description	Additional details for the form. You can include content like images and links to other sources of information (for example, a knowledge base article).

This page enables you to specify details for your first form, though you can also do that or edit the details, such as name and description, from inside the app after it's generated.



7. Select **Save and edit form** to get started editing your form.

What to do next

Now that you've added a form to your app, it's time to customize it! Remember: You can add multiple forms to your app for if you need to.

Customize your form for an app in Creator Studio

Forms help people ask for things they need, such as a new keyboard or permission to take time off. The default form that's added when you create an app in Creator Studio needs some changes to fit your needs. For example, you must add question labels to gather information about the request.

Before you begin

To customize a form, you must be given permission to work on the app.

About this task

You can add additional forms to your application if you need them. Additional forms are stored in the app's same table. No need to worry about those tables right now, but if you want to know how to add more forms besides the form that was generated when the app was created, check out Add more forms to an app in Creator Studio.

Summary: After finishing the following steps, you'll know how to:

- Open the form that was automatically generated when the app was created.
- Add questions and images to the form.
- Arrange the questions and images that you added.

Let's open a form and customize it.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

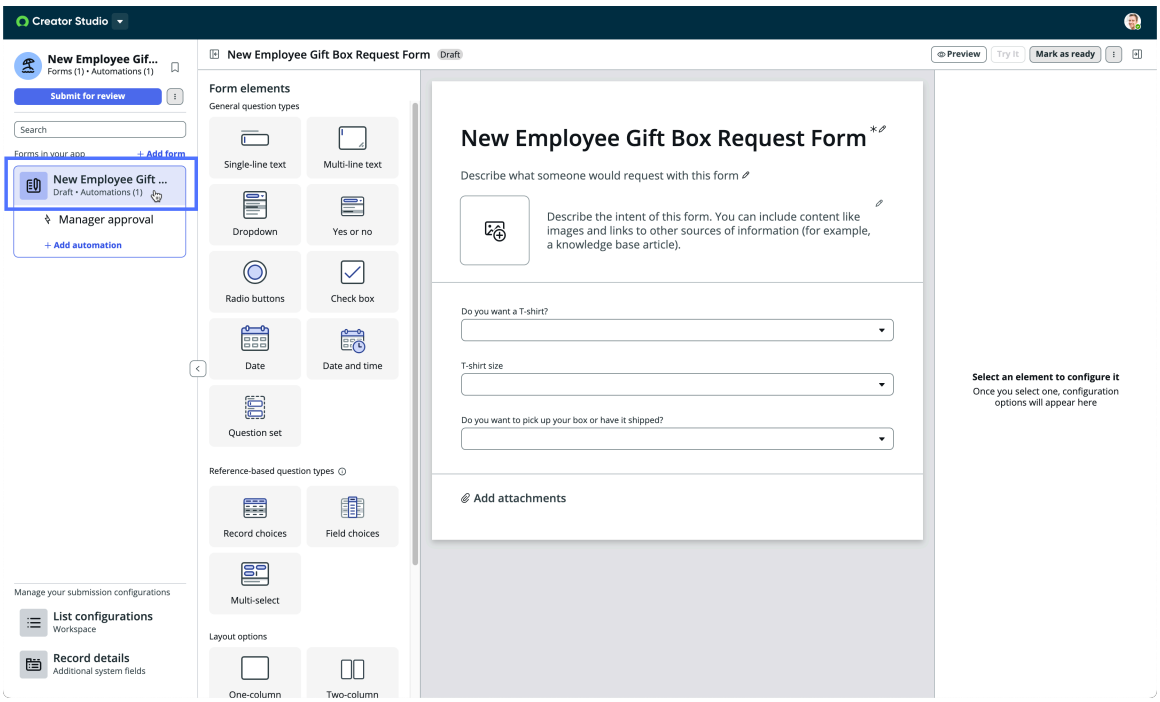
Note: For example, you can create a form that people use to register for a company event, with questions like dietary preference and accessibility needs.

Procedure

1. Let's head over to **All** > **App Engine** > **Creator Studio**.
   All the apps built in Creator Studio appear on the home page.
2. Open the application that contains the form you want to customize.
3. Select the form you want to customize in the navigation panel.

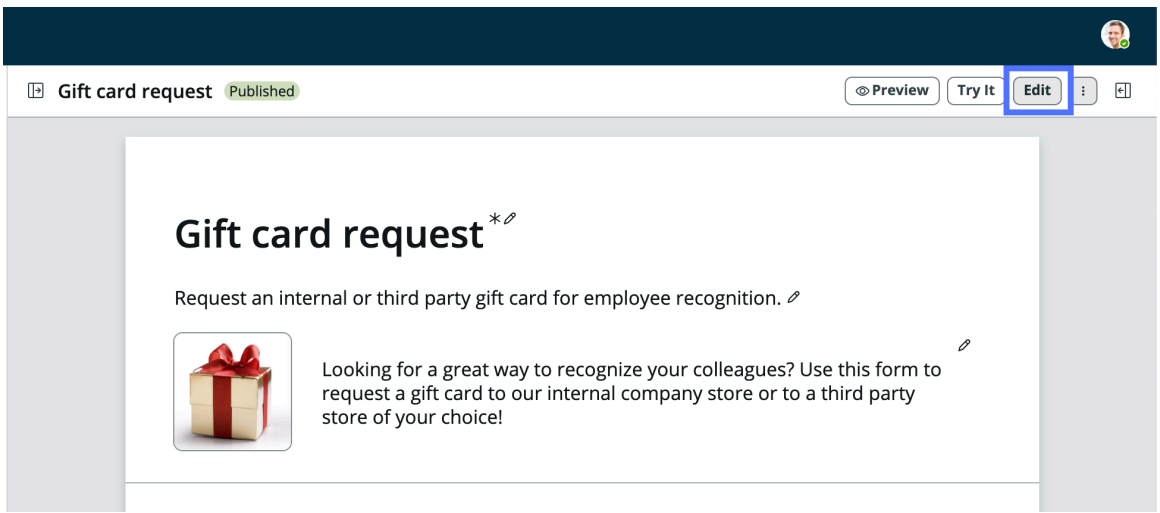
Note: If your app has a lot of forms, you can use the search box in the navigation panel to search for a specific form.

Select the form



4. **Optional:** Select the **Edit** button if you want to work on a form that's already been published.

Edit button



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

5. Let's customize the form by doing one or more of the following optional steps.

   a. Add or modify the image that appears on your form by selecting the add image icon ( ) and then selecting an image.

   b. Change the form's title, short description, and other text by selecting those parts of the form and typing in your changes.
   You can enhance how the longer **Description** appears using rich text, such as font changes and sizing.

6. Next, add questions to the form your fulfiller needs to evaluate the request. To add and customize a question, complete the following steps.

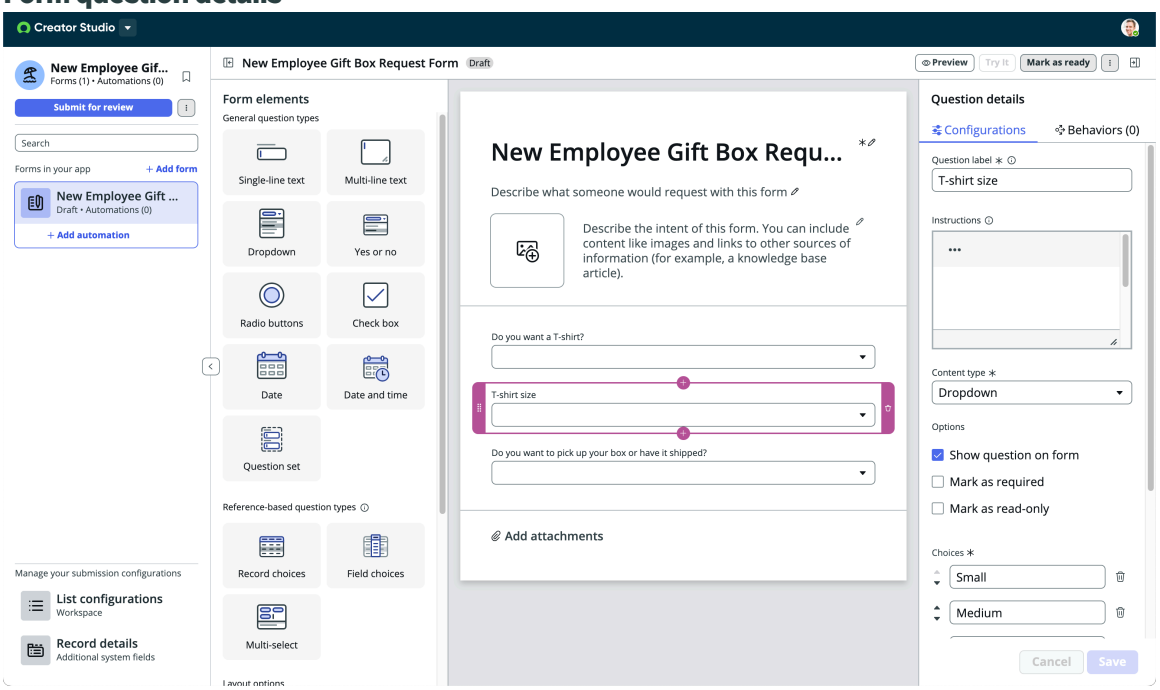
   a. Add another question by dragging the type of question you want from the Form elements panel onto the form, dropping it on the canvas where you'd like it.
   You can also add questions by selecting the add icon (+) that appears when you click on an existing question on the form. If you're adding a pre-configured **Question set**, you must select the question set from the modal that appears when you drag it onto the form.

   For a description of question types and how they're used, see Available question types in Creator Studio.

   b. Select the question.
   When you select a question, it's highlighted on the form so you know what you're working on.

   c. In the **Configurations** tab of the Question details panel, specify information about the question you added, such as a label for what the question is asking and whether it must be answered for the requester to submit the form.
   The details vary by question type. For example, if you add a **Dropdown** question, you must supply the options to choose from.

Form question details



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

d. **Optional:** Make the form's appearance change based on how users answer questions by adding dynamic behavior to it on the **Behaviors** tab.
For example, if a user says they want a T-shirt for an event they're attending, you can make a **T-shirt size** field required. Get the details on adding dynamic in Make a form change based on responses in Creator Studio.

e. **Optional:** Automatically populate a question's answer based on the answer to another question on the form (record choices questions only) by selecting the **Add auto-fill** option. For example, a form can enable someone to select a user for the **Choose the gift box recipient** field, which retrieves its values from the User table. An additional **Preferred name** field can be automatically populated from the **First name** field of the User table selection, but then edited if someone prefers a nickname.

The **Add auto-fill** option is available only if the form has a record choices question on it. You must also complete the following fields:

Details for auto-populating a question based on a record choices answer

Field	Description
Select the dependent question	Question whose answer makes the current question automatically populate.
Select the value field	Field in the source table for the record choices question that contains the value you want to use as the answer to the current question. (Optional) You must select the edit icon ( ), navigate to the field in the source table that contains the auto-populated answer, and select to **Apply** it. For more info on selecting fields in a table, see Dot-walking to data in related tables.

f. Select **Save and close** when you finish modifying the question.

g. Revise or add more questions to the form using this procedure. To change a question type, select the question and then select the new question type in the **Content type** field of the Question details panel. Selecting a new type may introduce new values you must supply. Keep these specifications in mind as you create your questions:

- You can't change an existing question into a question set. To include a question set on a form, you must newly add it to the form.
- If you put two checkbox questions side-by-side on a form, they make a section. You can't add other types of questions to that section.

7. Now, let's arrange the questions and images that you've added to the form.

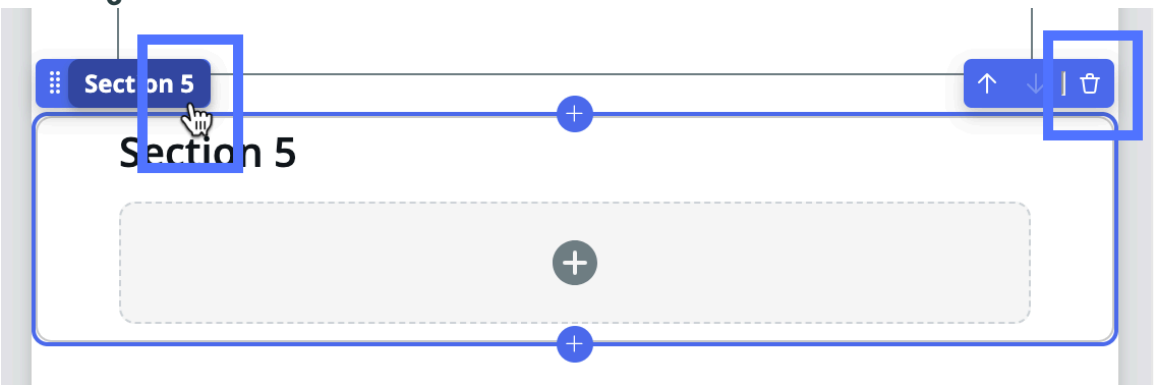
a. From the Form elements panel, drag the layout option you like onto the form and drop it where you want it to appear, for example, a **Divider line**.
Don't worry if you don't like the layout, just try another one by dragging it onto the form's canvas.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

b. Revise the form layout you chose using the Section details or Question details panel (depending on the layout you're working on). You can do things like make text bold, add links, and so forth.

Note: To edit or delete a section, you must hover over the section name and then select **Section** to see the section details in the properties panel, as well as the delete icon.

Selecting a section

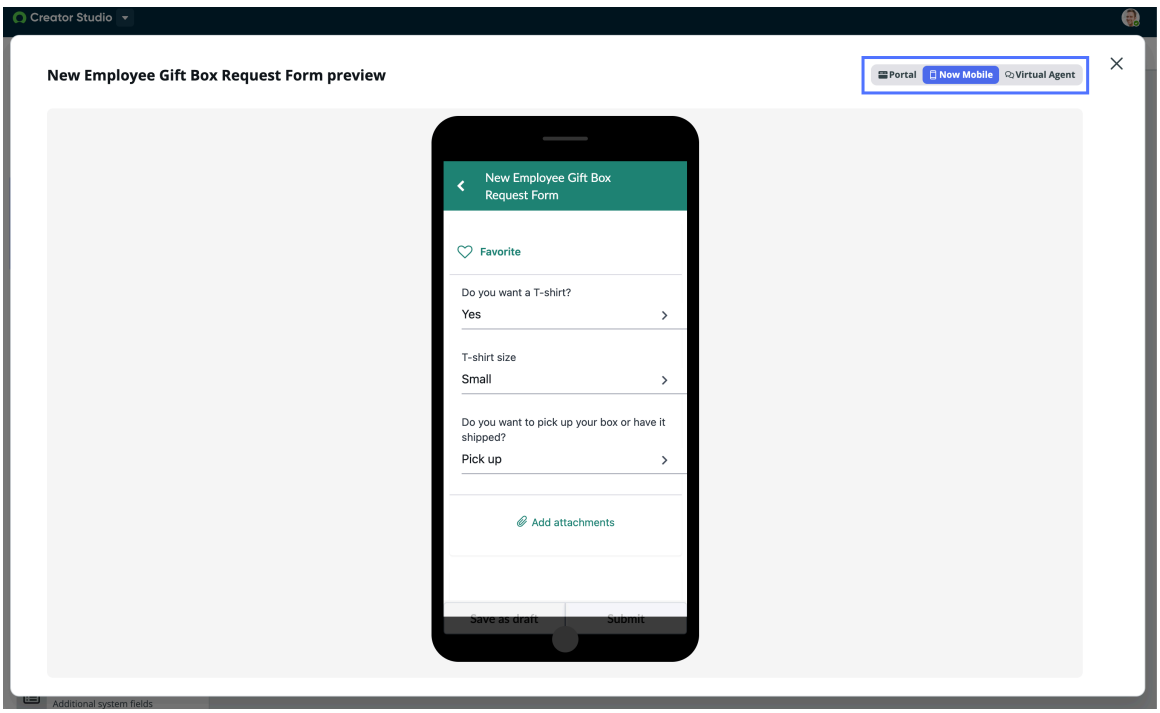


For more information, see Layout options for forms in Creator Studio.

c. Select **Save** in the Section details/Question details panel when you're done revising the form's layout.

8. **Optional:** See how your form will look in various experiences by selecting the **Preview** button. You can see a preview of the how the current version of the form will appear in various experiences by selecting **Portal** (such as Employee Portal), **Now Mobile**, or **Virtual Agent**. You can fill in the form when previewing, but selecting the **Submit** button doesn't generate a task record.

Preview a form



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

9. **Optional:** Test using the form and the record it generates by selecting the **Try it** button.

   (Optional) As you test the form, note any updates you'd like to make and then make those changes on the form before you publish it. For details, see Test one of your forms for an app.

Result

Congrats: Congratulations! You've customized the default form that came with your app.

What to do next
You must then mark the form as ready to publish it, which makes it available for the app to use. Find out how to publish a form in Publish a form for your app in Creator Studio.

You can select the **Undo all changes** option, available in the form header's more options icon (), to reset a form to the most recently published version.

Generating forms with AI

Learn how ServiceNow Otto for Creator helps you to create forms for you in Creator Studio.

AI in Creator Studio overview

While creating your app in Creator Studio, you can describe the form you want, and AI generates the form for you.

The application uses the ServiceNow Otto for Creator skills, such as Catalog item generation and App generation, to generate the forms.

Note: You must be assigned the now.assist.creator role to create the forms using AI.

Forms generated with AI appear with an AI icon in the application banner.

AI in Creator Studio benefits

The benefits of using AI to generate forms in Creator Studio include the following:

- Forms of your choice are created quickly with just a direction.
- If you provide more details about your form while writing the direction, AI can generate the form specifically to meet your needs.
- You can also use the pre-available examples in the directions available to generate forms.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ServiceNow Otto language availability notice

Important:

- Not all model providers are available for customers with in-country SKUs, and some AI products/features are currently unavailable for in-country customers. For more information, see the KB1584492 article in the Now Support Knowledge Base. Be sure to check for model provider availability updates in future releases.
- Some AI products/features are currently unavailable for customers in the FedRAMP, NSC DOD IL5, or Australia IRAP-Protected data centers, self-hosted customers, or in other restricted environments. For more information, see the KB0743854 article in the Now Support Knowledge Base. Be sure to check for availability updates in future releases.
- Some AI products/features are currently available only for customers in some regions. Be sure to check for availability updates in future releases.
- Some AI products and skills are not available in Regulated Markets. For more information, see KB2593939: Regulated Markets AI Products/Skills Not Available. Be sure to check for availability updates in future releases.

AI limitations

This application uses artificial intelligence (AI) and machine learning, which are rapidly evolving fields of study that generate predictions based on patterns in data. As a result, this application may not always produce accurate, complete, or appropriate information. Furthermore, there is no guarantee that this application has been fully trained or tested for your use case. To mitigate these issues, it is your responsibility to test and evaluate your use of this application for accuracy, harm, and appropriateness for your use case, employ human oversight of output, and refrain from relying solely on AI-generated outputs for decision-making purposes. This is especially important if you choose to deploy this application in areas with consequential impacts such as healthcare, finance, legal, employment, security, or infrastructure. You agree to abide by ServiceNow's AI Acceptable Use Policy, which may be updated by ServiceNow.

Data processing

This application requires data to be transferred from ServiceNow customers' individual instances to a centralized ServiceNow environment, which may be located in a different data center region from the one where your instance is, and potentially to a third-party cloud provider, such as Microsoft Azure. This data is handled per ServiceNow's internal policies and procedures, including our policies available through our CORE Compliance Portal.

Data collection

ServiceNow collects and uses the inputs, outputs, and edits to outputs of this application to develop and improve ServiceNow technologies including ServiceNow models and AI products. Customers can opt out of future data collection at any time, as described in the AI Opt-Out page.

Make a form change based on responses in Creator Studio

Make a form update based on how users answer a question using dynamic behavior. For example, if a user says they want a T-shirt for an event they're attending, you can make a **T-shirt size** field required.

Before you begin

You must name the form before you can add dynamic behavior to it.

To add dynamic behavior to a form, you must be given permission to work on the app.

About this task

A different way that you can make a form change is to have questions automatically populate based on a user's answer to another question on the form. For more information, see Customize your form for an app in Creator Studio.

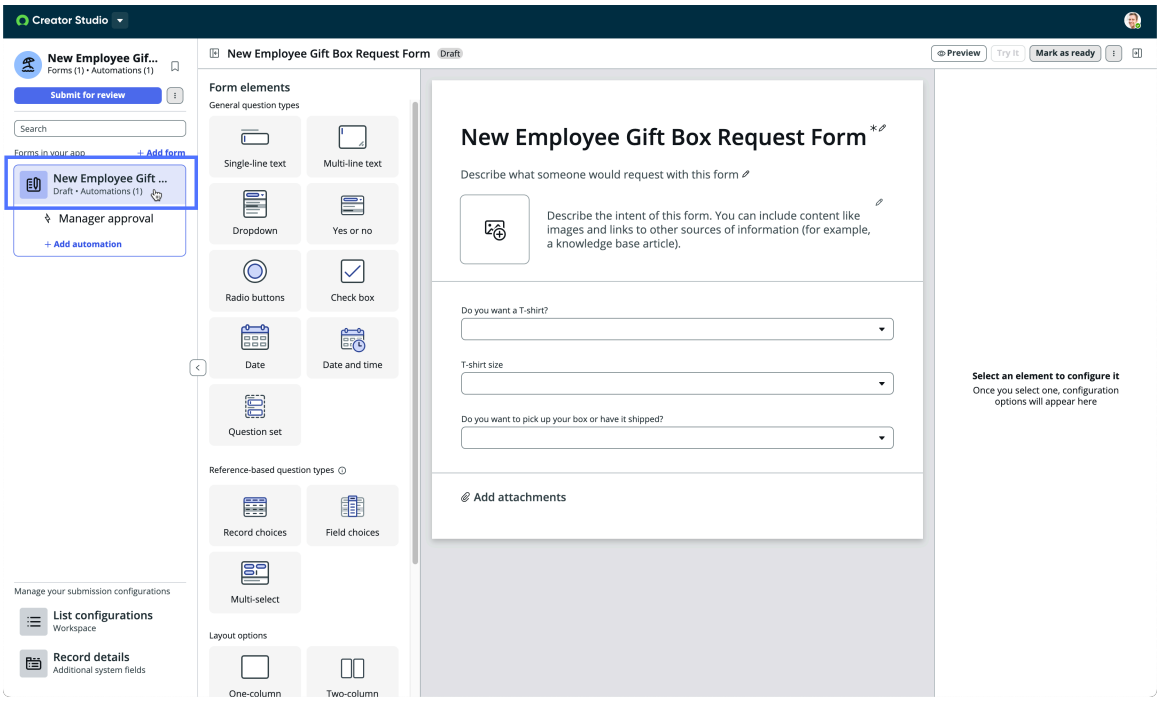
Procedure

1. Go to **All** > **App Engine** > **Creator Studio** to see all the apps on the Creator Studio home page.
2. Open the application that contains the form you want to add dynamic behavior to.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

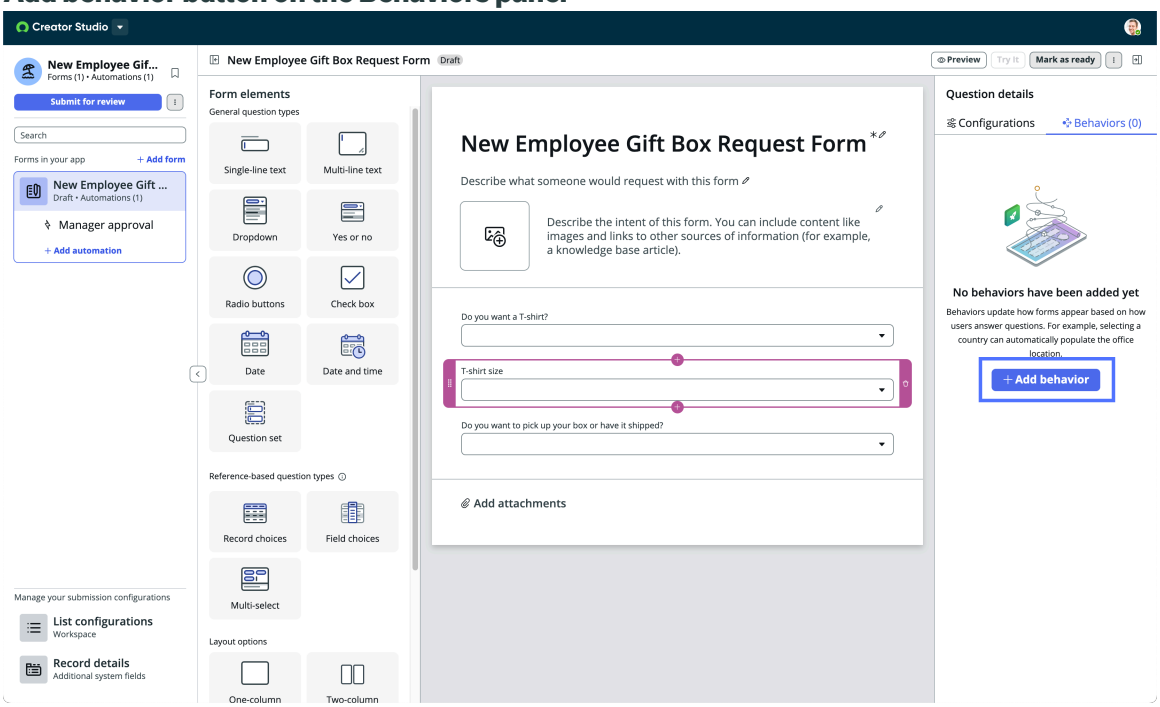
3. Select the form you want to work on in the navigation panel.

Select the form



4. Select the question that should be affected by how users answer one or more previous questions.
   For example, select the **T-shirt size** field to make it required if a user answers `Yes` to the **Do you want a T-shirt?** question.
5. Select the **Behaviors** tab of the settings panel.

Add behavior button on the Behaviors panel



6. Define the conditions that will make the form change in the Conditions section.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. Select **Questions** in the **Field** field to specify that you're choosing questions from the form. The name of the form automatically appears in the **Item** field.

b. Choose the question whose answer triggers the dynamic behavior in the **Question** field. For example, select the **Do you want a T-shirt?** question.

c. Select the condition **Operator** to determine the status of the selected field that will trigger the dynamic behavior.
For example, you could select **Is** as the operator for the **Do you want a T-shirt?** field.

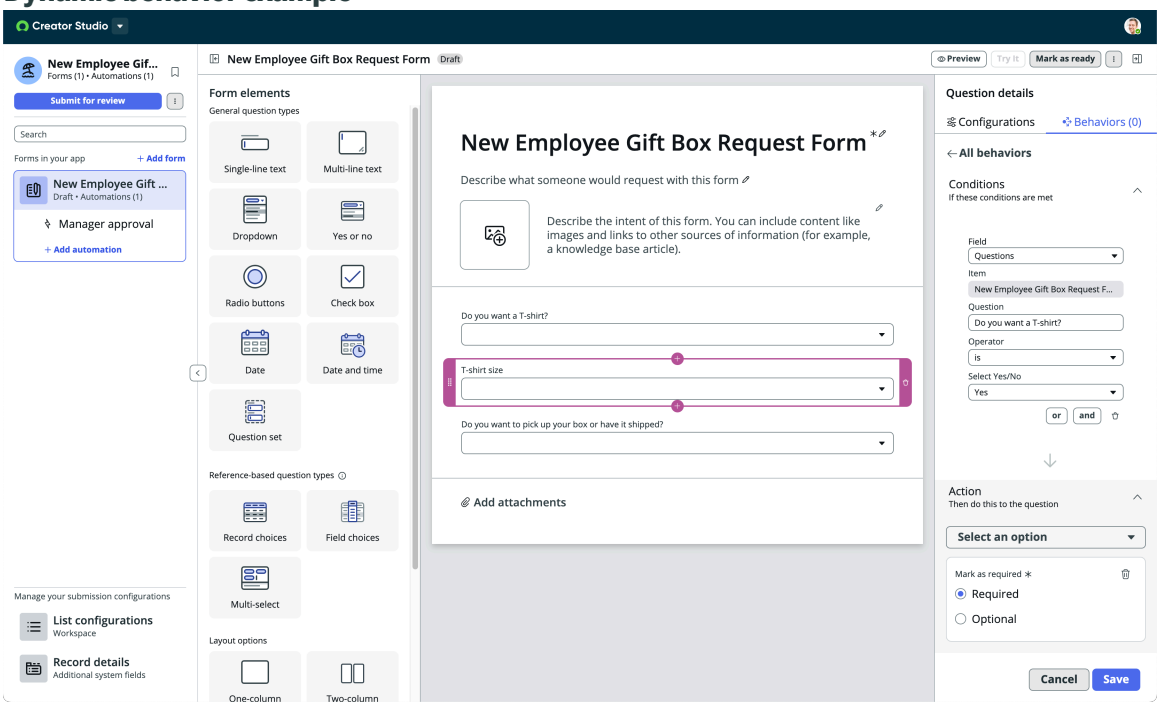
For more information, see Condition builder.

d. Enter or select the value for the user's response to the trigger question.
For example, the value could be **Yes** if they want a T-shirt.

e. Optional: Specify another field and value to create a more complex set of triggering conditions by selecting the **or** button or the **and** button.

f. Optional: Build another set of conditions by selecting the **Add condition set** button and repeating steps a-e.

Dynamic behavior example



7. Define how the form changes in response to the conditions that you defined in the Action section of the **Behaviors** panel.

a. Choose what the form does when the triggering conditions are met in the **Select an option** field.
Depending on the conditions you created, the following actions may be available:

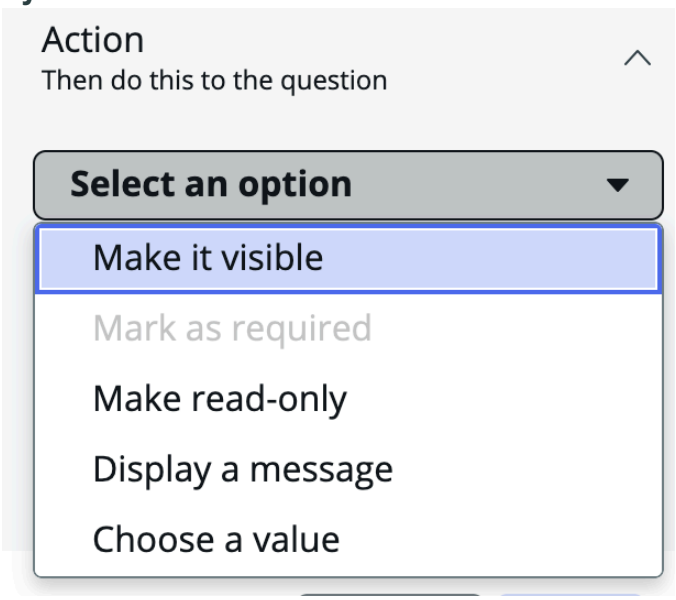
- **Make it visible**
- **Mark as required**

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- **Make read-only**
- **Display a message**
- **Choose a value**

You can add multiple actions, but you can add only one of each type of action for each set of dynamic behavior.

Dynamic behavior actions



b. Choose what happens for each action that you select.
For example, if you chose to **Display a message**, you can make it an **Info**, **Warning**, or **Error** message, and then you must enter the message that appears.

8. Select **Save**.

Result

The question appears with a dynamic icon to indicate that it has dynamic behavior.

The dynamic behavior appears in a card on the **Behavior** tab of the settings panel. You can select the card at any time to view or edit its details, or select the **Add behavior** button to add another dynamic behavior.

Development states of a form in Creator Studio

Think of your app as an online store that contains a catalog of items. To request something, a person needs to fill out a form. It's your job to create and customize forms for every item in your catalog using Creator Studio.

Summary: Forms have two states:

Draft

The state of a form when you're editing it. It can be a new form or a copy of a published form. Forms in this state do not appear in your catalog.

Published

The state of a fully-designed form that appears in your catalog.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

You want to publish forms only when they're ready—not some half-baked version you're working on. That's why forms have states.

Draft state for forms

Forms in the Draft state don't appear in your catalog of items. That means you can work on them without affecting your online catalog.

The forms you edit can be new, or copies of forms that have already been published. You can revise forms as much as you like in this state, but the revisions are kept behind the scenes until you publish the form. So, no worries about leaving a form unfinished when it's in Draft state!

Published state for forms

When you finish editing a form, you mark it as ready, which puts it in the Published state. This version of the form then appears in its specified catalog of items.

Note: The published form appears only on the non-production instance where you're developing your app. You must request that your admin deploy the app to production for the form to appear in the catalog on the production instance.

If you edit a form that's been published, Creator Studio creates a copy of the form, and puts the copy in the Draft state so you can work on it. The published version of the form remains available in your online catalog until you submit the revised copy of the form, which will replace the published form.

Add more forms to an app in Creator Studio

Add as many forms as you need to create catalog items for your app. For example, you could have an office booking app with three different forms: one for requesting parking spaces, one for requesting a desk, and one for reserving conference rooms.

Before you begin

To add forms to an app, you must be given permission to work on the app.

About this task

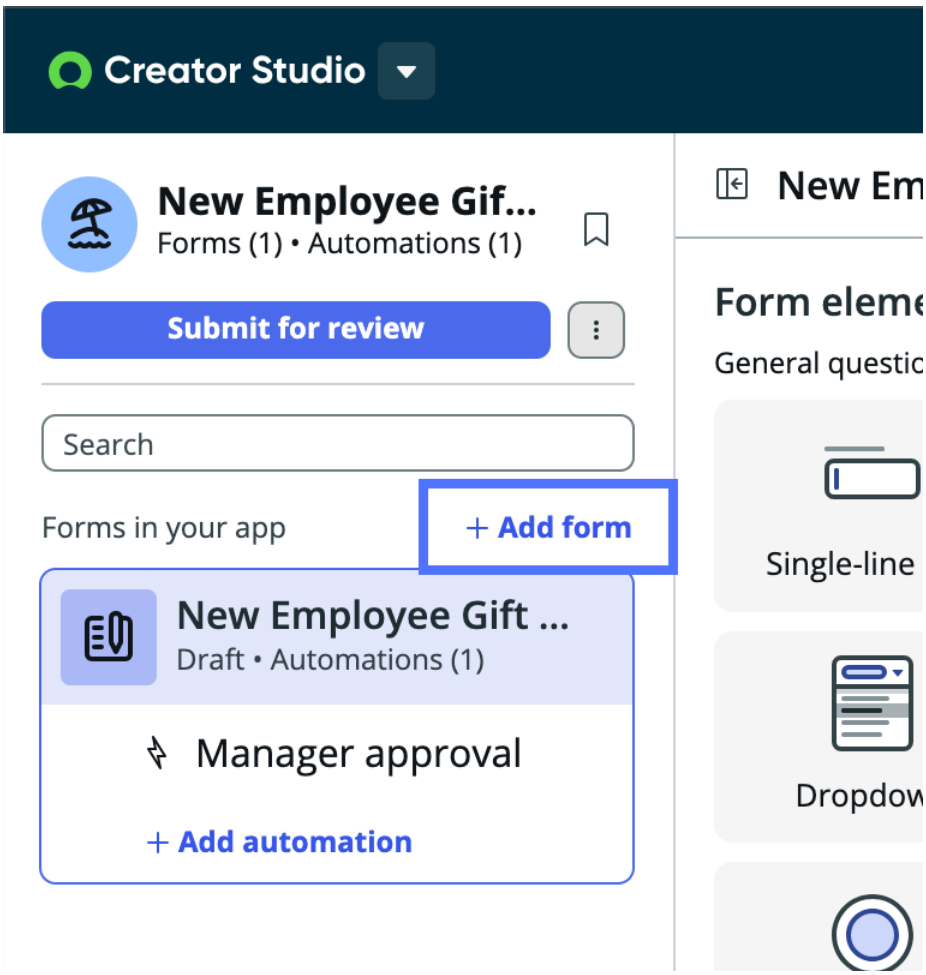
Note: For example, an IT fulfillment app could already have one form for requesting hardware accessories (such as a mouse or headphones), and you can add a second form for requesting software applications.

Procedure

1. Go to **All** > **App Engine** > **Creator Studio** to see all the apps on the Creator Studio home page.
2. Open the app that you want to add another form to.
3. Select **+ Add form** in the navigation panel.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add a form



4. Select the catalog template that you want to use to build the form from the list.
5. Select the **Apply template and continue** button.
6. Choose how you want to build the form:
   - To use AI, stay on the **Build with AI** tab and enter directions to provide a description for the form you want. To learn more, see Generating forms with AI.

     Note: You can select **Try an example** to see some examples of the available directions. To see a quick mock up of what the form will look like, select the **Generate form preview** button. If you want a different form, change the provided direction and select **Regenerate preview** for a quick look.

   - To manually build the form, select the **Build on your own** tab and enter the information for the form:

Form details

Field	Description
Form name	A descriptive name for the form.
Short description	Some information on what the form is used for.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Long description	Additional details for the form. You can include content like images and links to other sources of information (for example, a knowledge base article).

7. Select the **Save and edit form** button.

Result

Congrats: Hooray! You've added another form to the app.

What to do next

Next, you can edit the form to populate its questions and customize its layout. For more information, see Customize your form for an app in Creator Studio.

Edit the settings for a form in Creator Studio

Edit form settings if you need to change its basic attributes, such as its associated image or attachments are allowed.

Before you begin

To edit the settings for a form, you must be given permission to work on the app.

About this task

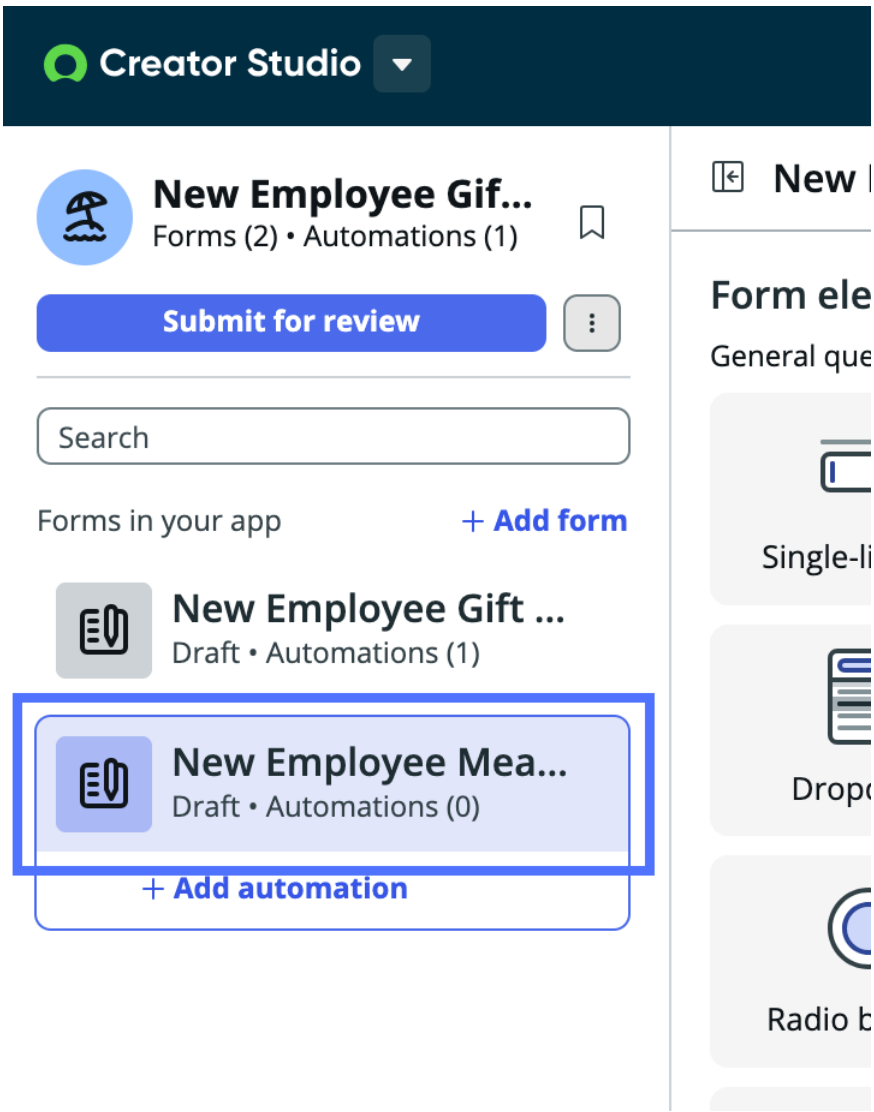
Note: After you create the app, its **Template** can't be edited. If you want to change it, create a new form using a different catalog template.

Procedure

1. Go to **All** > **App Engine** > **Creator Studio** to see all the apps on the Creator Studio home page.
2. Open the application that contains the form you want to edit.
3. Select the form you want to work on in the navigation panel.
   If you have multiple forms, make sure to select the form you want to edit settings for.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

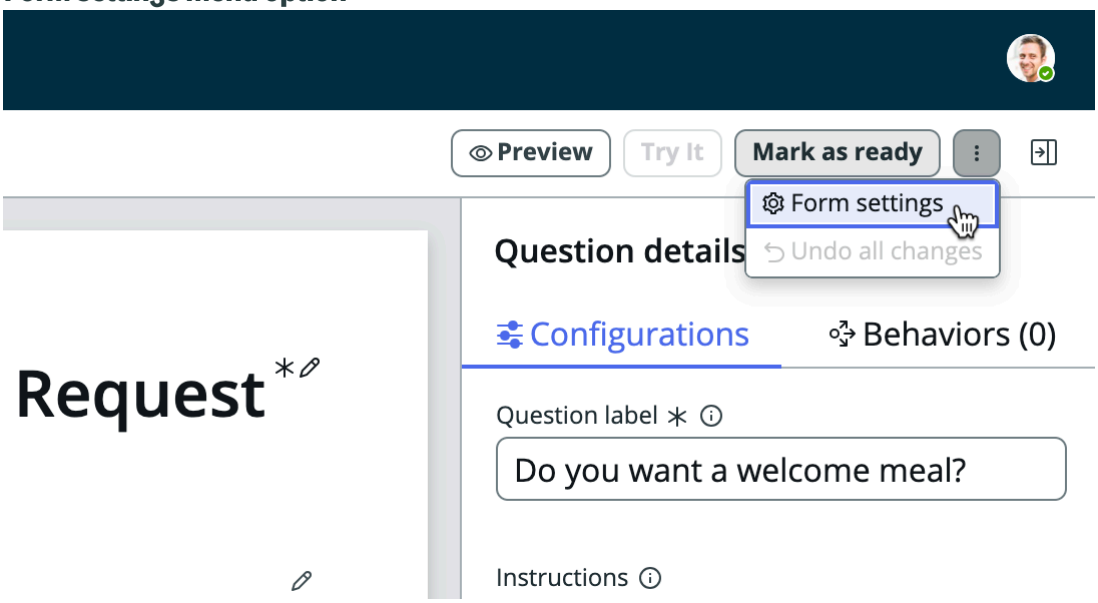
Select the form



4. Select the more actions icon ⋯.
5. Select **Form settings**.

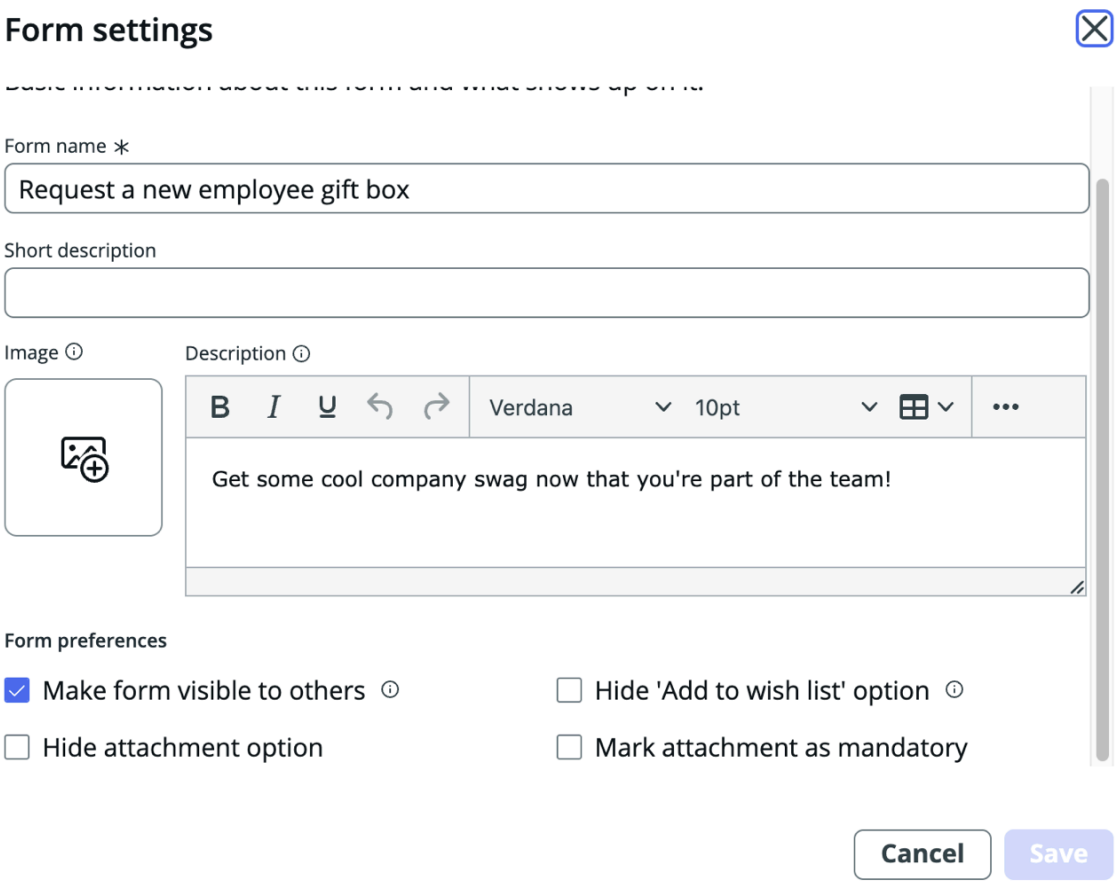
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Form settings menu option



6. Update settings on the **General** tab.
   For details on specific form settings, see Creator Studio form settings.

Form settings modal



7. Select the **Location** tab to update where the form appears in a catalog, how it's categorized, and which topics it appears for.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. Select the edit icon Edit for the **Catalogs and categories** card.

b. Select the catalog that represents the business area the app will use.
For example, you could select a service catalog that contains software and laptop cables for an IT fulfillment app.

c. Expand the caret for each catalog to see its sub-catalogs.

d. Select items in the catalogs, as many as you need.

e. Select the **Apply** button to save your changes.

f. Select the edit icon Edit for the **Topics** card.

g. Choose the **Taxonomy** page where you want the form to appear, such as **Employee**.
Taxonomy is a method of categorization that Employee Center uses.

h. Select the topic(s) that represent the Employee Center areas where you want the form to appear.
For example, choose a topic that contains technology services, and expand its carat to see each of its sub-topics. Find out more about topics in Associate a catalog item with a taxonomy topic in Employee Center, and more about taxonomy, which is a categorization method, in Unified Taxonomy for Employee Center.

i. Select the topics where you want the form to appear, as many as you need.

j. Select the **Apply** button to save your changes.

8. Select the **Access** tab to change which roles and groups can view and submit the form.

a. Select the edit icon Edit to edit users that the form should be **Available for**.

b. Select the roles and groups that should have access to the form.

c. Select the **Apply** button to save your changes.

d. Select the edit icon Edit to edit users that the form should be **Not available for**.

e. Select the roles and groups that shouldn't have access to the form.
Work with your admin to restrict or provide access to the roles and groups for this setting in non-production and production environments. For more information, see Administering user access for deployed Creator Studio apps.

f. Select the **Apply** button to save your changes.

9. Select **Save all settings**.

Result

The form's settings are updated. Remember: You just updated the settings for the form you selected, not for all of the app's forms if it has multiple.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Publish a form for your app in Creator Studio

Publishing forms once they're ready makes them available as catalog items in the production instance for published apps.

Before you begin

To publish a form, you must be given permission to work on the app.

About this task

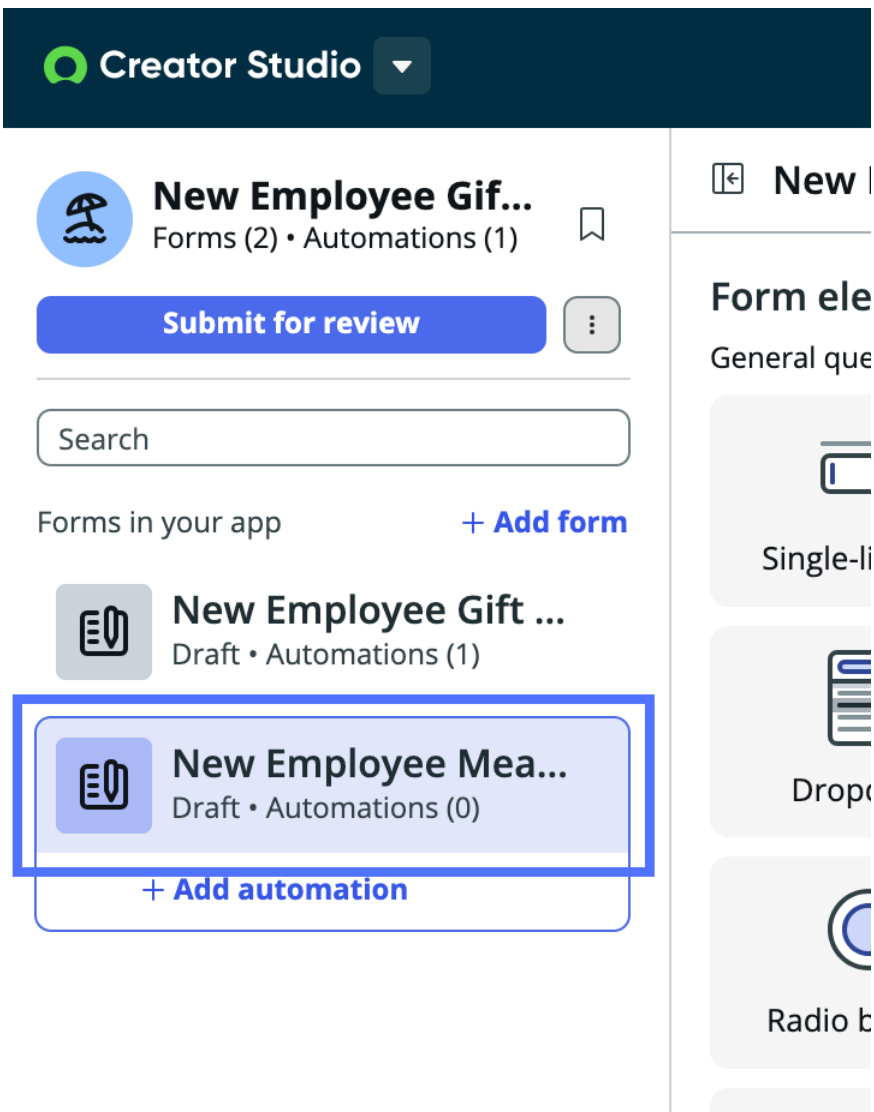
After you publish a form, it's available in the specified catalog on the instance you're working on. However, you still need to deploy your app to the production instance for users to access the form. For more information, see Deploying your Creator Studio app.

Procedure

1. Go to **All** > **App Engine** > **Creator Studio** to see all the apps on the Creator Studio home page.
2. Open the application that contains the form you want to publish.
3. Select the form you want to publish in the navigation panel.
   If you have multiple forms, make sure to select the form you're publishing.

   Note: If your app has a lot of forms, you can use the search box in the navigation panel to search for a specific form.

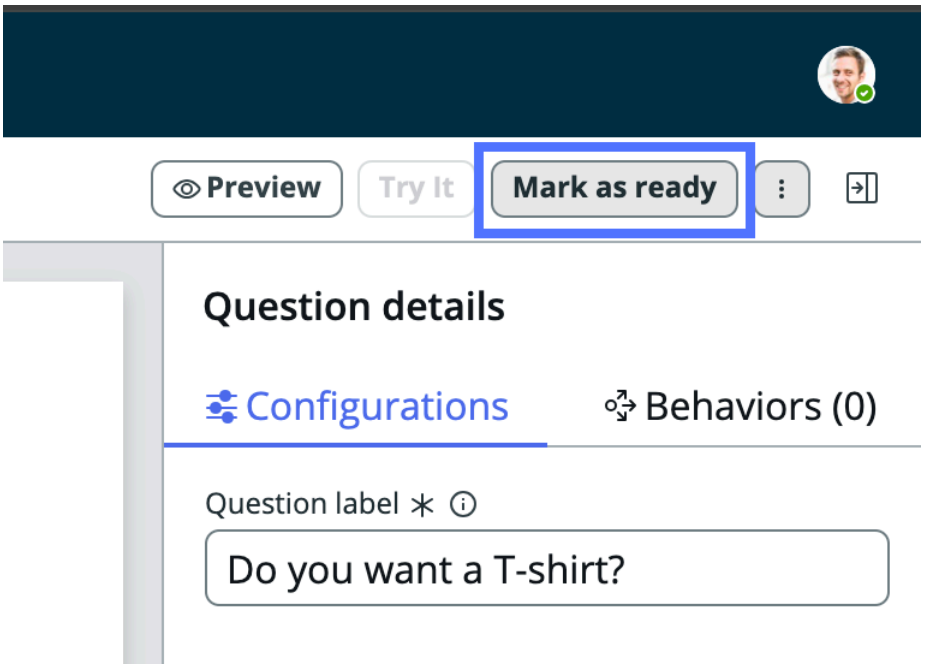
Select the form



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

4. Select the **Mark as ready** button.

Publish a form



Note: If you haven't specified where the form will live, such as in a catalog or in the Employee Center, you're prompted to **Edit location setting**. For more information, see Edit the settings for a form in Creator Studio.

Result
The catalog item for the form is ready to be deployed with your app. Once the app is deployed, the form is available in the associated service catalog. If the form's app hasn't been deployed, you need to deploy it.

Note: If you published a form and then edit it, you can't use the version that's being edited in a condition for a playbook, for example as a trigger for the automation. However, you can use the published version of the form or publish the version that you're editing, which makes its question answers available to use in playbook conditions.

Working with published forms

After a form has been published, you can update it, undo changes, and hide it from being visible.

Changing published forms

It's inevitable. You deploy your app and find that you need to revise the questions on your form. No problem! Just edit the form as you did when you created it.

Editing a form creates a new Draft version of the form that you can update and then **Mark as ready** when you're finished. Any changes you save to the Draft version won't affect the catalog item in your deployed app. To update the forms in the deployed app, you must redeploy the app.

Undoing form changes

You can select the **Undo all changes** option, available in the form header's more options icon (⋯), to reset a form to the most recently published version.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Hiding forms

Hiding a catalog item for your app's form effectively makes the form inactive. Hidden forms are unavailable in both the app and the catalog it belongs to. If you want to remove a published form from your app, hide it.

Deleting forms

You can delete only unpublished forms.

- If a form is in Draft state, you can just choose not to **Mark as ready**.
- If you know that you'll never need a Draft form again, you can delete it. For more information, check out Delete a form in Creator Studio.
- If the form is in a Published state, you can hide it so it no longer appears in the catalog. See Hide a form from use in the ServiceNow AI Platform in Creator Studio for more information on hiding forms.

Deleting forms and questions in Creator Studio

Deleting unpublished forms and questions from Creator Studio removes them completely from the ServiceNow AI Platform.

To delete a form or question, you must have the admin role, or be an owner or editor of the app. For details on owning and editing apps, see Collaborating with others to build apps in Creator Studio.

If a form has been published, you can hide it to make it inactive without deleting it. Check out Hide a form from use in the ServiceNow AI Platform in Creator Studio for more information.

Deleting forms

Deleting a form completely removes the form's record from the ServiceNow AI Platform, including from Catalog Builder.

Note: You can't delete the following items:

- Forms after they've been published
- Drafts of previously published forms

Deleting questions from forms

Deleting a question from an unpublished form completely removes the record for the question from the ServiceNow AI Platform, including from Catalog Builder.

However, if you delete a question from a form that has already been published, the record for the question remains in Catalog Builder but is marked as inactive.

To delete a question from a form, select the question and the select the delete icon. You must then confirm the deletion in a modal.

Deleting a form question



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Delete a form in Creator Studio

Delete an unpublished form from a Creator Studio app to remove it completely from the ServiceNow AI Platform.

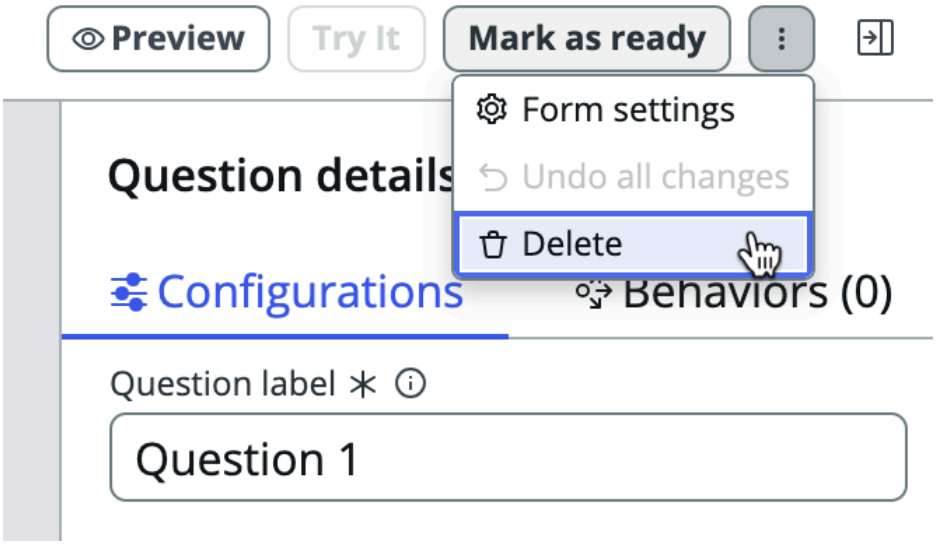
Before you begin

To delete a form, you must be given permission to work on the app.

Procedure

1. Go to **All** > **App Engine** > **Creator Studio** to see all the apps on the Creator Studio home page.
2. Open the app that contains the form that you want to delete.
3. Select the form that you want to delete in the navigation panel.
   If you have multiple forms, make sure to select the form you're deleting.
4. Select the More actions icon ⋯.
5. Select **Delete**.

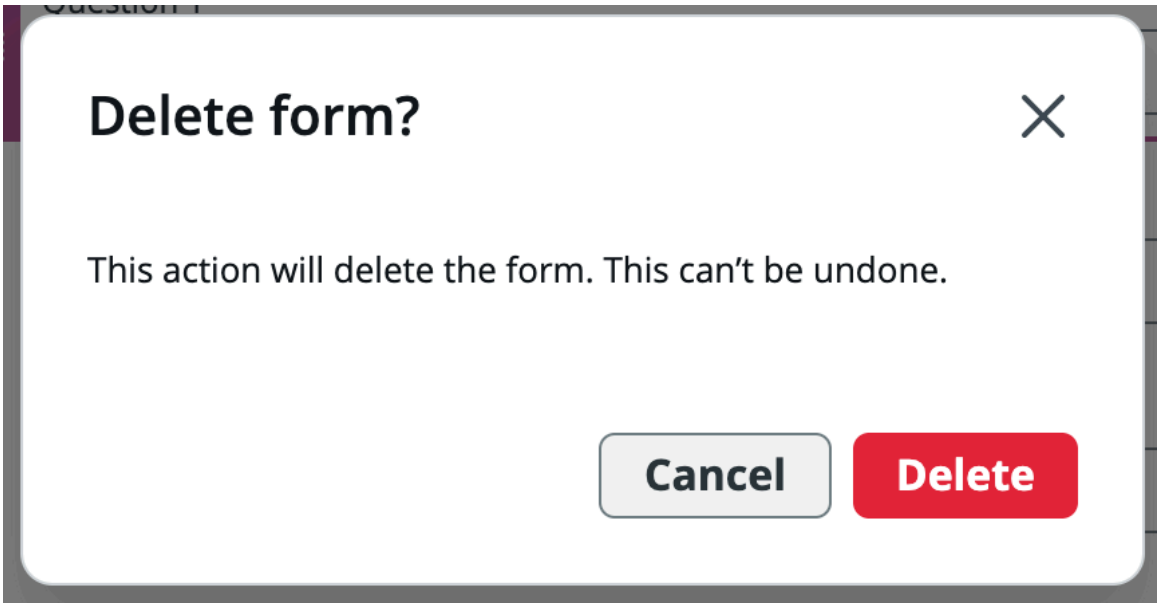
Delete form menu option



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

6. Select **Delete** in the confirmation modal.

Confirmation modal



Result
The form is deleted, and the next form associated with the app appears in the editor. After deletion, the form's record is removed completely from the ServiceNow AI Platform.

Hide a form from use in the ServiceNow AI Platform in Creator Studio

Hiding a catalog item for your app's form effectively makes the form inactive. Hidden forms are unavailable in both the app and the catalog it belongs to.

Before you begin
To hide a form, you must be given permission to work on the app.

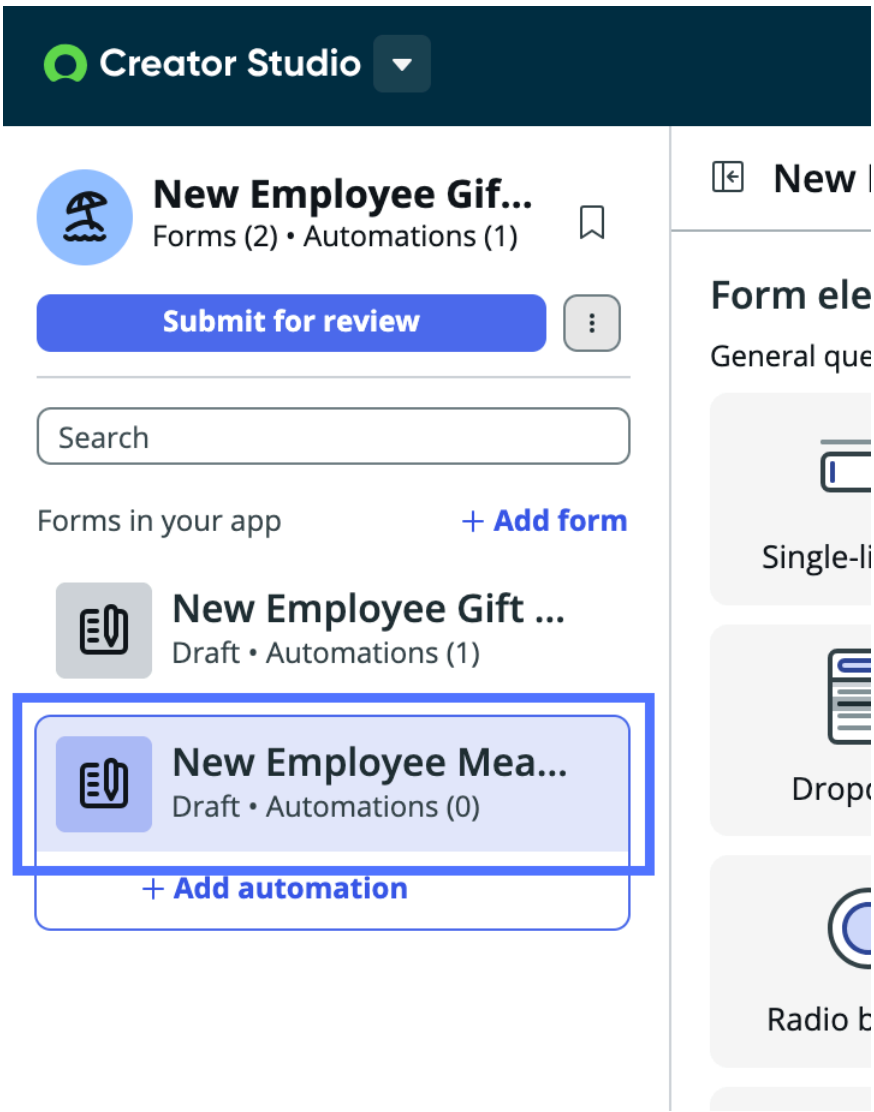
About this task
If a form hasn't been published yet, you can delete it instead of hiding it. See Delete a form in Creator Studio for details.

You can hide only forms that have already been published. For more information, see Deploying your Creator Studio app.

Procedure
1. Go to **All** > **App Engine** > **Creator Studio** to see all the apps on the Creator Studio home page.
2. Open the app that contains the form you want to hide.
3. Select the form that you want to hide in the navigation panel.
   If you have multiple forms, make sure to select the form you're hiding.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

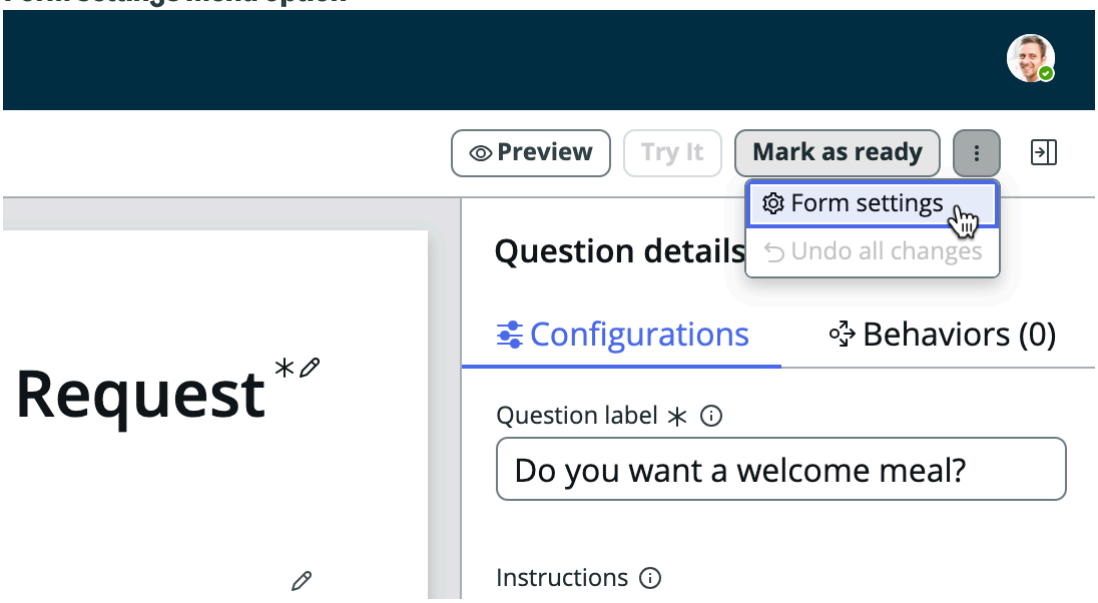
Select the form



4. Select the More actions icon ⋯.
5. Select **Form settings**.

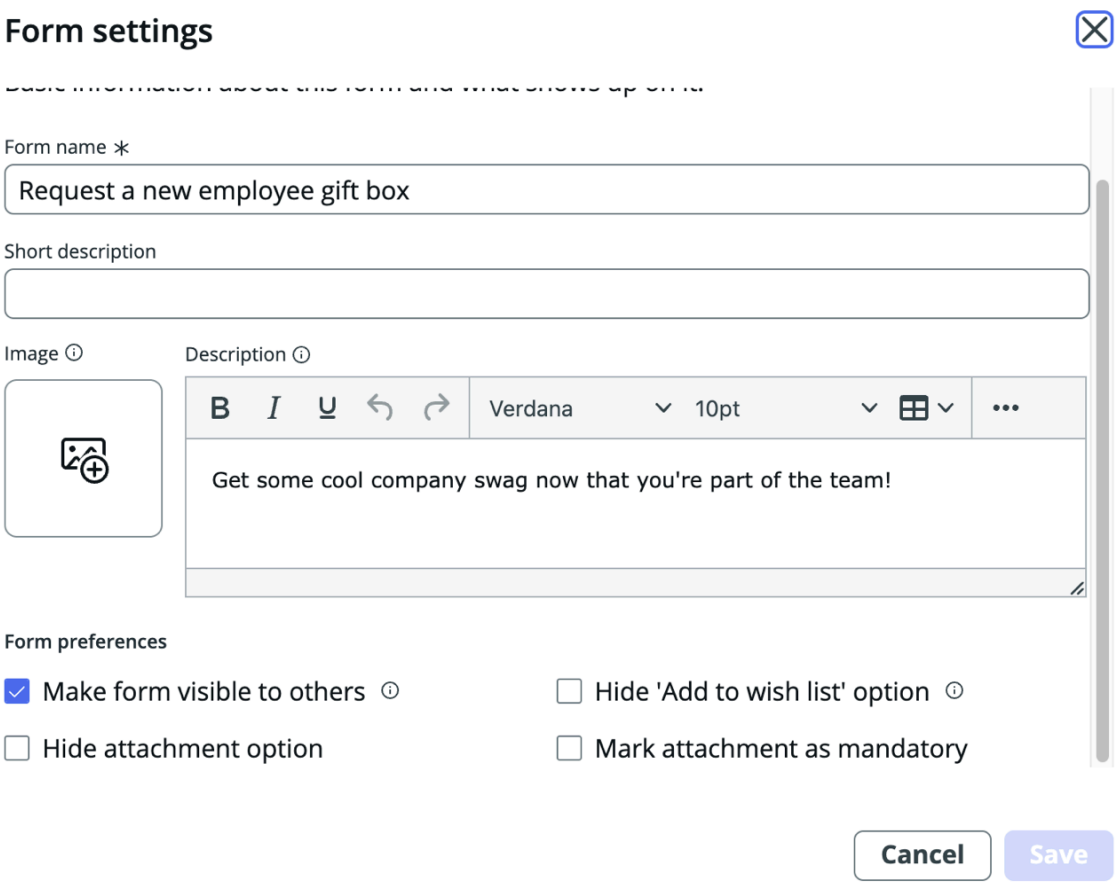
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Form settings menu option



6. De-select the **Make form visible to others** option.

Form settings modal



7. Select **Save**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

What to do next

Your admin must re-deploy the app to hide the inactive form (catalog item) on the production instance.

Working with automation in Creator Studio

How do you simplify and streamline tasks for fulfillers? Use automation! For example, you can have a task assigned automatically to someone based on what type of request it is. We'll use playbooks to accomplish adding this automation to your app.

Summary: After reading this section, you'll understand how to use automation to do the following things:

- Use playbooks to guide agents and fulfillers through processes from start to finish, improving the customer experience and the way tasks are carried out.
- Create a consistent process for your tasks or activities to be carried out.
- Consolidate business decisions and processes across the organization.

Note: Because playbooks are based on forms, you must publish a form (mark it as ready) before you use it to create a playbook. For more information, see Publish a form for your app in Creator Studio.

Playbooks are a series of steps triggered by an event. You can add multiple playbooks to an app if needed

Key terms:

Activity

Activities in a playbook complete automated tasks, such as obtaining approvals, assigning records, and sending email.

Automation

Tasks or requests moving through a workflow without manual intervention.

Playbook

A set of actions based on an initiating action called a trigger.

Record

A task or request that must be acted on. All rows in ServiceNow tables are individual records. Records have different states depending on what actions have been taken.

Trigger

The event that starts making the playbook run.

Workflow

The path a record takes from creation to completion.

What is automation?

Automation in Creator Studio enables you to create simplified playbooks, which are basic, automatic workflows. Simple playbooks consist of a trigger (such as a record changing states, or a user choosing a specific answer for a question) and an activity (such as the record being assigned to a new person). Playbooks enable you to add the following:

- Activities to your automation to define the workflow's path.
- Decisions to create different paths, or branches, depending on the decision's outcome.
- Custom activities defined by your admin or ServiceNow AI Platform owner.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Playbooks take the guess work out of assigning tasks and help users focus on the tasks and information that matter to their job. A well-designed playbook can do these things:

- Automatically start or trigger changes for different types of records, such as task assignments, or specific fields on the submitted form's record.
- Reuse activities and business decisions from other workflows you've already created.
- Clearly show the next steps that fulfillers must take to move to the next step of completing the record's life cycle.

There are several types of activities that you can add to playbooks that, when paired with decisions, remove the manual thinking behind what to do next when addressing tasks. Depending on your configuration, some of the activities you can add include:

- Request approval for a task
- Assign the task to someone
- Create a new task as a next step
- Send an email
- Choose a custom action configured by your admin
- Update fields on a submitted form's record

Note: In addition to creating custom activities, your admin can also hide standard activities, so you may not see all of the above options.

You can even use answers from a playbook's associated form to power activities. For example, if someone's requesting a gift box and they answer `Yes` to the question "Do you eat meat?" then you can create an Assign to activity to route the gift box to the Meat department for fulfillment.

You can also use a placeholder activity for more complex activities that must be configured outside of Creator Studio, or for another activity to be swapped in later. Complex activities are configured in a different tool called Workflow Studio.

What goes into building automation?

Automations in Creator Studio include the following parts.

Automation parts

Concept	Description
Playbook	An automatic workflow that guides users through tasks based on the record's status changes. Find out more about creating playbooks in Add an automated playbook to an app in Creator Studio.
Trigger	The event that specifies when the playbook should start running, such as when a form is submitted or updated to a specific status. Check out Edit the trigger for a playbook in Creator Studio for more information.
Activity	The thing that happens during the playbook, such as assigning a task or requiring approval.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Automation parts (continued)

Concept	Description
	Note: If you're already familiar with optional activities in playbooks, that's great! However, to keep automations streamlined and simple, Creator Studio doesn't support optional activities. If you must create optional activities, you can add a placeholder activity in Creator Studio and then edit the placeholder in Workflow Studio or swap it out for another activity later in Creator Studio. Additionally, admins can create custom activities for you to include in a playbook. Learn how to add activities in Add activities to an app's playbook in Creator Studio.
Decision	A series of inputs, conditions, and results that create different branches, or conditional paths that users follow based on the decision's outcome. Get the scoop on adding decisions in Add a decision to an app's playbook in Creator Studio.
Branch	A distinct path from a decision's outcome, or what happens for each of the decision's possible choices. For more information, see Add a decision to an app's playbook in Creator Studio.
Parallel process	Branches for activities and stages that run in parallel to another branch of activities. Learn about parallel processes in Add a parallel process to an app's playbook in Creator Studio.
Connectors	Lines that connect activities and decisions in a playbook that enable you to add more activities, decisions, and parallel processes. You can drag connectors between activities to create parallel activities.

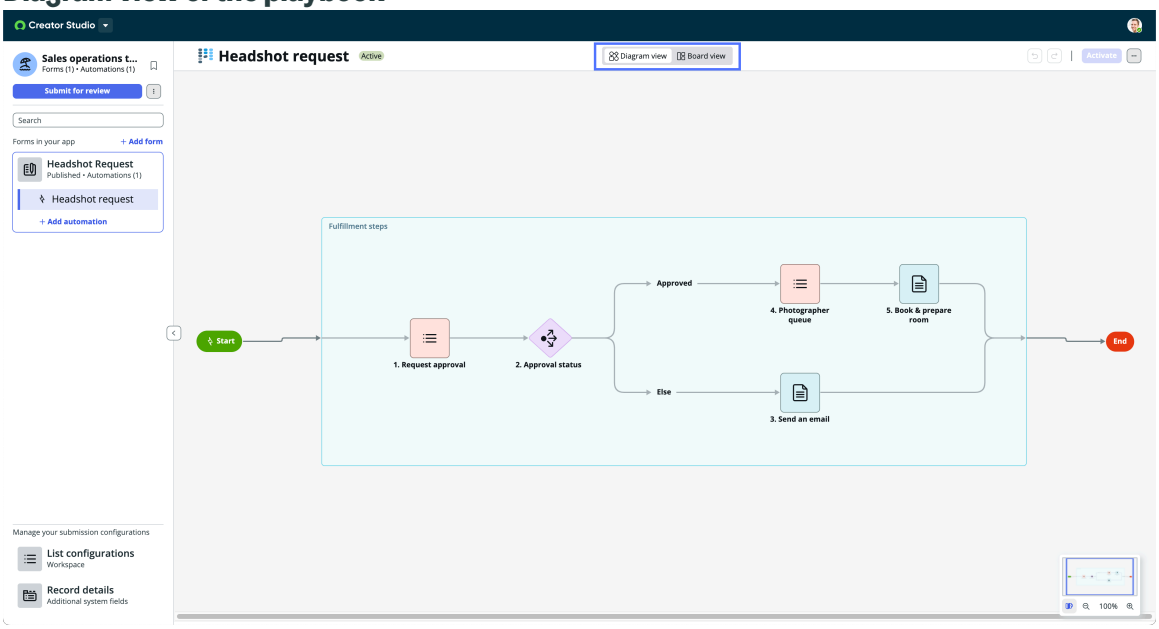
Board and Diagram views of a playbook

Creator Studio offers two ways of working with playbooks: the Board view and the Diagram view.

The Diagram view is a visually intuitive diagram that shows you the order that activities occur in the playbook. When you select an activity, such as the trigger or a decision, a modal pops up where you can edit its details. You must use the Diagram view to add decisions.

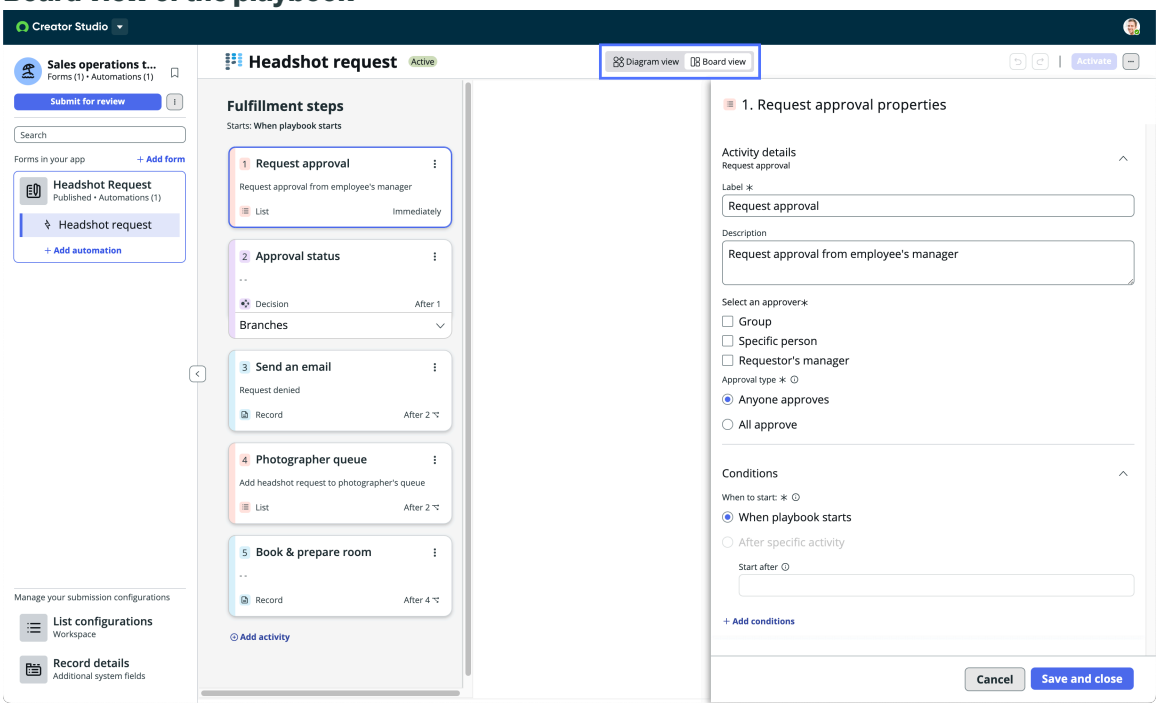
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Diagram view of the playbook



The Board view is a more traditional layout, which shows each activity as a card in the navigation panel. When you select an activity, for example a task, its editable details appear in the canvas for you to work with them. You must use the Board view to rearrange activities.

Board view of the playbook



This help document will primarily give instructions using the Diagram view, with alternate directions on how to complete automation tasks in the Board view.

Where do playbooks run in the ServiceNow AI Platform?

Users work on requests in the Request App Workspace on the ServiceNow AI Platform, which lists a record for each request. The playbooks run on the specified records in the app's workspace category.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Specifically, the **Catalog tasks** tab of the record details walks users through the activities they must do to complete the task and close the record.

Workflow for creating automations in Creator Studio

Provide your organization's fulfillers with guidance when addressing records generated from the app by adding automated playbooks to it.

Before you begin

Note: Because playbooks are based on forms, you must publish a form (mark it as ready) before you use it to create a playbook. For more information, see Publish a form for your app in Creator Studio.

To work with automations, you must be given permission to work on the app.

Procedure

1. Create an app.
   If you have an existing playbook that's close to what you want to create, you can duplicate and edit it, which may be faster than creating an entirely new playbook. For more information, see Copy an automated playbook in Creator Studio.
2. Add an automated playbook to the app.
   For more information, see Add an automated playbook to an app in Creator Studio.
3. Add activities to the playbook.
   For more information, see Add activities to an app's playbook in Creator Studio.
4. Create different branches of the playbook by adding decisions to it.
   For more information, see Add a decision to an app's playbook in Creator Studio.
5. **Optional:** Create any parallel processes to run at the same time in the playbook.
   For more information, see Add a parallel process to an app's playbook in Creator Studio.
6. Make any additional changes to the playbook's trigger.
   For more information, see Edit the trigger for a playbook in Creator Studio.
7. Make any additional changes to the playbook settings.
   For more information, see Edit a playbook in Creator Studio.
8. Activate the playbook when it's ready.
   For more information, see Activate a playbook in Creator Studio.

Add an automated playbook to an app in Creator Studio

Create an automated playbook, which is a process that runs whenever a record is created or updated by the app's specified form.

Before you begin

Playbooks run on the records generated by catalog items you create for the app, so you must add forms to the app before you can add automation. You can use only published forms (that is, forms that have been marked as ready) to create automations. For more information, see Working with forms in Creator Studio.

To add a playbook, you must be given permission to work on the app.

About this task

Note: For example, you can build a travel request app where people can ask for permission for different types of travel, which automatically routes the request to a manager for approval.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

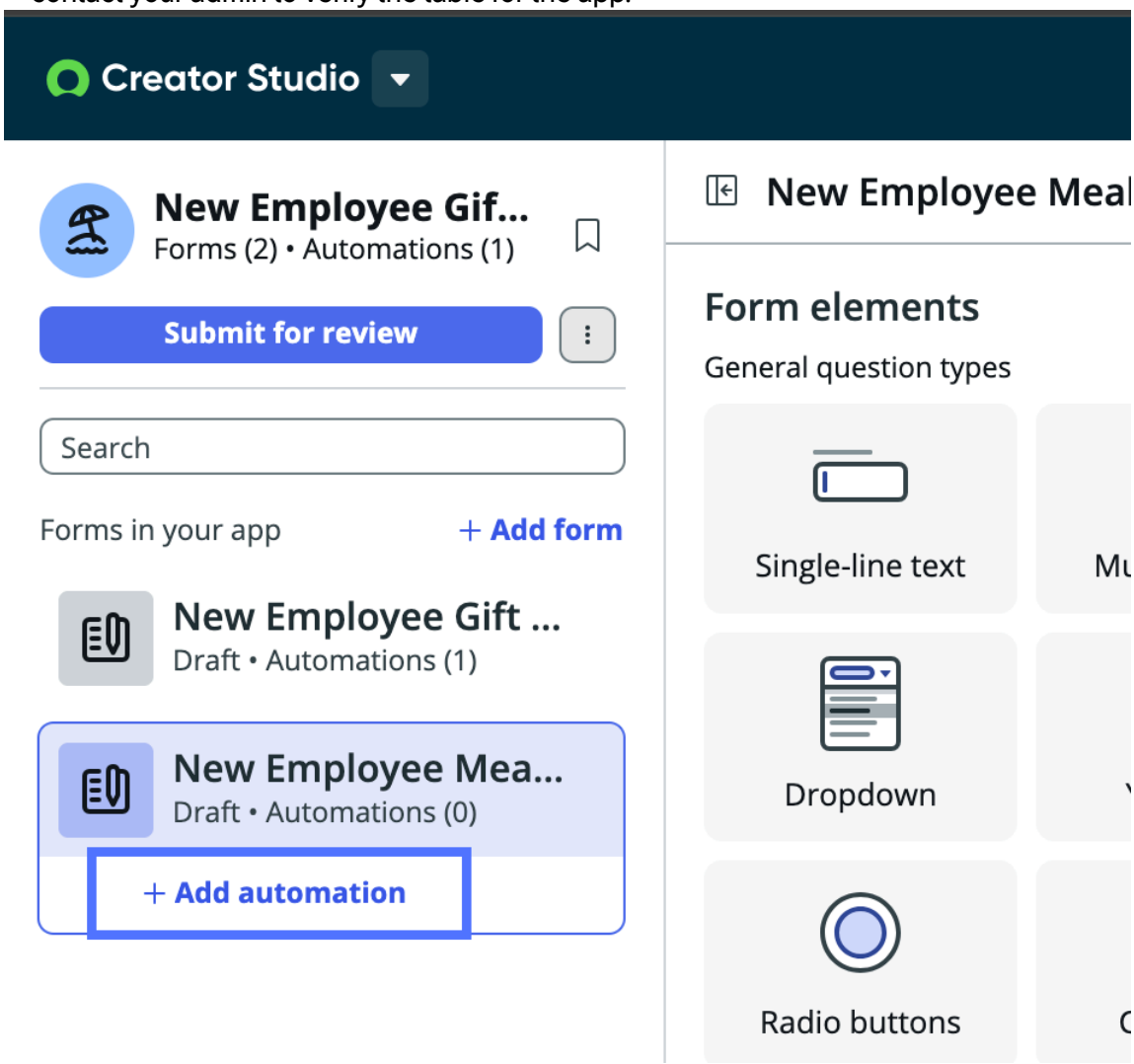
Procedure

1. Go to **All** > **App Engine** > **Creator Studio** to see all the apps on the Creator Studio home page.
2. Open the app that you want to add automation to.
3. Select **+ Add automation** for the form that you want to add a playbook to.

   Note: If your app has a lot of forms, you can use the search box in the navigation panel to search for a specific form.

   The following rules apply:

   - If the form is still in draft, a message pops up telling you that you must publish the form first. Check out Publish a form for your app in Creator Studio for more information.
   - You can select only published forms to run the playbook. If you published a form and then edit it, you can't use the version that's being edited in a condition for a playbook, for example as a trigger for the automation. However, you can use the published version of the form or publish the version that you're editing, which makes its question answers available to use in playbook conditions.
   - The table for the form you choose is the app's associated table, which is specified when the app is created. However, your admin can change the table, see Administering an app's associated table for details. If you get a message that playbook creation is unavailable, contact your admin to verify the table for the app.



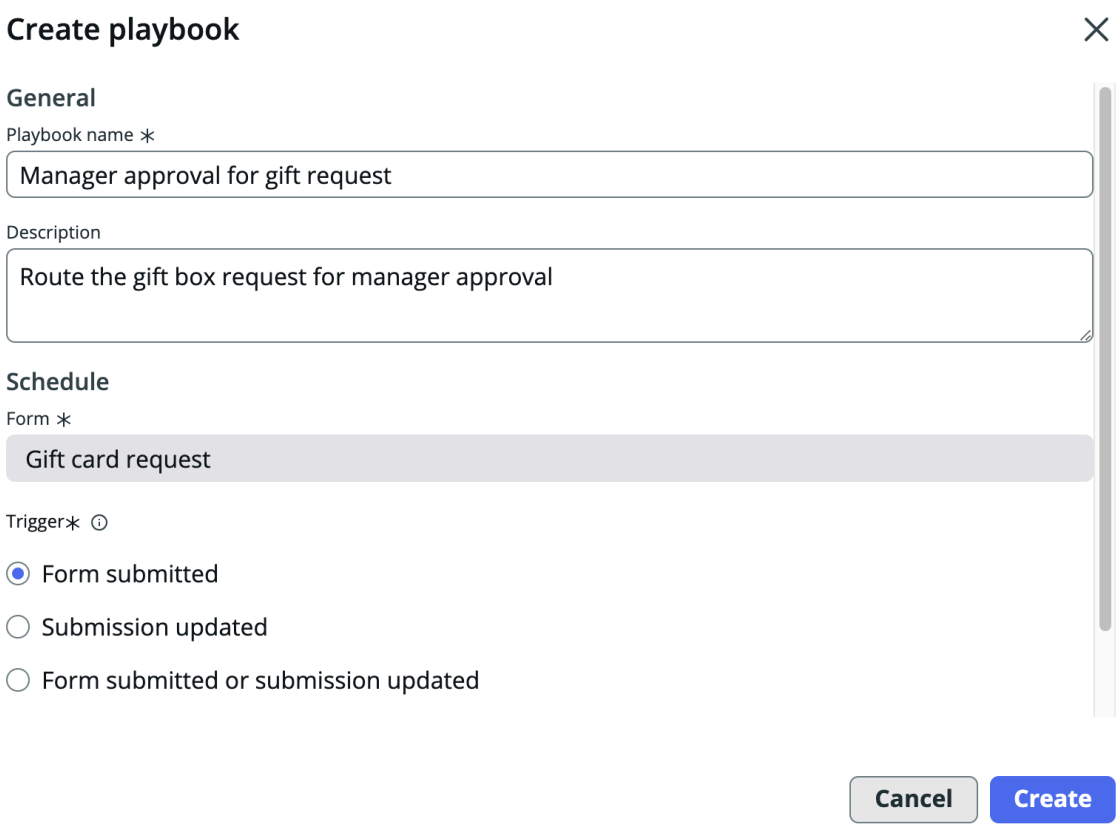
4. Specify the playbook's General attributes in the Create playbook modal.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

General playbook definition fields

Field	Description
Playbook name	Descriptive name for the playbook you're creating.
Description	Brief explanation of what the playbook does, for example, the end goal for the record type.

Create a playbook



5. Specify the Schedule for the playbook.

   a. Select the type of **Trigger** that initiates the playbook.

Playbook trigger options

Trigger	Description
Form submitted	Start running the playbook when a user submits the form you chose.
Form updated	Start running the playbook when a user updates the form you chose.
Form submitted or updated	Start running the playbook when a user submits or updates the form you chose.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: You can't change an playbook's trigger type after you finish creating the playbook. Instead, create a new playbook with a different trigger.

However, you can edit the trigger condition, such as making the playbook run conditionally based on a specific answer to a question. For more information, see Edit the trigger for a playbook in Creator Studio.

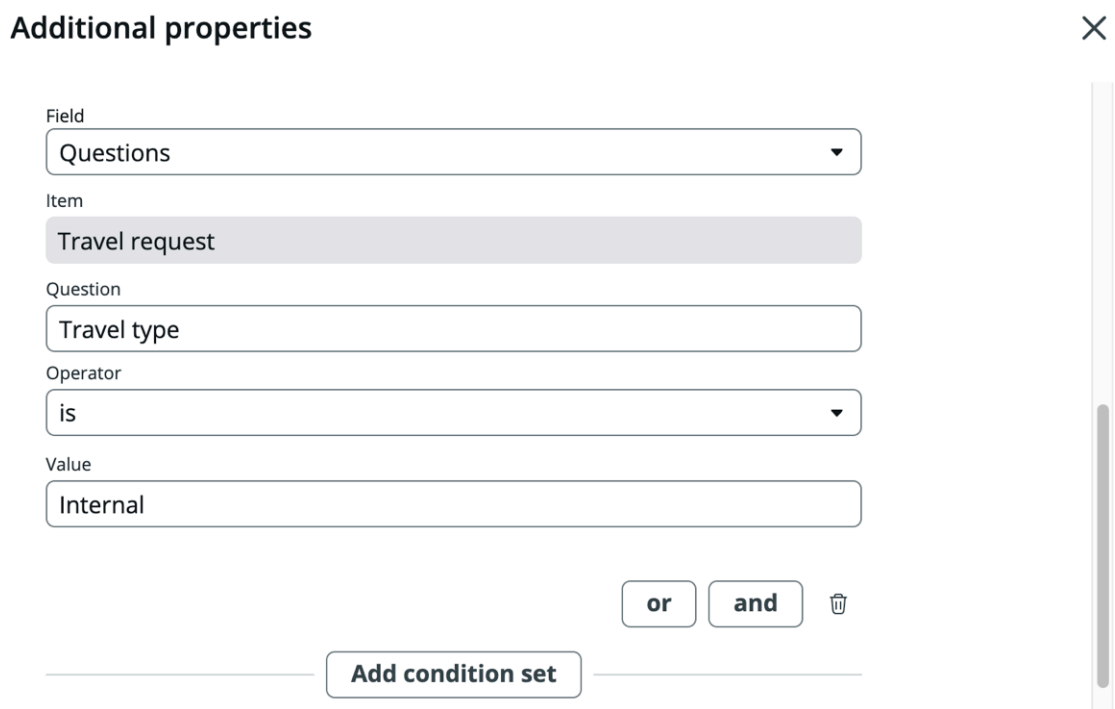
b. If you chose a trigger that includes a form being updated, specify how often that app should **Run your playbook**.
The options are:

- **Once**
- **For each unique change**
- **Only if not currently running**
- **For every update**

c. Specify the conditions that must be met for the playbook to begin running by selecting **Add conditions**.

- If you want to trigger the playbook based on the value of a column in a table, select the **Field** that you want to be the trigger, as well as its condition **Operator** and the specific trigger **Value**. For example, when a **Start date** is **after** the **Date** needed.
- If you want to trigger the playbook based on the response from a form, select **Questions** as the trigger **Field**. Then select the question you want in the **Question** field, the condition **Operator** and the answer's **Value**.

Question answer as trigger for an automation



Add as many conditions as you need. For more information, see Create a condition statement using the condition builder.

d. Select **Create**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Result

Congrats: Congratulations, you've created an automation!

What to do next

Next, you need to add activities to the playbook. For more information, see Add activities to an app's playbook in Creator Studio.

Copy an automated playbook in Creator Studio

Duplicate an existing playbook and make small changes to it, which may be faster than creating an entirely new automated playbook.

Before you begin

Because playbooks are based on forms, you must publish a form (mark it as ready) before you use it to create a playbook. For more information, see Publish a form for your app in Creator Studio.

To copy a playbook, you must be given permission to work on the app.

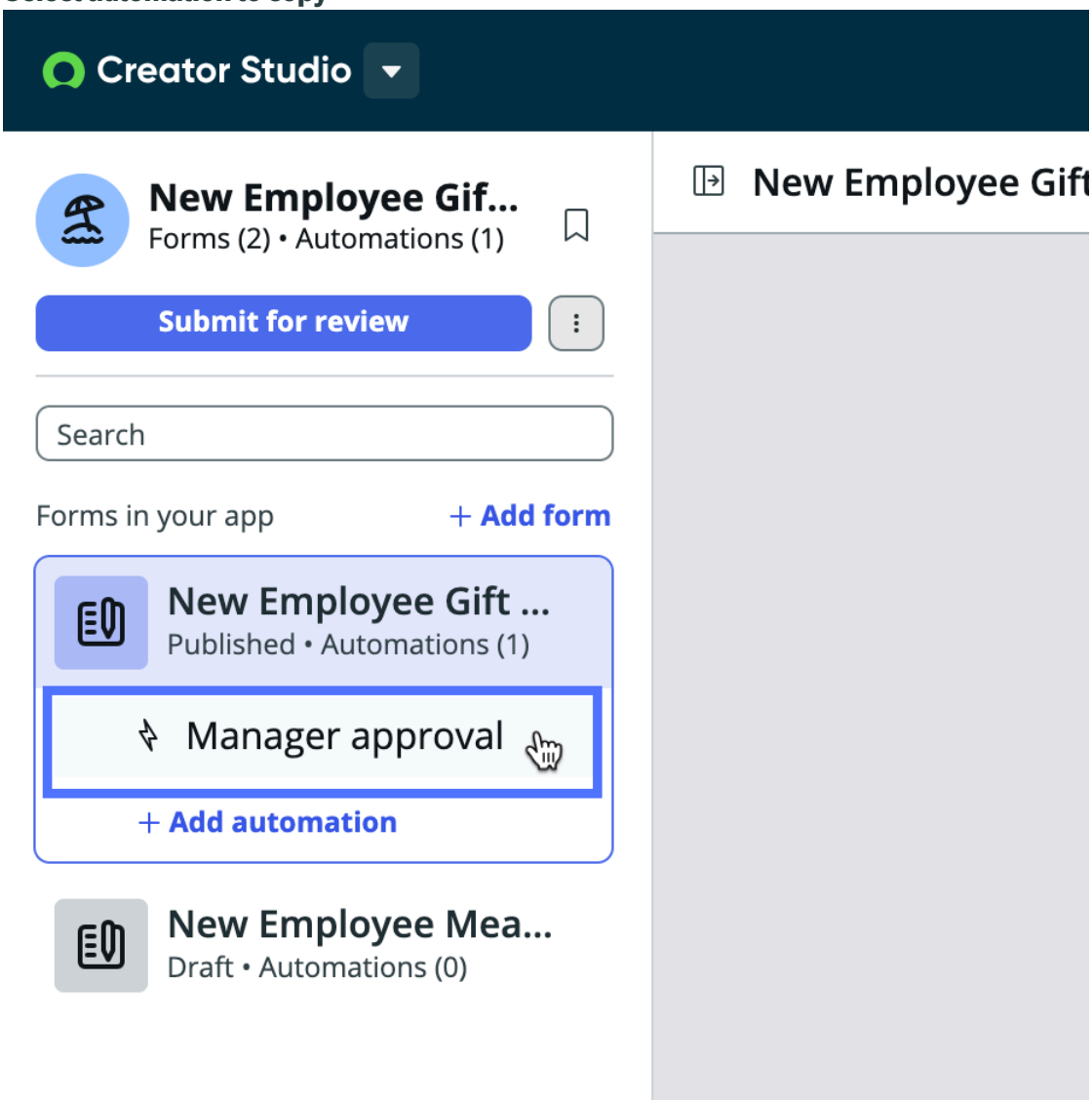
Procedure

1. Go to **All** > **App Engine** > **Creator Studio** to see all the apps on the Creator Studio home page.
2. Open the app with the playbook you want to copy.
3. Select the form that contains the automation that you want to copy.
4. Select the automation that you want to copy.

Note: If your app has a lot of forms, you can use the search box in the navigation panel to search for a specific form.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

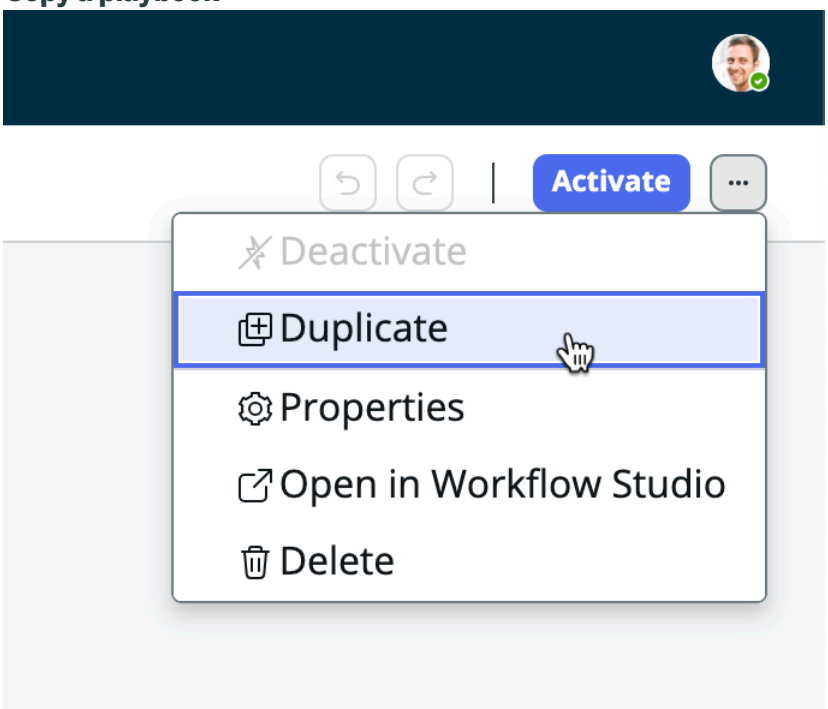
Select automation to copy



5. Select the more actions menu icon ⋯.
6. Select **Duplicate**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Copy a playbook



7. Choose the **Duplicate** button in the confirmation modal.

Result
The duplicated playbook opens, and appears in the navigation panel below the original playbook.

Add activities to an app's playbook in Creator Studio

Add activities to an app's playbook to specify what the automation does to the designated record type. An activity defines what actually happens when a playbook executes.

Before you begin
To add activities to a playbook, you must be given permission to work on the app.

About this task
The activities that you can add to a playbook are limited by your administrator to ensure you use the correct ones for your organization. Your admin may have created some custom activities for you to choose from, or hidden some standard activities.

Procedure
1. Go to **All** > **App Engine** > **Creator Studio** to see all the apps on the Creator Studio home page.

2. Open the app that contains the playbook you want to add an activity to.

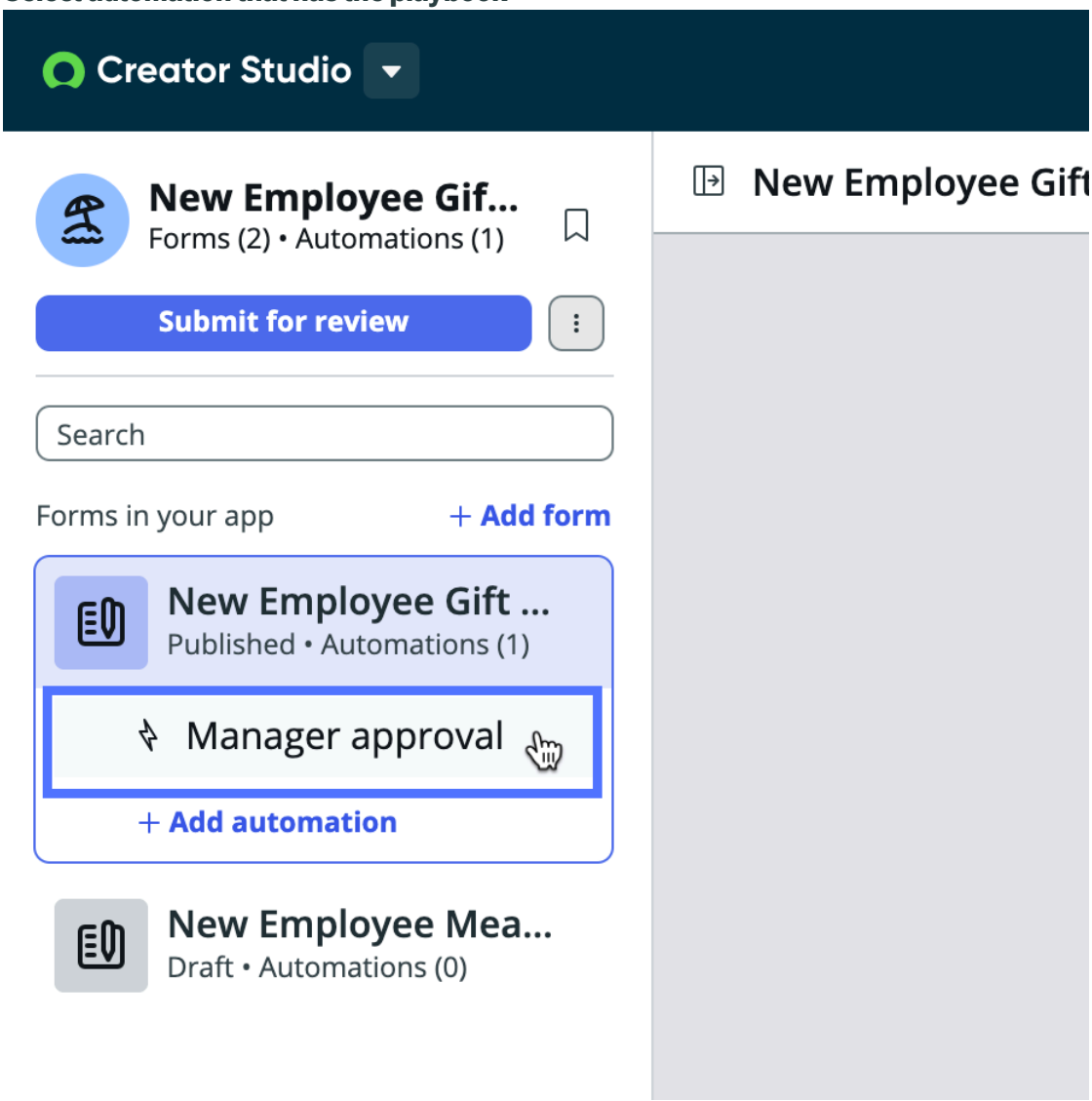
3. Select the form that contains the playbook that you want to add an activity to.

4. Select the automation that contains the playbook that you want to add an activity to.

> **Note:** If your app has a lot of forms and automations, you can use the search box in the navigation panel to search for a specific automation.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Select automation that has the playbook



5. Select the add icon on the connector where you want to add an activity and choose the square **Add an activity** icon ( ) in the menu that pops up.

How to add an activity in Board and Diagram views

View	Steps
Diagram	**a.** Select the **+** icon in the stage. **b.** In the mini-picker, select the square icon to add an activity.
Board	Select **+ Add activity**.

6. Choose the type of activity that you want from the Activity library pop-up.

Note: In addition to the following standard activities, you may see some custom activities that your admin created.

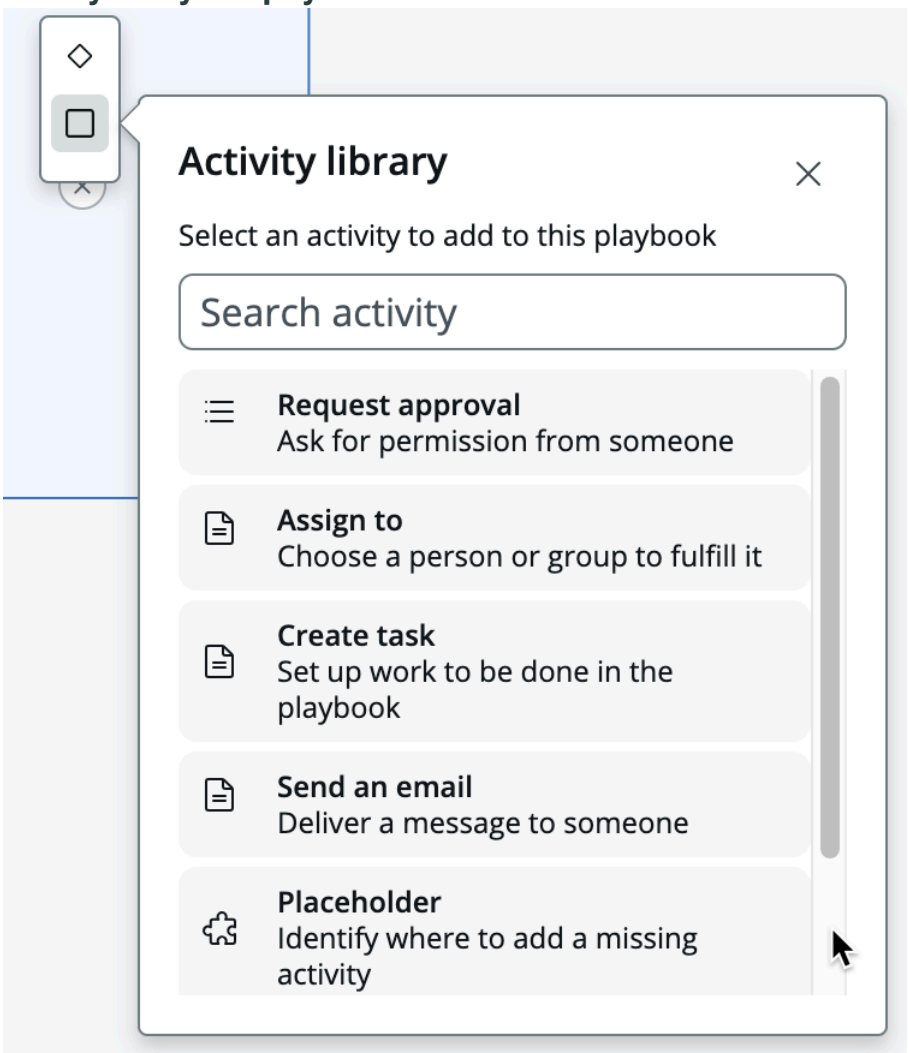
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Types of activities

Activity type	Description
Request approval	Ask someone for permission to accomplish a task.
Assign to	Choose a person who should fulfill the task.
Create task	Specify a process that must be done as part of the playbook.
Send an email	Send an email to one or more people. You can specify images and enrich text for the email that gets sent automatically.
Placeholder	Set an undefined activity to be specified later, or a more advanced activity such as an email notification, when an activity is completed. Placeholder activities don't have any logic assigned to them yet, and must be edited in Workflow Studio. Or, you can swap them out later for another type of activity in Creator Studio.
Update submission	Update the value of the specified fields on the record that the form generates when a user submits it.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Activity library for a playbook



Your new activity and its Activity properties panel appear, where you define what the activity is.

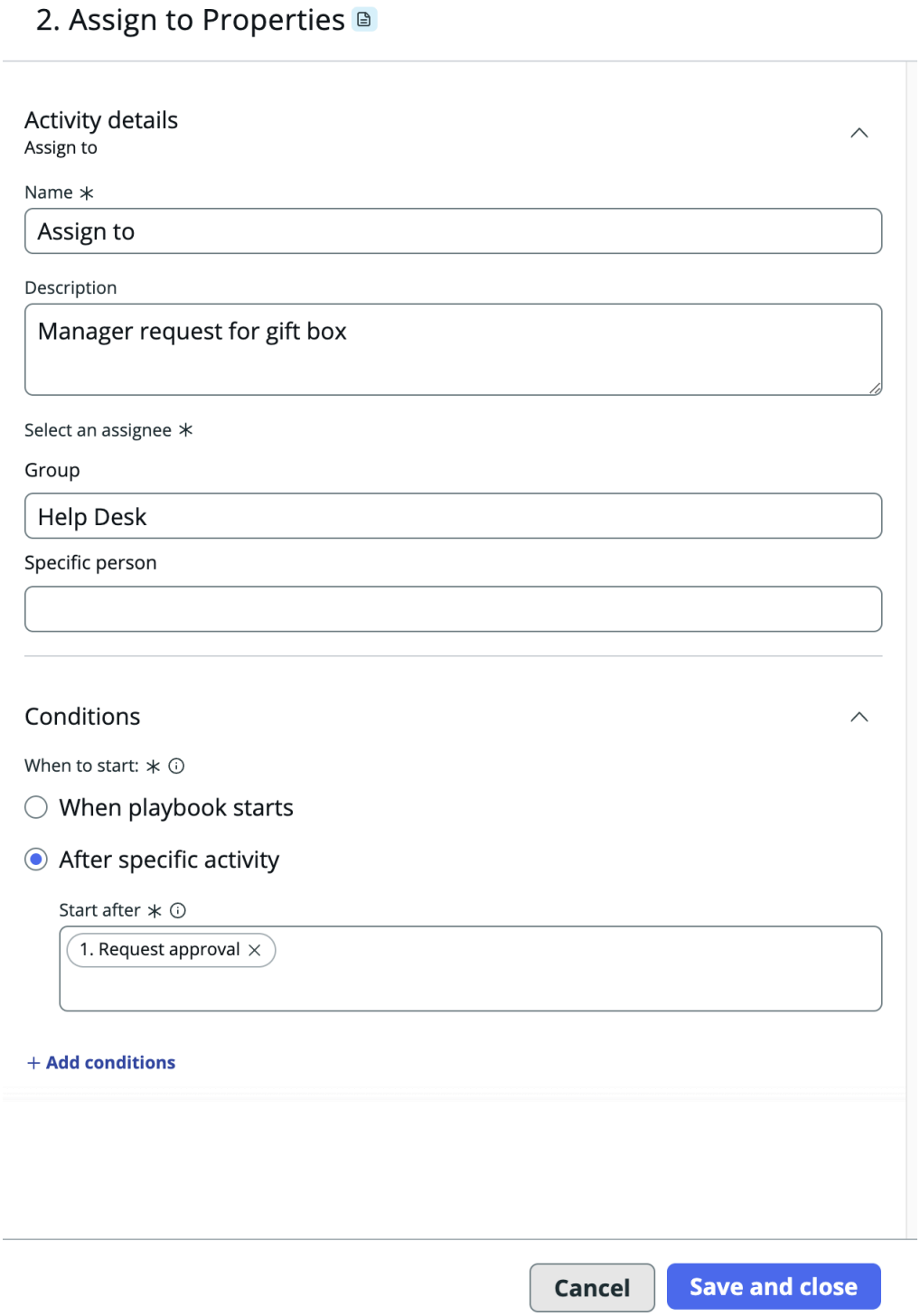
7. Enter the basic details for the activity.

General activity details

Field	Description
Label	Unique, user-facing name for your activity, which appears to agents and fulfillers while the playbook is running.
Description	Optional details about what the activity accomplishes.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Activity details panel



8. If needed, specify the approver or assignee in the **Select an approver** or **Select an assignee** fields.
   Approvers and assignees are the person or people who must approve the task, or who the task gets assigned to. The options are:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- **Group**: Enter one or more user groups in the field that appears. For example, `App engine Admins`.
- **Specific person**: Enter the person's name as it appears on the ServiceNow AI Platform. If you entered a group, you can select only from users in the specified group.
- **Requester's manager**: (Request approval activities only) Automatically routes the approval to the manager of the person who's making the request.

9. Enter details for specific types of activities.

a. For Request approval activities only, specify the **Approval type**.
Approval type defines whether a single approver is needed or if everyone must approve the request. The options are:

- **Anyone approves**: Only one of the specified approvers must approve the request.
- **All approve**: Every one of the specified approvers must approve the request.

  If you select this option and specified a group, all members of the group must approve the request.

b. For Create task activities, specify the **Priority**.
The options are:

- **Critical**
- **High**
- **Moderate**
- **Low**
- **Planning**

c. For Send an email activities, specify the recipients and contents of the email.

- Enter the name (for ServiceNow AI Platform users) or email address for people who should receive the email in the **To** and **Cc** fields.

  Select the **Requester** option for the **To** or **Cc** fields to include the person who made the request in the email's recipients.

- Enter the email's subject in the **Subject** field.
- Enter the body of the email in the textbox. Select the additional toolbar buttons icon  to display more formatting options. For example, you can add images, tables, and links to the email.

d. For Update submission activities, select the fields that the activity should update on the submitted form's record.
Choose one or more of the following fields to be automatically updated and specify the field value:

- **Short description**: Then enter a value for the updated short description.
- **Description**: Then enter a value for the updated description.
- **State**: Then select the updated record state. For example, **Open**.
- **Priority**: Then select the updated value. For example, **4 - Low**.

For example, you can add an **Update submission** activity that automatically closes a record if a previous approval activity is declined.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

10. Select at what point the activity should start running in the **When to start** field of the Conditions section.
    The options are:
    - **When *playbook* starts**: The approval request runs when the trigger occurs, which is the form's record being created or updated.
    - **After specific activity**: The approval request runs after one or more activities run, which you must specify in the **Starts after** field. You can select only activities that occur sequentially before the activity you're working on.

      Note: This option isn't available for the first activity in a playbook.

11. **Optional:** Define any extra conditions that must be met for the activity to happen by selecting **+ Add conditions**.
    Use the condition builder that appears to specify any other things that must happen for the activity to run. For more information, see Create a condition statement using the condition builder.

    You can use the answer from the associated form to power the activity. For example, you can specify that if someone answers Yes to the question "Do you eat meat?" the Assign to activity is assigned to the Meat department to fulfill the gift box. To use the response from a form as a condition, do the following:

    a. Choose **Questions** in the **Field** field.
    b. Select the form's question to use as the condition from the **Question** field.
    c. Specify the condition **Operator** and the answer's **Value**.
    d. Select **Apply** when you're done defining the condition.
12. Select the **Save and close** button to finish setting up your activity.

What to do next
Continue adding activities and decisions, as well as connectors and parallel processes if needed, to finish creating your playbook. For more information on decisions, see Add a decision to an app's playbook in Creator Studio.

Add a decision to an app's playbook in Creator Studio

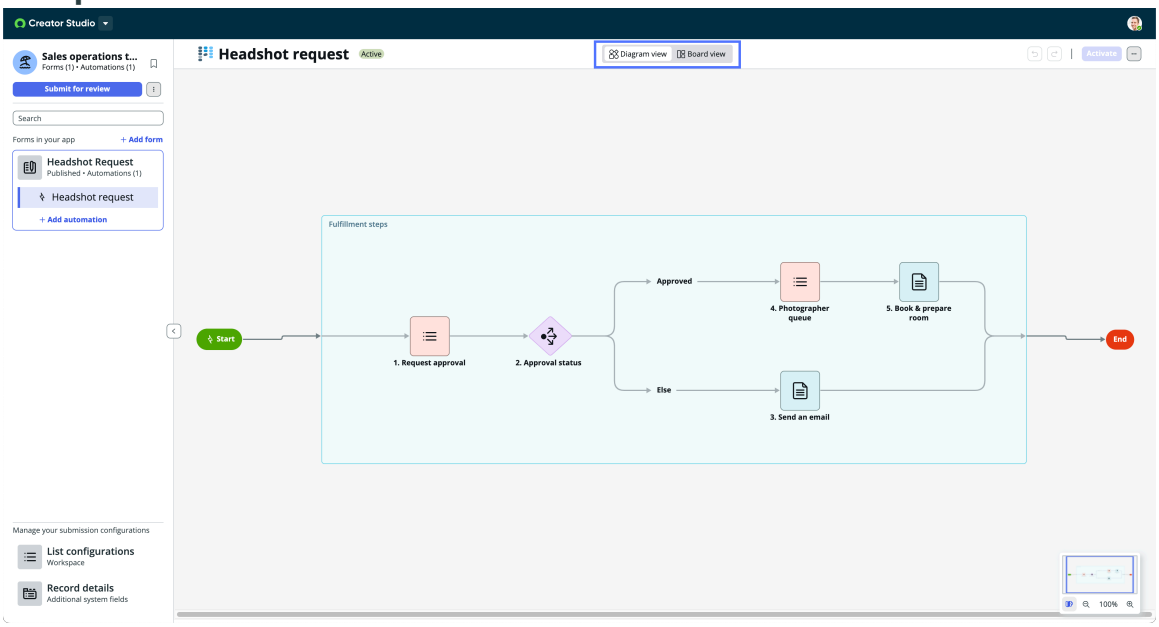
Add decisions, which are if/then conditions, to define branches, or different paths of an automation's playbook in Creator Studio.

Before you begin
To add decisions to a playbook, you must be given permission to work on the app.

About this task
Each decision should have at least two branches, with each branch representing a possible outcome for the decision. The second branch can be the ELSE branch, which is required. The ELSE branch will be followed when all other conditions on other branches are false. Think of the else branch as the catch-all case for the decision.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example decision with branches

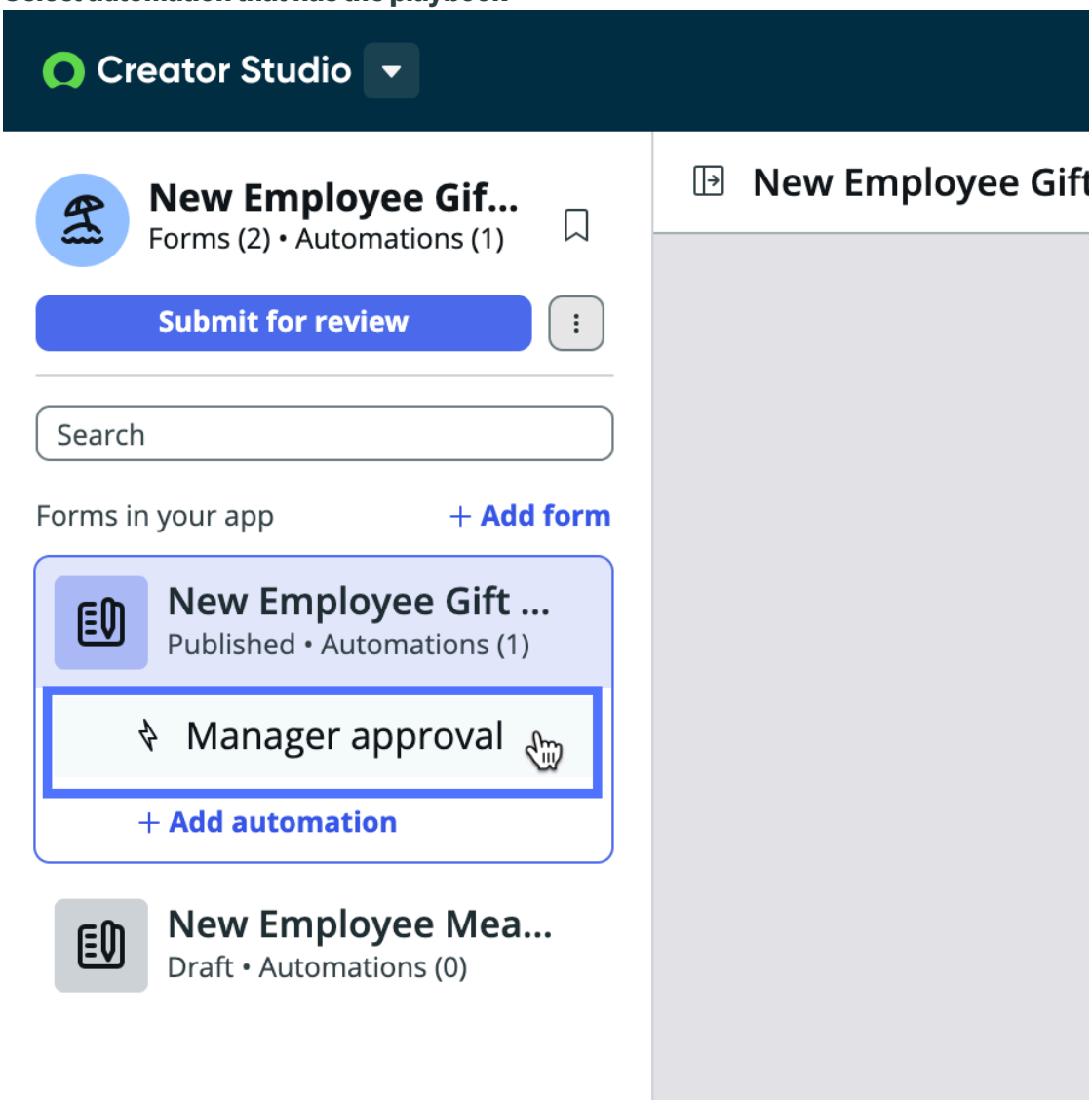


Procedure

1. Go to **All** > **App Engine** > **Creator Studio** to see all the apps on the Creator Studio home page.
2. Open the app that contains the playbook you want to add a decision to.
3. Select the form that contains the playbook that you want to add a decision to.
4. Select the automation that contains the playbook that you want to add a decision to.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Select automation that has the playbook



5. Select the add icon on the connector here you want to add a decision and choose the diamond-shaped **Add a decision** icon in the menu that pops up.

   Note: You must be in the Diagram view to add a decision.

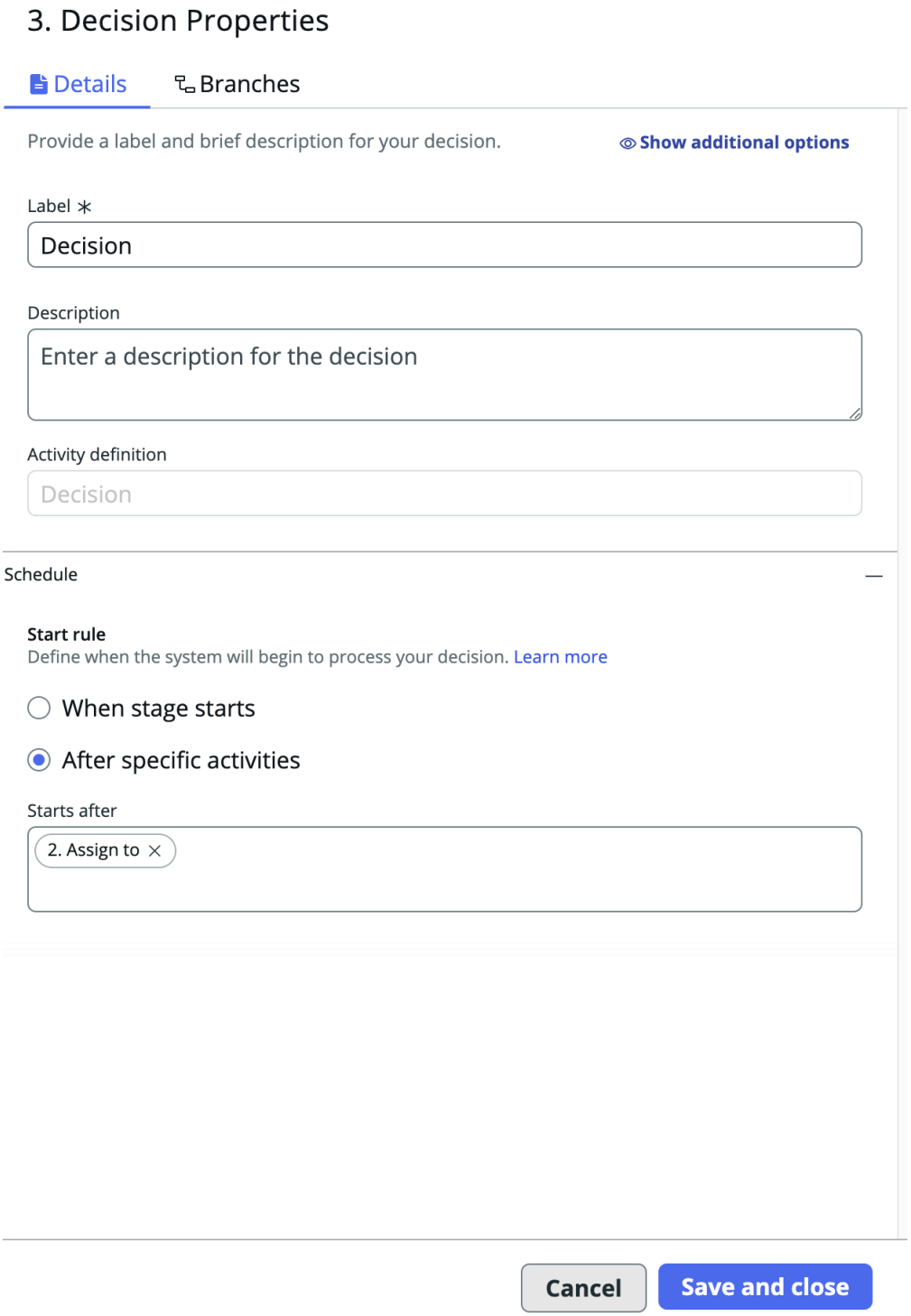
6. Specify the decision's basic attributes on the **Details** tab of the Decision properties panel that appears.

Decision basic options

Field	Description
Playbook name	Unique, user-facing name for the decisions, which appears to agents and fulfillers while the playbook is running.
Description	Optional details about what the decision accomplishes.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Playbook decision properties



7. Define the decision's schedule.

Decision schedule options

Field	Description
Start rule	Choose when you want your activity to start running. The options are:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
	○ **When stage starts**: The decision starts running as soon as the stage starts running, which is when your playbook is triggered. ○ **After specific activities**: The decision starts running after the specified activities have finished running.
Starts after	Select the activity that must finish running for the decision to happen. This field is editable only if you selected to start the rule **After specific activities**. You can choose only activities that happen before this decision in the playbook.

8. **Optional:** Specify more scheduling conditions for the decision, such as whether there's a delay, by selecting **Show additional options**.

Decision schedule additional options

Field	Description
Display order	Order in which this activity occurs during a playbook run.
Start with delay	Toggle to specify that the ServiceNow AI Platform waits for a duration of time before running the decision after the start rule is met. For more information on how to specify the delay duration, see Start with delay input properties.
Restart rules	What the decision does when a playbook is restarted. The options are: ○ **Skip on restart**: Skip this decision when the playbook run is due to a restart. ○ **Run always**: Always run this decision, including first runs. ○ **Skip on first run**: Skip this decision during the first run. For more information, see Restart a playbook.

9. Create the conditions for each branch, or possible outcome for the decision on the **Branches** tab of the Decision properties panel.

   a. Enter a name for the branch in the **Branch label** field.

   b. Select the **Add condition** button and specify what conditions should be met for the branch to take effect.
   You can use the answer from the associated form to define the conditions for the decision branch. To do so, complete the following steps:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

i. Choose **Questions** in the **Field** field.

ii. Select the form's question to use as the condition from the **Question** field.

iii. Specify the condition **Operator** and the answer's **Value**.

iv. Select **Apply** when you're done defining the condition.

For details on working with conditions, see Create a condition statement using the condition builder.

c. **Optional:** Select the **Add new branch** button and add as many branches as needed.

d. If you add two or more branches, select how you want your decision activity to run the different branches.

Branch processing instructions

Option	Description
Process any & all branches that are true	The app processes all branches with conditions that are met.
Process only the first one that is true	The app processes only the first listed branch with conditions met. If you select this option, drag the branch that you want processed first to the top.

e. **Optional:** Adjust the order of branches as needed by dragging them into a new position.

10. Select **Save and close**.

Add a parallel process to an app's playbook in Creator Studio

Create parallel processes in Creator Studio to make things happen in your app at the same time.

Before you begin

To add a parallel process to a playbook, you must be given permission to work on the app.

About this task

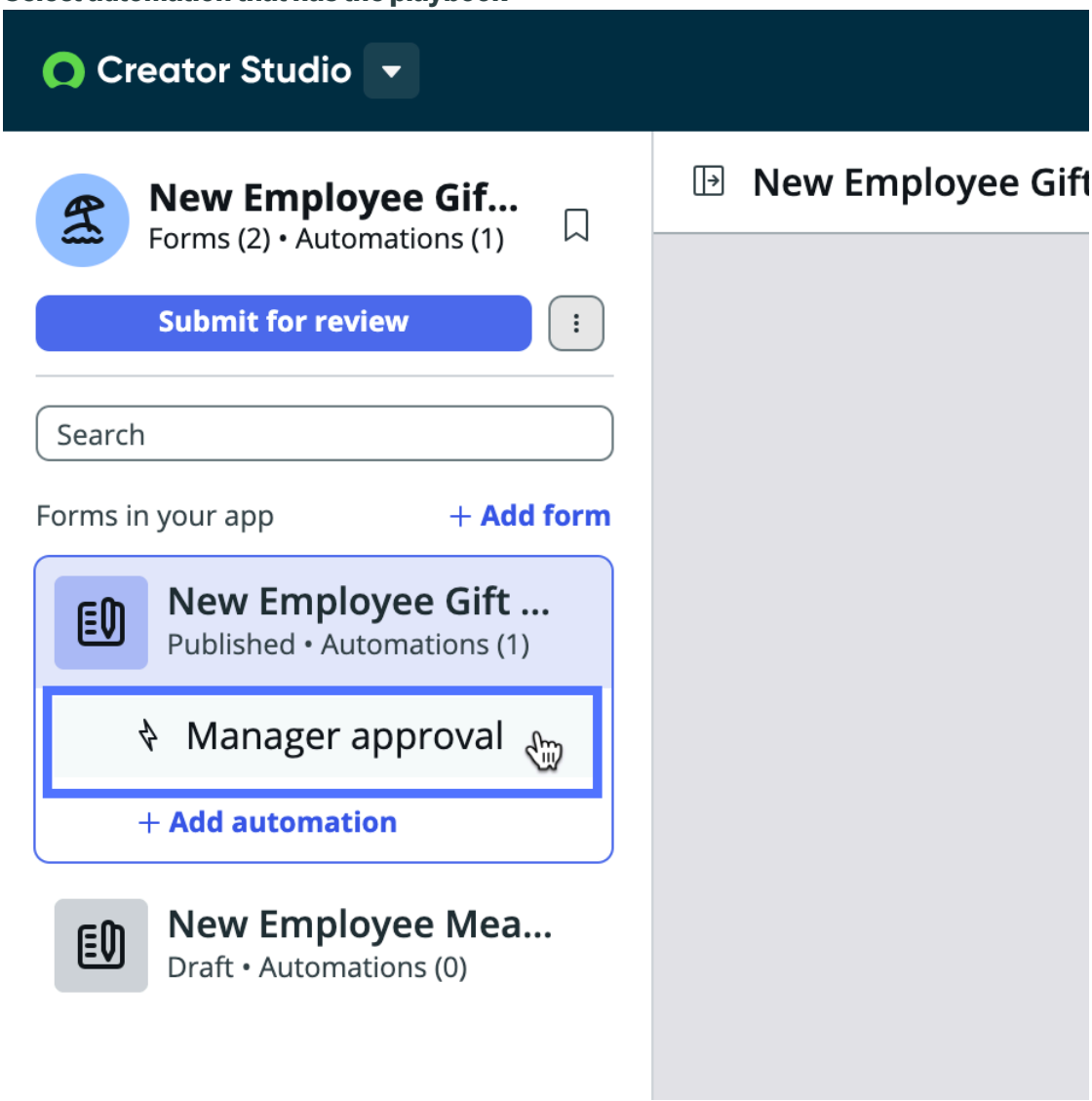
Activities on a process's parallel branches run at the same time as other specified parallel activities. Unlike decision activities, the activities and stages on a parallel can run on the same conditions as other branches.

Procedure

1. Go to **All** > **App Engine** > **Creator Studio** to see all the apps on the Creator Studio home page.
2. Open the app that contains the playbook you want to add a parallel process to.
3. Select the form that contains the playbook that you want to add a parallel process to.
4. Select the automation that contains the playbook that you want to add a parallel process to.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

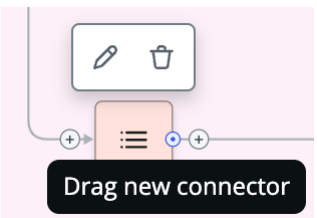
Select automation that has the playbook



5. Choose one of the following ways to add a parallel branch in the diagram view.

- Select the add icon and choose the **Add a parallel path** icon in the menu that pops up.
- Select the Drag new connecter dot icon for the parallel process's starting activity and drag it to the activity or decision that should be the end point of the parallel process.

Note: If you can't see the Drag new connector dot icon, you might need to hover over the activity to see it.



6. Add activities to the parallel branch as needed.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For more information, see Add activities to an app's playbook in Creator Studio.

7. If you want to have several activities happen in a parallel process as soon as the playbook is triggered, complete the following steps.
   a. Display the Board view.
   b. For each activity that should be happen in parallel when the playbook is triggered, select the activity's card and set the **When to start** field to **When playbook starts**.
   c. Select the **Save and close** button.

Edit the trigger for a playbook in Creator Studio

Define the trigger for a playbook in Creator Studio to specify what makes the playbook start running.

Before you begin

The trigger initiates a playbook to run on one of the app's forms. For example, when a user selects a specific answer to a question, the playbook begins to run. The form you select must already be configured for you to select it for the trigger.

To edit the trigger for a playbook, you must be given permission to work on the app.

About this task

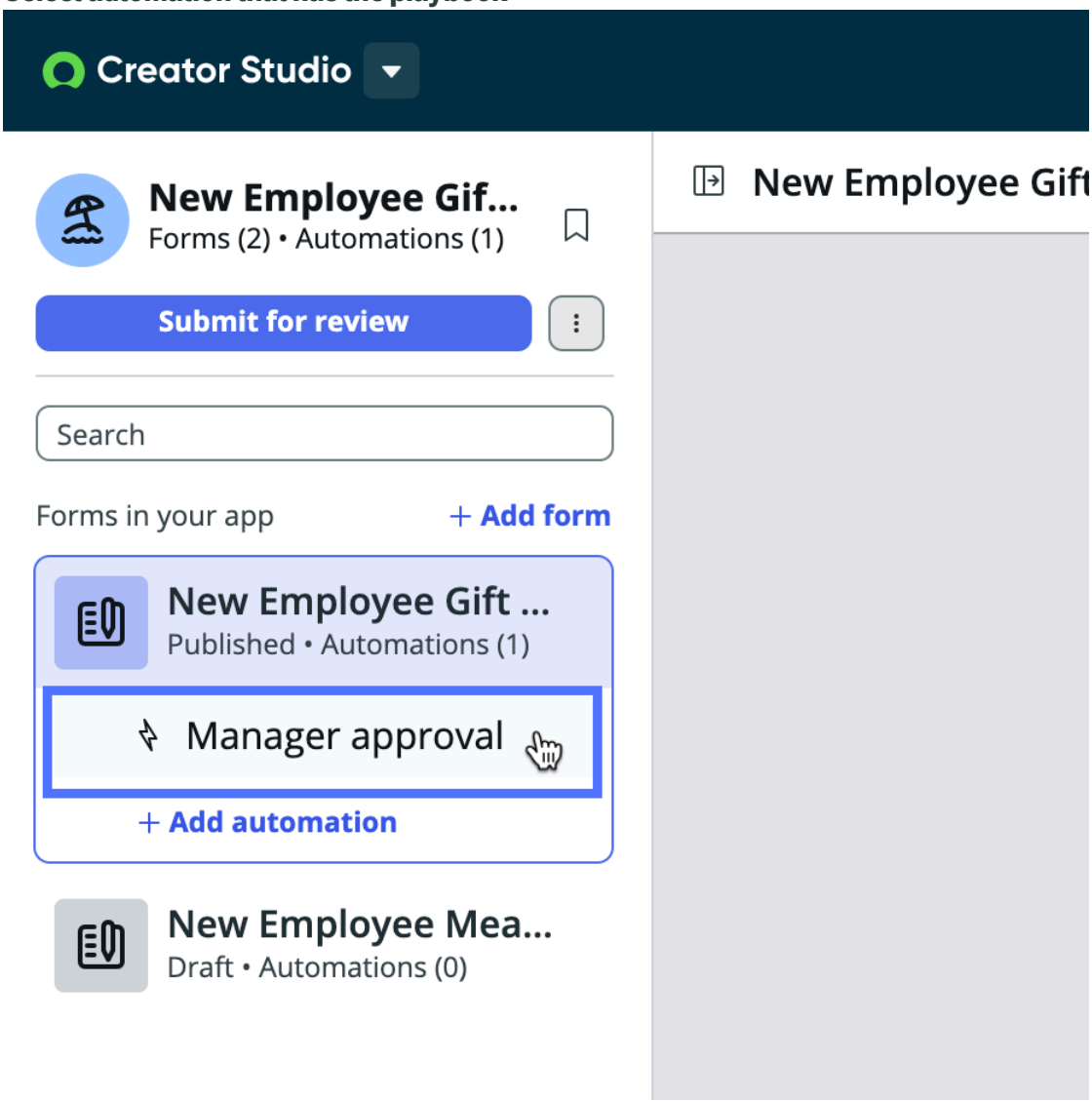
Note: You can't change a playbook's trigger (whether the form is submitted or updated to initiate the playbook) or how frequently it should run after you create the playbook. Instead, create a new playbook with a different trigger.

Procedure

1. Go to **All** > **App Engine** > **Creator Studio** to see all the apps on the Creator Studio home page.
2. Open the app that contains the playbook you want to add actions to.
3. Select the form that contains the playbook that you want to edit the trigger for.
4. Select the automation that contains the playbook that you want to edit the trigger for.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Select automation that has the playbook



5. Select the **Trigger** (Trigger).
   The Additional properties modal appears, where you can edit some of the trigger's settings.
6. Make any changes to the trigger's general attributes.

General playbook definition fields

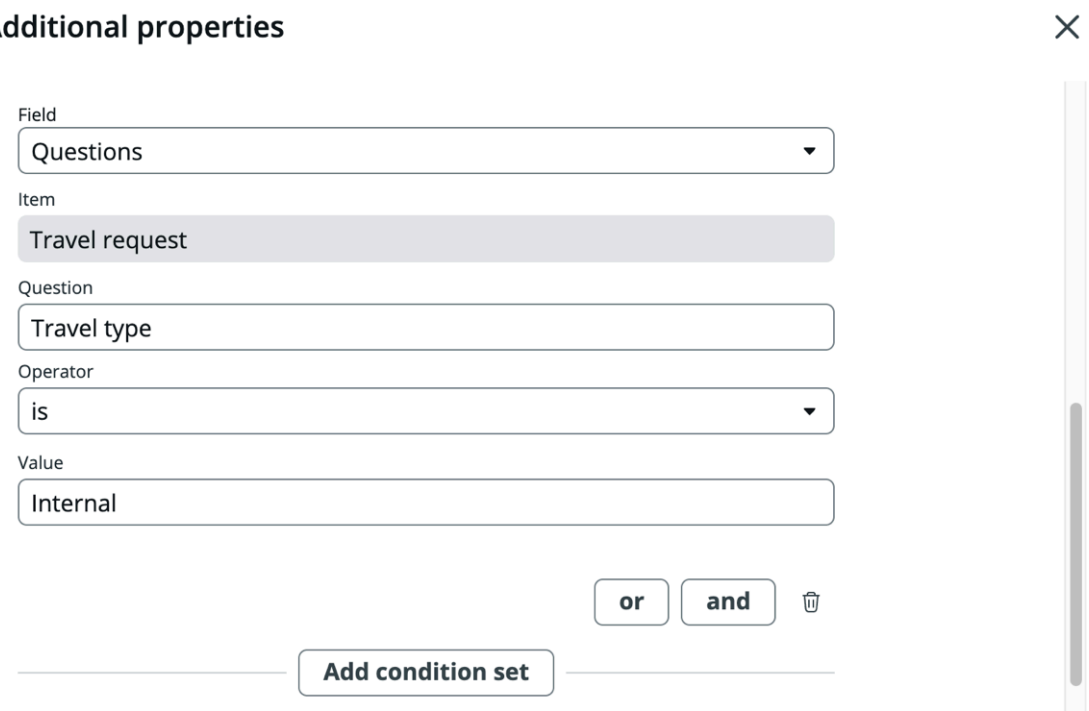
Field	Description
Playbook name	Descriptive name for the playbook you're editing.
Description	Brief explanation of what the playbook does, for example, the end goal for the record type.

7. If necessary, change the **Form** whose catalog item generates the record type that you want the playbook to run on.
8. Change the conditions that must be met for the playbook to begin running by selecting **Add condition set**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- If you want to trigger the playbook based on the value of a column in a table, select the **Field** that you want to be the trigger, as well as its condition **Operator** and the specific trigger **Value**. For example, when a **Start date** is **after** the **Date** needed.
- If you want to trigger the playbook based on the response from a form, select **Questions** as the trigger **Field**. Then select the question you want in the **Question** field, the condition **Operator** and the answer's **Value**.

Question answer as trigger for an automation



Add as many conditions as you need. For more information, see Create a condition statement using the condition builder.

9. Save your changes by selecting the **Done** button.

Edit a playbook in Creator Studio

Update a playbook to change its settings, or rearrange or remove tasks, to make it work for your app.

Before you begin

To edit a playbook, you must be given permission to work on the app.

About this task

Note:

Not all items that can be added to playbooks are supported in Creator Studio, for example, notifications.

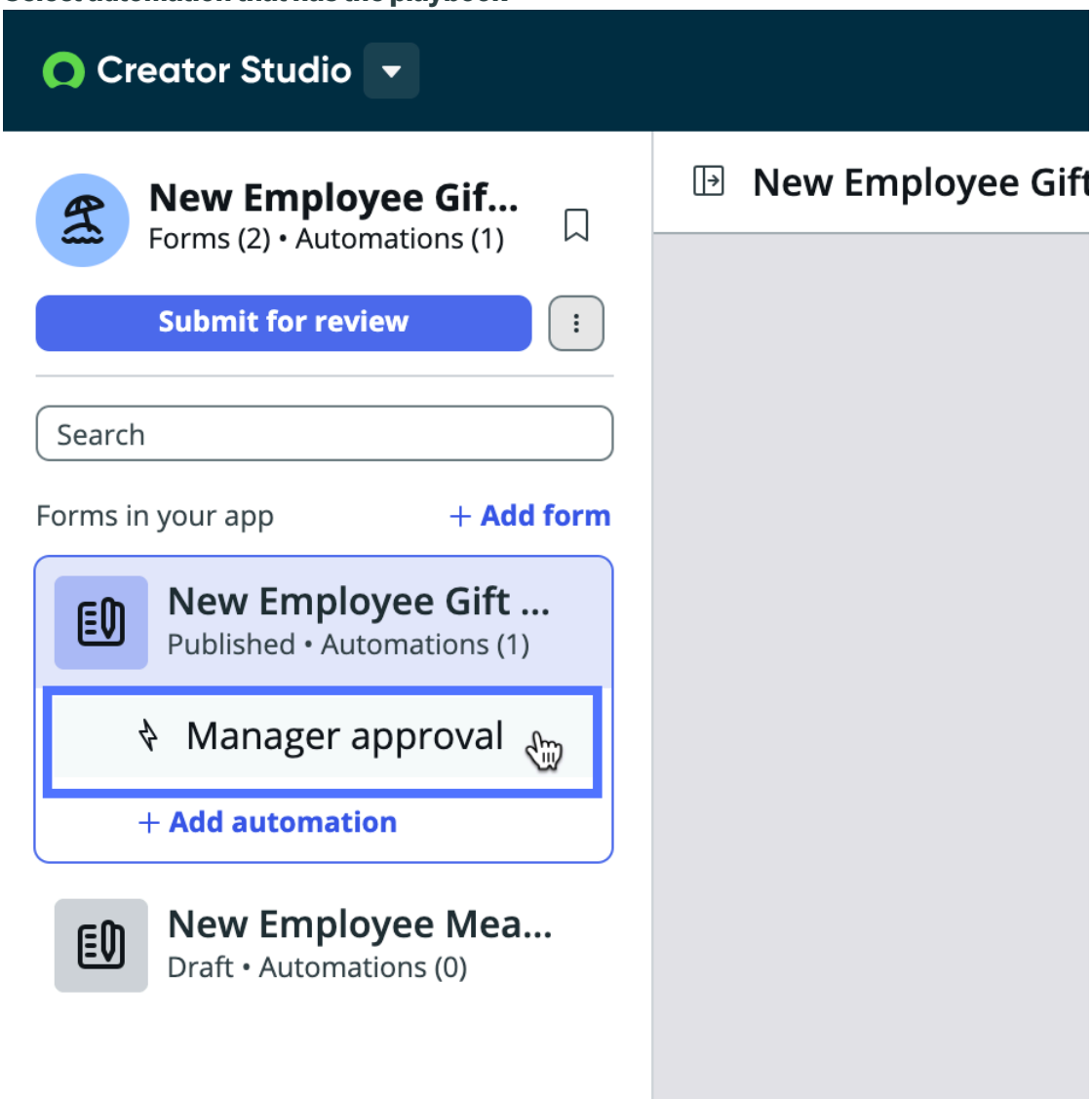
If you want to add complex, unsupported items to a playbook (such as optional activities, multiple stages, or data pills from previous activities), you must open and edit that playbook in Workflow Studio. After you edit a playbook in Workflow Studio, you can't edit it in Creator Studio. However, you can still work on the rest of the app in Creator Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

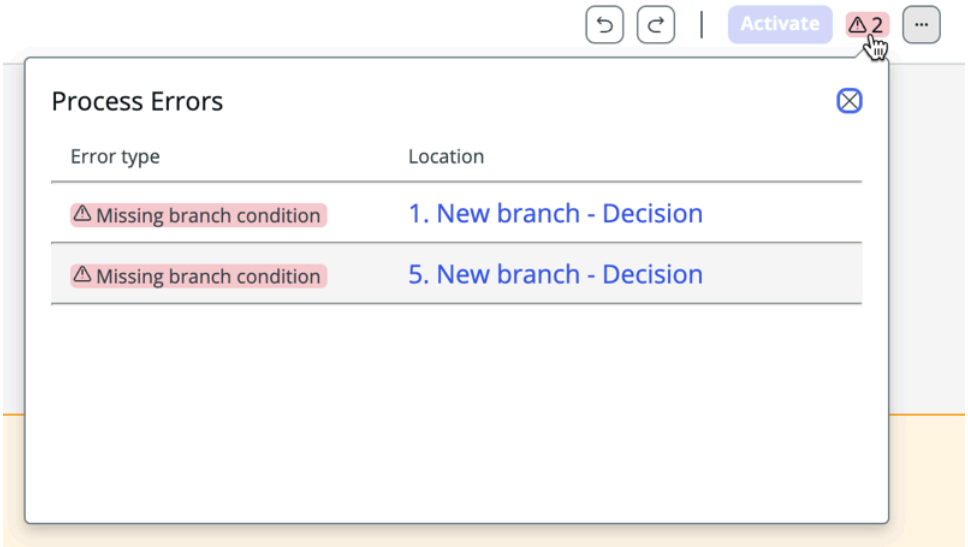
1. Go to **All** > **App Engine** > **Creator Studio** to see all the apps on the Creator Studio home page.
2. Open the app that contains the playbook you want to edit.
3. Select the form that contains the playbook that you want to edit.
4. Select the automation that contains the playbook that you want to edit.

Select automation that has the playbook



5. Address any errors in the playbook's logic by selecting the error message icon.
   In the error tray that appears, you can open the properties to fix each error by selecting the **Location** link.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



6. Edit the playbook's settings.

   a. Select the more actions icon ··· and select **Properties**.

   b. Update any of the settings.

   > **Note:** You can't change a playbook's trigger (whether the form is submitted or updated to initiate the playbook) or how frequently it should run after you create the playbook. Instead, create a new playbook with a different trigger.

   c. Select the **Done** button.

7. Rearrange the order of tasks as needed.

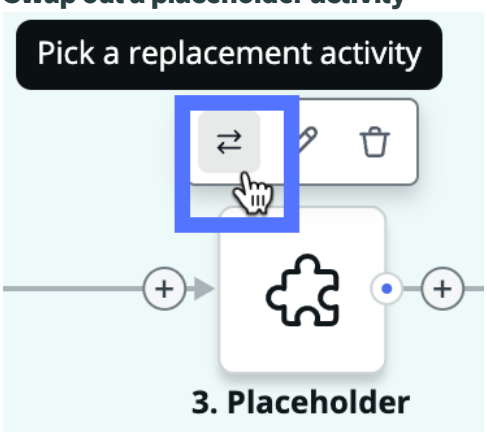
   a. Display the **Board view** for the playbook.

   b. Drag the cards for each activity into the order that you want.
   Creator Studio automatically saves the changes.

8. Swap a placeholder activity if you want to replace it with another activity type.

   a. Hover over the placeholder activity and select the replace activity icon ⇄ to directly open the activity picker.

Swap out a placeholder activity



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

b. Select the new activity from the activity picker.

c. Optional: Update the **Label** and **Description** as needed, or any of the other properties of the activity.
For more information, see Add activities to an app's playbook in Creator Studio.

9. Delete any unnecessary activities by hovering over the activity and selecting the delete icon .

10. Make any advanced edits in Workflow Studio by selecting the more actions icon and selecting **Open in Workflow Studio**.

Activate a playbook in Creator Studio

Activating a playbook means that it will run when its related form is created or updated on your non-production, development instance. However, the app must still be deployed to production.

Before you begin

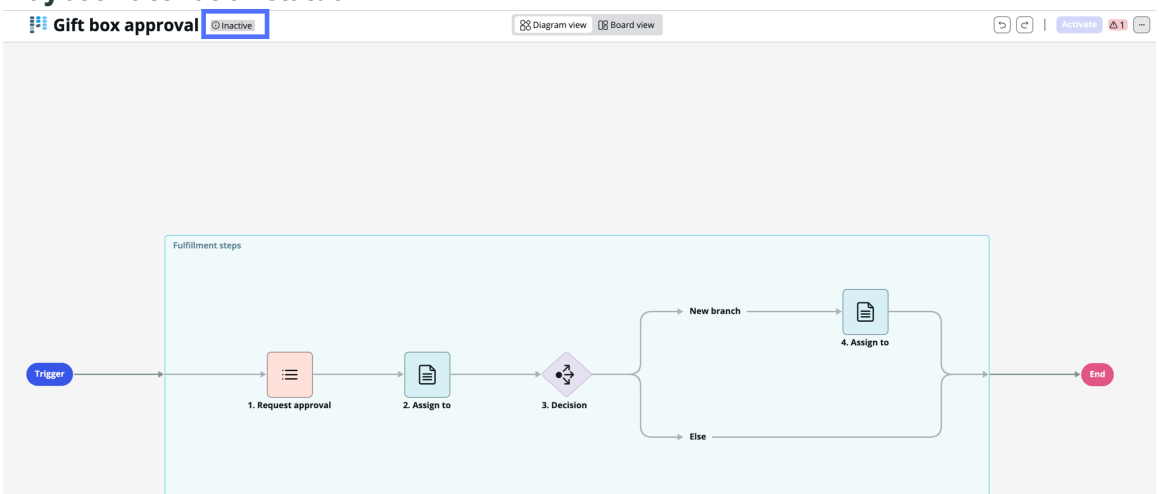
You must resolve any errors in the playbook before you can activate it.

To activate a playbook, you must be given permission to work on the app.

About this task

You can check the activation status of a playbook at any time in the app header.

Playbook activation status



If you don't activate a playbook and its app is deployed to production, the automation won't run on the applicable records. However, your App Engine admin can activate the deployed playbook for you.

Procedure

1. Go to **All** > **App Engine** > **Creator Studio** to see all the apps on the Creator Studio home page.

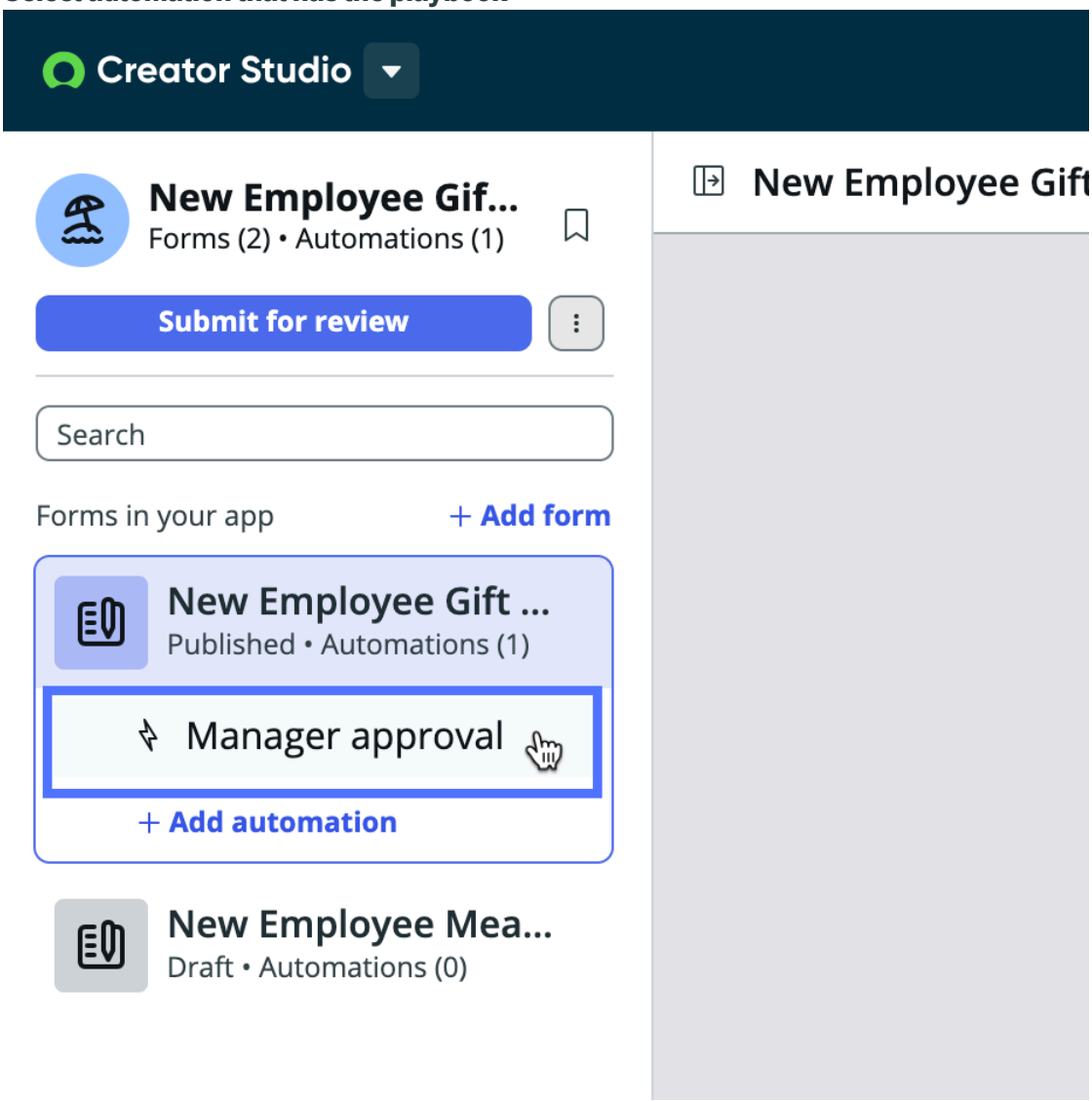
2. Open the app that contains the playbook you want to activate.

3. Select the form that contains the playbook that you want to activate.

4. Select the automation that contains the playbook that you want to activate.

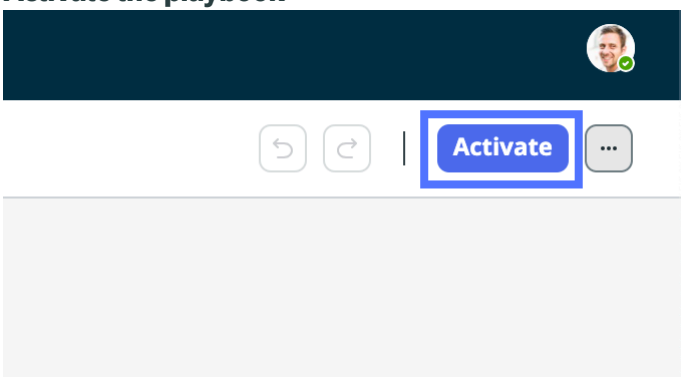
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Select automation that has the playbook



5. Select the **Activate** button.

Activate the playbook



Result

The status displayed in the app header updates from **Inactive** to **Saving** to **Active**, informing you of when changes are made.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The playbook is ready to be deployed with your app. Once the app is deployed, the form is available in the associated service catalog. If the form's app hasn't been deployed, you need to deploy it.

Delete a playbook in Creator Studio

Delete a playbook to remove it from the app completely.

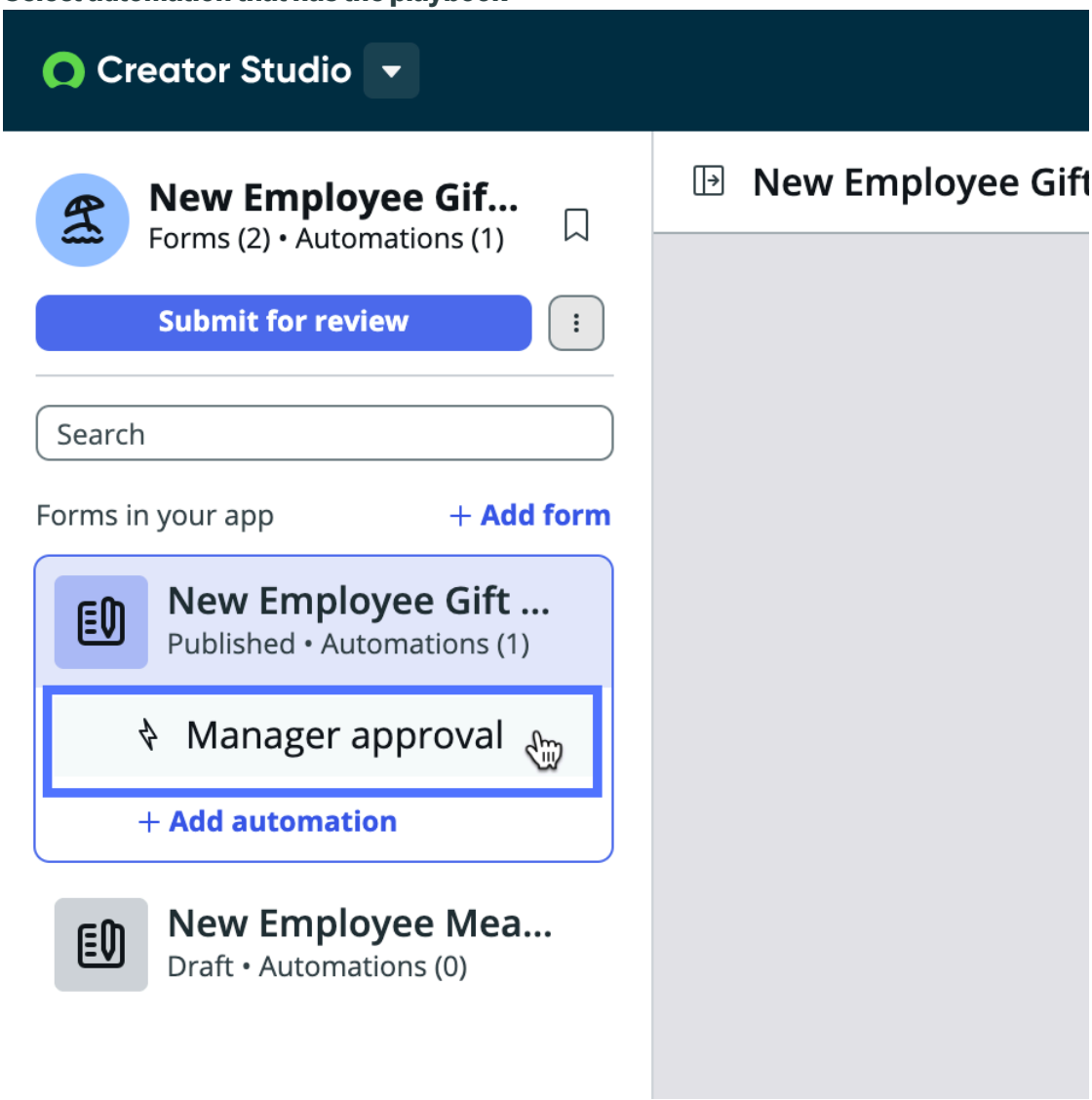
Before you begin

To delete a playbook, you must be given permission to work on the app.

Procedure

1. Go to **All** > **App Engine** > **Creator Studio** to see all the apps on the Creator Studio home page.
2. Open the app that contains the playbook you want to delete.
3. Select the form that contains the playbook that you want to activate.
4. Select the automation that contains the playbook that you want to delete.

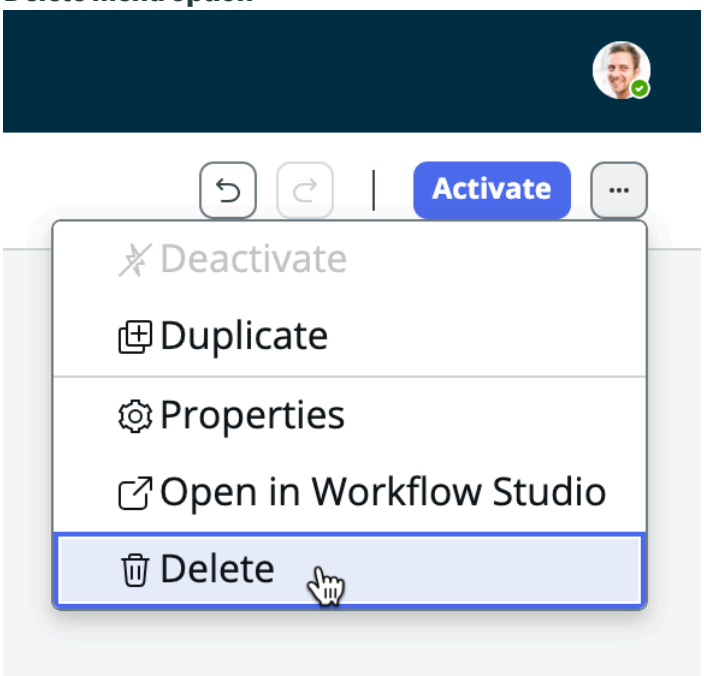
Select automation that has the playbook



5. Select the more actions icon ⋯.
6. Select **Delete**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Delete menu option



7. On the confirmation modal, select the **Delete** button.

Configuring workspace lists and records for form submissions

So, you've built your app, and people are submitting requests through it, that's great! However, you might be wondering where those requests go after they're submitted. They live in lists in a workspace where fulfillers can work on submitted forms, which you can customize in Creator Studio to suit your needs.

A little more about workspaces

The Request App Workspace holds form submissions that come in from each of your apps. Fulfillers (remember, these are the people who work on requests) can access the workspace through the ServiceNow AI Platform and see submissions for all the apps they're assigned to.

There is only one Request App Workspace per instance. Each app that you build in Creator Studio has its own category in the Request App Workspace, with as many lists as you configure, regardless of how many forms or automations it has.

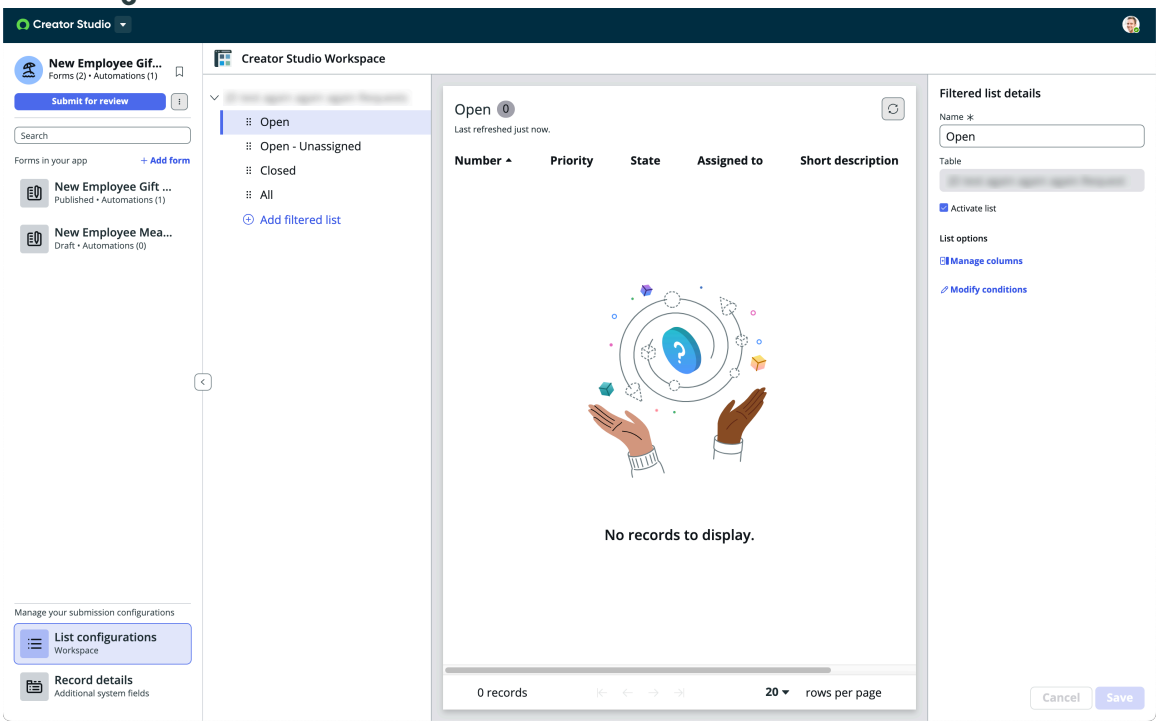
As the app owner, you're responsible for the look and feel of the workspace lists that the app's fulfillers use. Thankfully, the workspace should be ready to use out-of-the-box, but you can also customize it on the **List configurations** section of your app's navigation panel in Creator Studio.

When you open the **List configurations** tab, you'll see some filtered lists that appear in the default workspace. These filtered lists (**Open**, **Open – Unassigned**, **Closed**, and **All**) are intended to give fulfillers a quick view of all the form submissions. You can add more filtered lists as needed, for example, you could add a filtered list for each of the app's forms or for canceled requests.

If it would help your fulfillers to have additional categories, you must ask your admin to add them using the List Category [sys_ux_list_category] table on the ServiceNow AI Platform.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

List configurations section



Creating lists to support multiple forms

When fulfillers open the workspace to view an app's form submissions, the default "Open" list displays all open requests made through the app. However, if you have multiple forms in your app, fulfillers might find it easier to have separate "Open" lists for each form in the app. In that case, you should create a different list for each separate form and name each of them after the forms. Find out how in Add a filtered list to a workspace in Creator Studio.

Fulfillers can use the "My Lists" feature in Workspace to create custom lists that fit their needs. Tell them how to by checking out ServiceNow AI Platform® list administration.

How fulfillers access submitted forms

Fulfillers access the default workspace on the ServiceNow AI Platform by going to **All** > **App Engine** > **Request App Workspace**. The Request App Workspace displays a category for each app with filtered lists of records, both default and any extra that you added. Fulfillers then select a record to view the details of the request and make updates as needed.

Note: You can't access the Request App Workspace as the owner of an app, though you can access it if you also have an admin or the applicable agent role. The app's agent role is <app scope>.agent, for example, x_snc_app_name.agent.

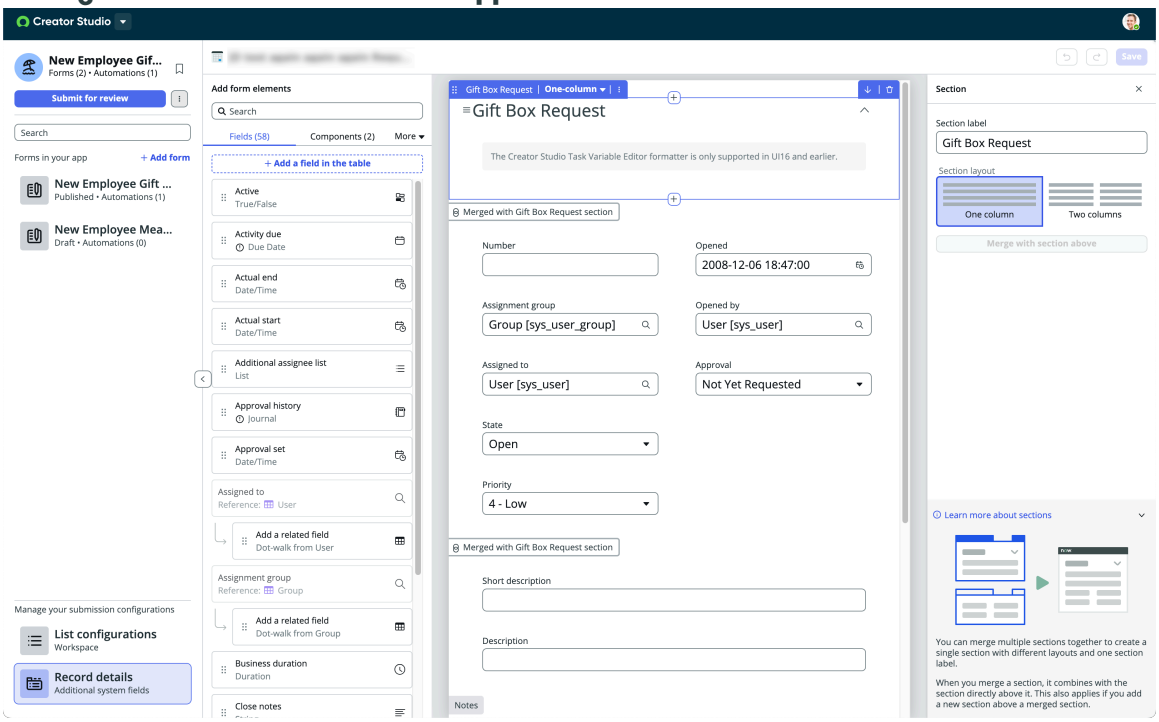
You can preview and interact with how records appear in the workspace category directly within Creator Studio by navigating to the **List categories** section of the navigation panel and selecting a record.

- Answers to the form's questions (which are catalog variables) appear on the **Details** tab of the submitted record. For more on form questions, see Working with forms in Creator Studio.
- Results of any playbooks appear on the **Automations** tab of the submitted record. For more on playbooks, see Working with automation in Creator Studio.

Edit how the record a form generates will appear by navigating to the **Record details** section of the navigation panel.

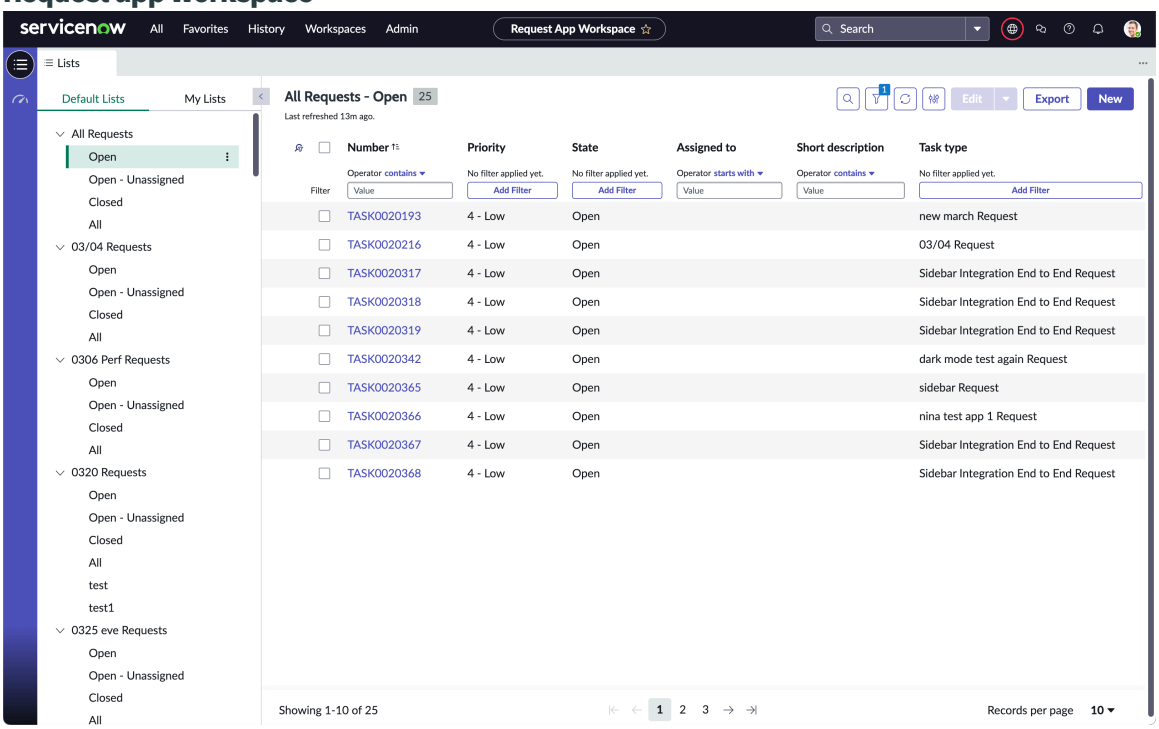
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Editing how a submitted record will appear



What fulfillers can see in the workspace lists depends on their permissions. For example, admins can see workspace categories for all apps built in Creator Studio. However, most users see only the workspace category for their particular app.

Request app workspace



When fulfillers view a record for an individual request in the workspace, the **Catalog tasks** tab displays the playbook that you configured in Creator Studio.

Fulfillers or admins may also want to monitor the app's performance as requests come in. This is where the **Dashboard overview** in the Request App Workspace is helpful! On the **Dashboard overview** tab, each app has its own analytics dashboard that shows applicable data, like how

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

many open requests you have or what tasks are related to the app. There's a lot of helpful information here. Find out more about working with data in Platform Analytics overview.

Add a filtered list to a workspace in Creator Studio

Create custom filtered lists in the app submission workspace to view records that meet specific conditions. For example, if your app has multiple forms, you can create a list for each form by filtering on the record type.

Before you begin

To add a list to a workspace configuration, you must be given permission to work on the app.

About this task

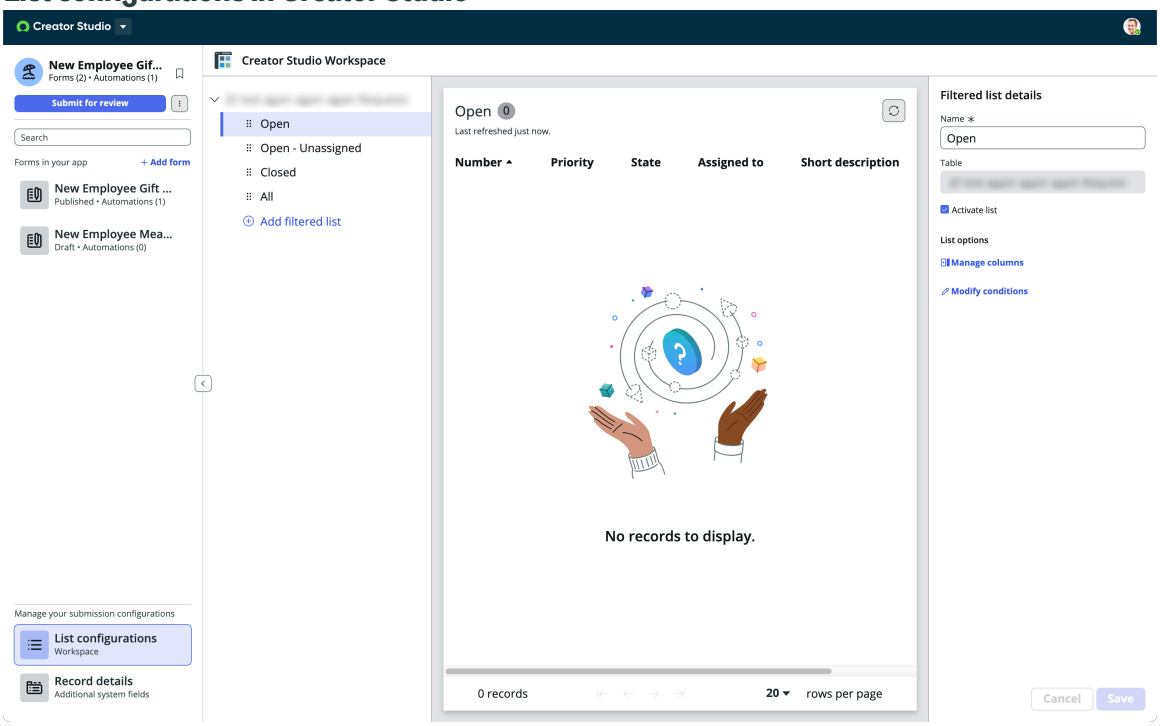
Note: You can create a filtered list to display open records assigned to a specific user, or all requests that have been cancelled.

Procedure

1. Go to **All** > **App Engine** > **Creator Studio** to see all the apps on the Creator Studio home page.
2. Open the app that contains the list configuration that you want to add a list to.
3. Select the **List configurations** section in the navigation panel.

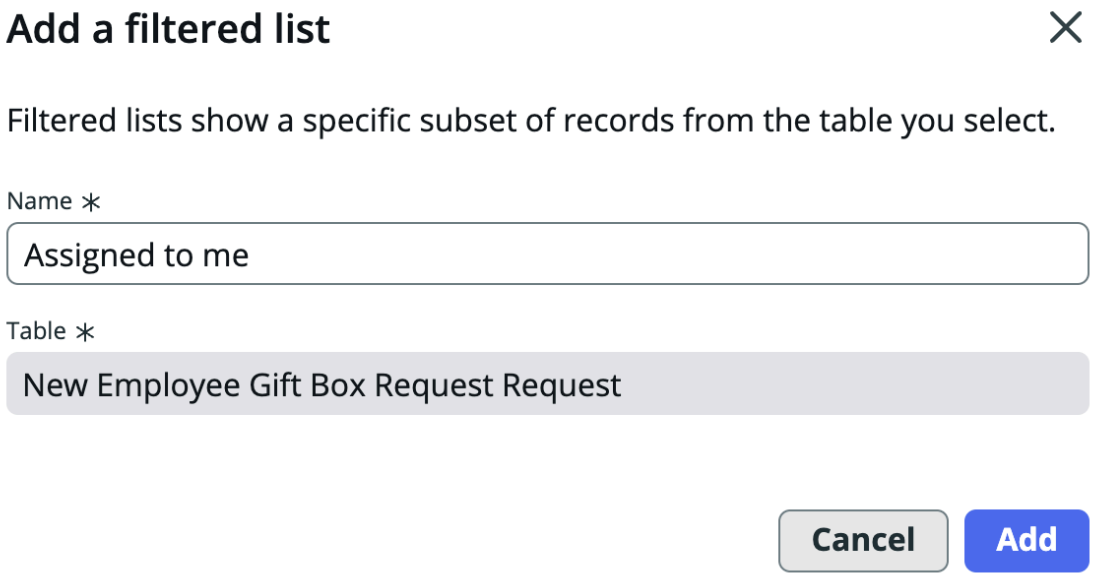
   If you tested submitting a record using the **Try it** button, the workspace displays the generated records that you can view.

List configurations in Creator Studio



4. Select **Add a filtered list**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



5. Enter a **Name** for the new list in the modal that appears.

   Choose a name to help users easily identify their category in the workspace, such as HR `timeoff requests`.

   Note: You can't change the table for the workspace, as it's linked to the app.

6. Select the **Add** button.
   The new list appears under the category, with its details appearing in the Filtered list details panel.
7. Select **Apply conditions** in the Filtered list details panel and use the condition builder to specify what types of records the list should contain, selecting **Apply** when you're done. For example, you can select **Urgency** as the field and **High** as the value to have the list show only urgent records. For more information, see Create a condition statement using the condition builder.

   For example, the following conditions specify open and closed requests:

   - `Active is true` specifies only open records.
   - `Active is false` specifies only closed records.

8. **Optional:** Change which columns appear and the order they're displayed in by selecting **Manage columns**.
   a. Move columns from **Available columns** to **Selected columns** in the Manage columns dialog box.
   b. Drag to rearrange the columns.
   c. Select **Apply**.
9. Ensure that the **Activate list** option is selected.

Result
Your new list will appear in the Request App Workspace after your admin deploys it!

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Customize an app's workspace configuration in Creator Studio

Make changes to the app's workspace category where fulfillers review submitted requests from apps built in Creator Studio to adjust how it appears. For example, you can rearrange columns in its lists and customize how the record generated by the request app looks.

Before you begin

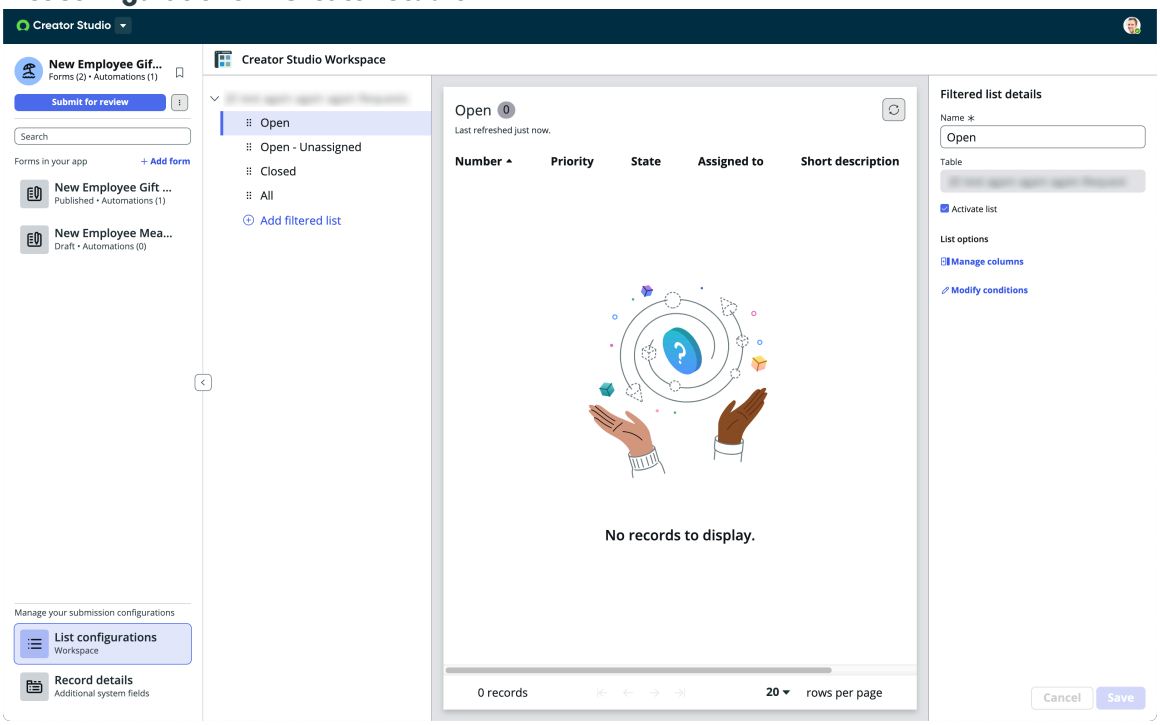
To customize how the app's workspace lists appear, you must be given permission to work on the app.

Procedure

1. Go to **All** > **App Engine** > **Creator Studio** to see all the apps on the Creator Studio home page.
2. Open the app that contains the workspace list configuration that you want to edit.
3. Select the **List configurations** section in the navigation panel.

   If you tested submitting a record using the **Try it** button, the workspace displays the generated records that you can view.

List configurations in Creator Studio



4. Make the edits that you need to customize the workspace list configuration.

Edits you can make to a workspace list configuration

Update	Process
Edit the name of a filtered list	**a.** Select the list in the list navigation panel. **b.** Update the **Name** of the list in the Filtered list details panel that appears. **c.** Select **Save**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Update	Process
Manage columns to change which columns appear and the order they're displayed in	**a.** Move columns from **Available columns** to **Selected columns** in the Manage columns dialog box. **b.** Drag to rearrange the columns. **c.** Save the update by selecting the **Apply** button.
Update the types of records that appear in a list	For example, you can select **Urgency** as the field and **High** as the value to have the list show only urgent records. For more information, see Create a condition statement using the condition builder. **a.** Select **Modify conditions** in the Filtered list details panel. **b.** Use the condition builder to update what types of records the list should contain. **c.** Save the update by selecting the **Apply** button.
Delete a list	**a.** Hover over the list you want to delete in the list navigation panel. **b.** Select the delete icon that appears ( ). **c.** Confirm the deletion in the modal that appears by selecting the **Delete** button.

5. Select the **Save** button in the Filtered list details panel to ensure all updates are saved.

Change the layout of an app's record in Creator Studio

Adjust how the submitted records that your app generates will look, such as the order in which fields appear.

Before you begin

To change the layout of an app's submitted record in the fulfiller workspace configuration, you must be given permission to work on the app.

About this task

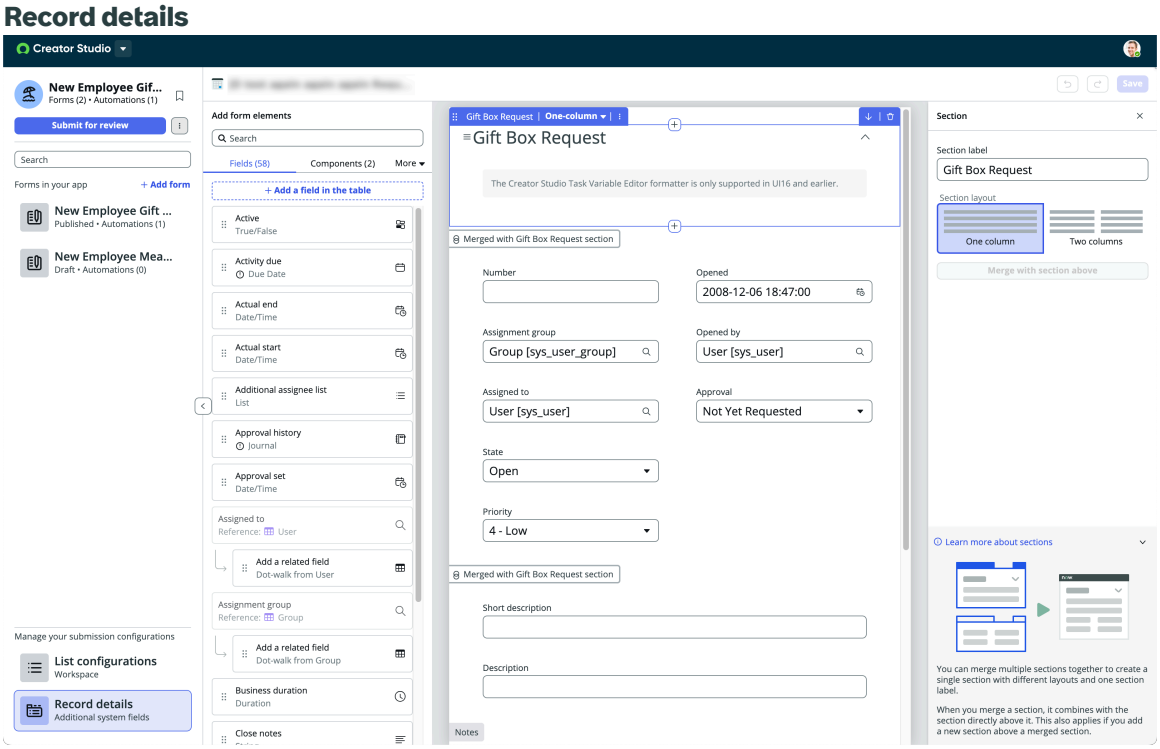
All columns of the app's associated table are available as fields on the record page, though you may want to adjust them. For example, you may want to move the **Priority** field to appear more visibly on the record so fulfillers can tell how important the request is. You could then build an automated playbook that runs when the priority changes to high.

Note: You can't edit UI policies or make more advanced edits to the form in Creator Studio. To make advanced edits, open the record in Table Builder. For more information, see Forms in Table Builder.

Procedure

1. Go to **All** > **App Engine** > **Creator Studio** to see all the apps on the Creator Studio home page.
2. Open the app that contains the workspace record details that you want to edit.
3. Select the **Record details** section in the navigation panel.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



4. Make any additional changes to how the record appears, such as moving the fields around. Check out Important Task table fields for a list of the most commonly used fields that you can work with and their definitions.
5. Select **Save**.

What to do next

After you're done customizing how the app's records will appear, you can check out how they'll look in the Request App Workspace while remaining in Creator Studio. For more information, see Preview how an app's records appear.

Using the Request App Workspace to fulfill requests

After users submit a request through an app built in Creator Studio, fulfillers work on the request's submitted record in the Request App Workspace.

The Request App Workspace holds app requests

The Request App Workspace holds form submissions that come in from each of your apps. Fulfillers (remember, these are the people who work on requests) can access the workspace through the ServiceNow AI Platform and see submissions for all the apps they're assigned to.

There is only one Request App Workspace per instance. Each app that you build in Creator Studio has its own category in the Request App Workspace, with as many lists as you configure, regardless of how many forms or automations it has.

Fulfillers access the default workspace on the ServiceNow AI Platform by going to **All** > **App Engine** > **Request App Workspace**. The Request App Workspace displays a category for each app with filtered lists of records, both default and any extra that you added. Fulfillers then select a record to view the details of the request and make updates as needed.

Note: To access the Request App Workspace, you must have a fulfiller role for the app built in Creator Studio. For example, x_snc_it_req.agent for an app with the scope x_snc_it_req.

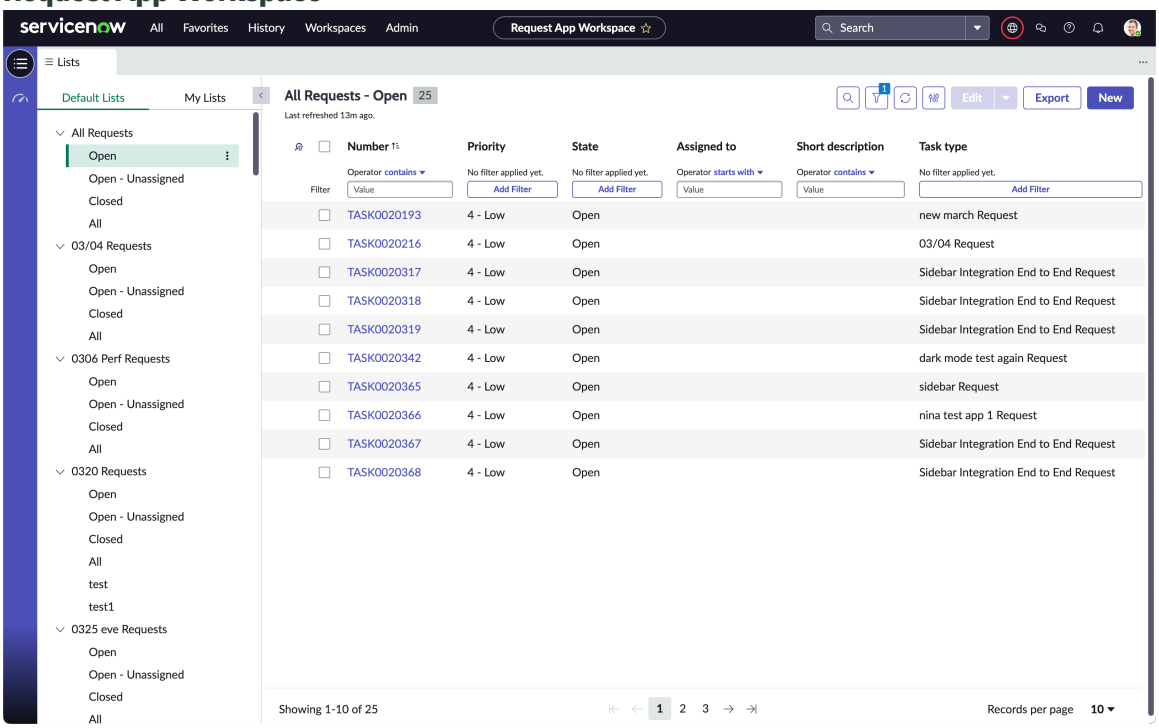
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Each app has its own section in the workspace, with lists to filter for submitted request records in various states.

Using lists to find tickets in various states

Each app's section contains lists, such as **Open** requests and any lists the admin or app creator configured. Fulfillers can use the "My Lists" feature in Workspace to create custom lists that fit their needs. Tell them how to by checking out ServiceNow AI Platform® list administration.

Request App Workspace



The Lists view contains two tabs: **Default Lists** and **My Lists**.

Using dashboards to analyze request fulfillment

The Dashboard overview view contains a dashboard for every app (that you have access to) that was made in Creator Studio.

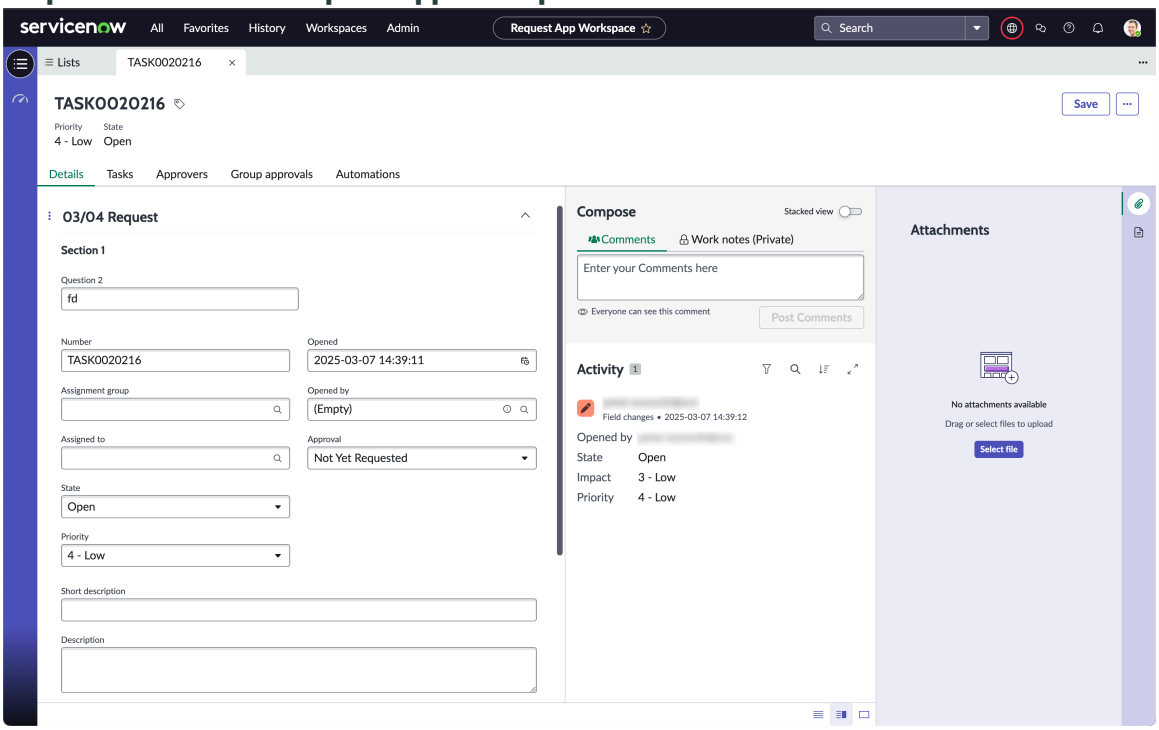
You can select an app's dashboard to see analysis of all active tasks, as well as the number of records in various states configured by your admin. For more information on working with analytics dashboards, see Dashboards in Platform Analytics.

Working on open requests in the workspace

After you find the subset of requests you need, such as open requests assigned to you, you can select the task record **Name** to open the record.After you find the subset of requests you need, such as open requests assigned to you, you can select the record **Name** to open the record.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Request record in the Request App Workspace



The submitted request record has various tabs that help you do your work.

Testing and previewing Creator Studio apps and records

Before you request deployment for your app, you can test out submitting a form and interact with any associated automation results on the record it creates.

Test one of your forms for an app

Use the **Try it** button on the **Request forms** tab to submit a request, which triggers any associated playbooks and creates a record that you can work with in Creator Studio.

About this task

Trying out a form differs from the preview that appears when you first view an app in Creator Studio in the following ways:

- Previewing doesn't submit the form, and thus doesn't generate a record from the form submission.
- Previewing lets you see how the form will look when accessed in different places, such as a portal, mobile, and Virtual Agent.
- You can preview a form at any time, but a form must be published for you to test it using the **Try it** button.
- Testing the form with the **Try it** button runs playbooks and generates a record.
  - Any automations (playbooks) associated with the form are run after you submit the form, and the results are available on the generated record.
  - You can then use that record, which opens in Creator Studio, to test how the record appears and functions for users in the Request App Workspace.

Before you begin

You must first create and publish (mark as ready) a form for you to test it using the **Try it** button. For more information, see Publish a form for your app in Creator Studio.

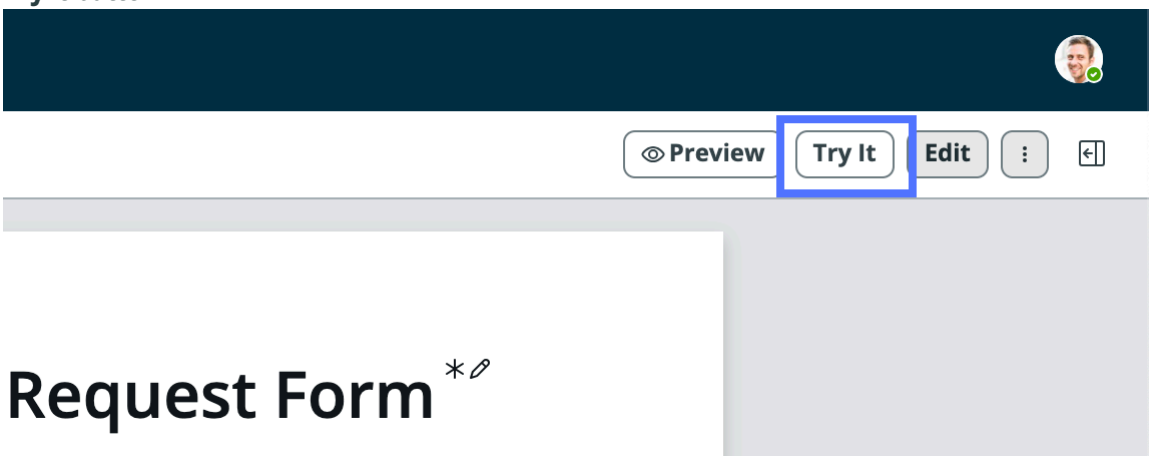
To test an app, you must be given permission to work on the app.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Head over to **All** > **App Engine** > **Creator Studio**.
   All the apps built in Creator Studio appear on the home page.
2. Open the application that contains the form you want to try out.
3. Select the **Try it** button.

Try it button



   The Try form that appears tells you which user will receive your request. For example, the administrator.
4. Fill in the fields for each question on the form.
5. Select the **Submit** button when you're done.

Result

After you submit your responses, the ServiceNow AI Platform runs any playbooks associated with the form.

The record that your submitted form creates appears in Creator Studio. You can view the results of the playbooks and interact with the record to see how it appears in the Request App Workspace.

- Answers to the form's questions (which are catalog variables) appear on the **Details** tab of the submitted record. For more on form questions, see Working with forms in Creator Studio.
- Results of any playbooks appear on the **Automations** tab of the submitted record. For more on playbooks, see Working with automation in Creator Studio.

For details, see Preview how an app's records appear.

Preview how an app's records appear

You can preview and interact with the submitted records generated by your app directly in Creator Studio, which mimics what the fullfiller sees in the workspace category for the app.

Before you begin

You can customize how the record will appear before testing it in a preview. For more information, see Change the layout of an app's record in Creator Studio.

You can create a submitted record to preview by testing the form with the **Try it** button. Testing the form records your answers, runs any associated playbooks, and generates a new record with the results of the automation. For more information, see Test one of your forms for an app.

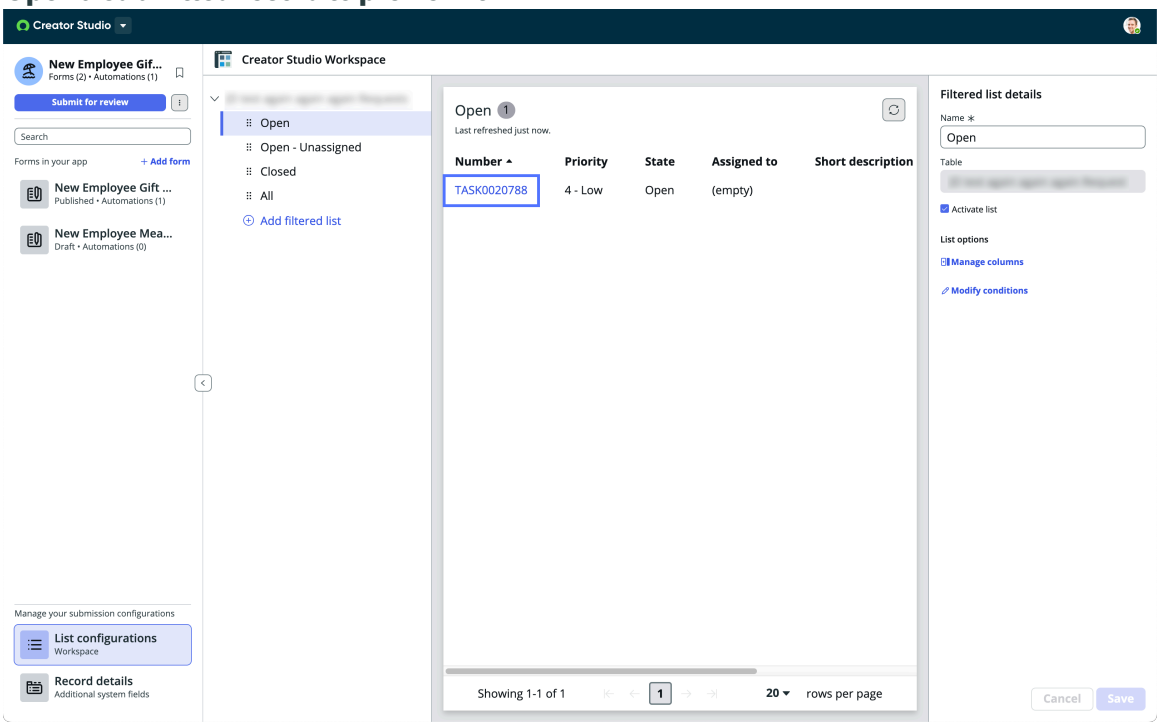
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

To preview how an app's submitted records will appear in the workspace configuration, you must be given permission to work on the app.

Procedure

1. Go to **All** > **App Engine** > **Creator Studio** to see all the apps on the Creator Studio home page.
2. Open the app that contains the workspace configuration that you want to edit.
3. Select the **List configurations** section in the application header.
4. Select a filtered list that contains a record, such as the *Open* records list.
5. Preview a record by selecting it in the canvas.

Open a submitted record to preview it



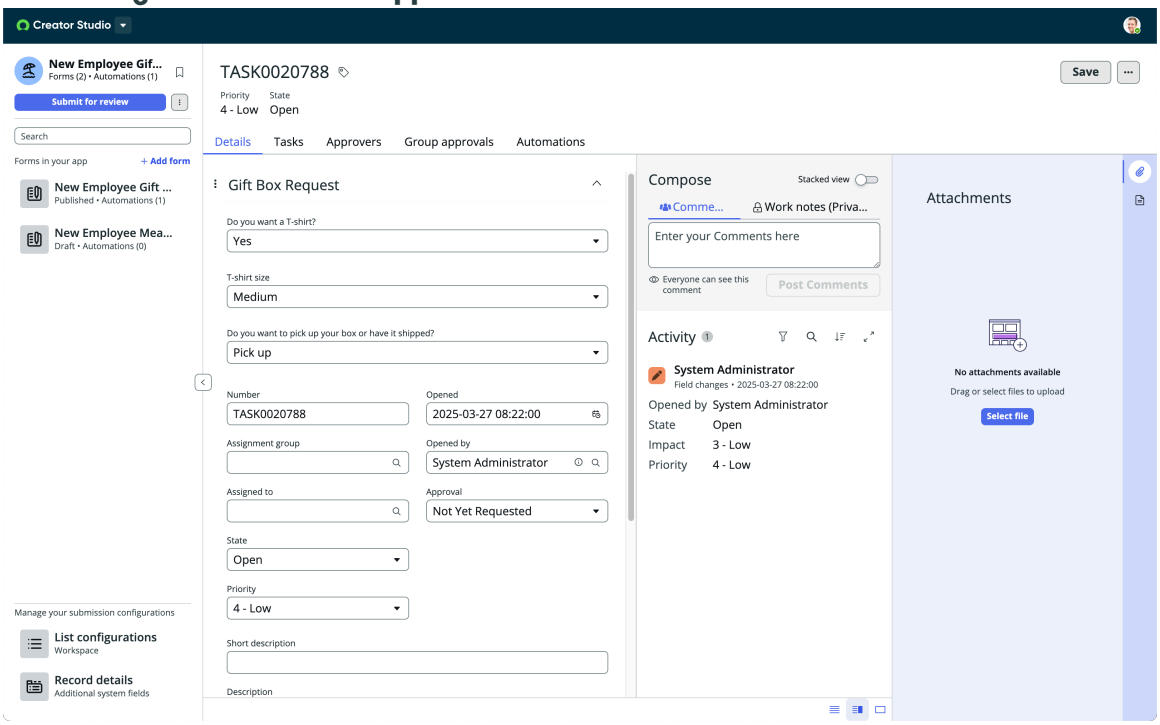
The record opens in Creator Studio.

6. Play with the record to see how its parts will appear.
   For example, you can select the record's **Approvers** tab to view how its approvers list will appear.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

If you're previewing a record that was generated using the **Try it** button, the form's questions (which are catalog variables) appear on the **Details** tab. Results of any playbooks appear on the **Automations** tab.

Previewing how a record will appear



Result

If you don't like what you see, you can tweak the record's appearance. Find out how in Change the layout of an app's record in Creator Studio.

Deploying your Creator Studio app

Congrats, you've built your app and are ready to share the first version of it with the world. That's great! Deployment is the next step in the process.

Deployment is a term that describes the process of an app moving from a non-production instance like development or QA through to production. Production is the instance that your customers see, or your "live" instance. To kick this process off, you should build your app in a non-production instance, and request that an admin deploy it to a production instance when you're ready.

Deploying an app automatically publishes the app, as well as all of the published forms and activated playbooks that you include, to a production instance.

Deployment uses pipelines configured in Pipelines and Deployments. If you don't have a pipeline configured, the Creator Studio home page alerts you and suggests you ask an admin to set it up for you.

The ServiceNow AI Platform uses the Application Repository to move apps between instances, so release notes and a version number are required. Find out more in App versioning and release notes for Creator Studio apps.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Requesting app deployment to production

You should test your app and all of its published forms and activated playbooks on a non-production instance. Once it's ready, you can submit the app for deployment to production. For more information, see Request deployment for your app from Creator Studio to production.

Apps aren't deployed directly from Creator Studio. Instead, your admin uses Pipelines and Deployments. Admins should check out Managing deployments using Pipelines and Deployments.

Deploying forms and automation with the app

You may feel some hesitation at this point, perhaps wondering if your app is actually ready. Well, you can take small steps and include only the forms and activated playbooks that you want in your deployment request.

When you submit an app for deployment, all of the app's published forms and activated playbooks are available to be deployed. When you request deployment though, you can specify:

- Published forms that you don't want to appear in the catalog after the app is deployed to production.
- Activated playbooks that you don't want to appear when users view records generated by the app after it's deployed to production.

If you don't activate a playbook and its app is deployed to production, the automation won't run on the applicable records. However, your App Engine admin can activate the deployed playbook for you.

Assigning users to a deployed app

Your app is officially published and ready to be used, congratulations! But, who gets to use it, and do they have access? Your admin gets to take charge here.

After your admin deploys the app to production, the admin must assign users and groups that can access the app. That is, you can't assign user access in the deployment/publishing process. See I've built my app in Creator Studio, now what? to find out more about where to access published parts of your app.

App versioning and release notes for Creator Studio apps

Each deployed version of an app must have a version number and release notes, which helps administrators track app usage and changes.

App version number

The version number enables admins to track which version of the app is deployed to each instance. Follow your organization's versioning guidelines, or use the x.y.z format, where x = major update, y = minor update, and z = patch.

Creator Studio automatically updates the next version when you update a deployed app, for example from 1.0.0 to 1.1.0 and then to 1.2.0.

App release notes

The release notes help admins track what the changes are between app versions. When submitting your app for deployment, specify information that helps people understand differences between app versions.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Request deployment for your app from Creator Studio to production

After you've tested your app's forms, playbooks, and workspace category on a non-production instance, for example a development instance, deploy it to a production instance so users can access it.

Before you begin

Your admin must set up a deployment pipeline before you can submit your app for review. Contact your admin if you need a pipeline configured.

To request that your app is deployed to production, you must be given permission to work on the app.

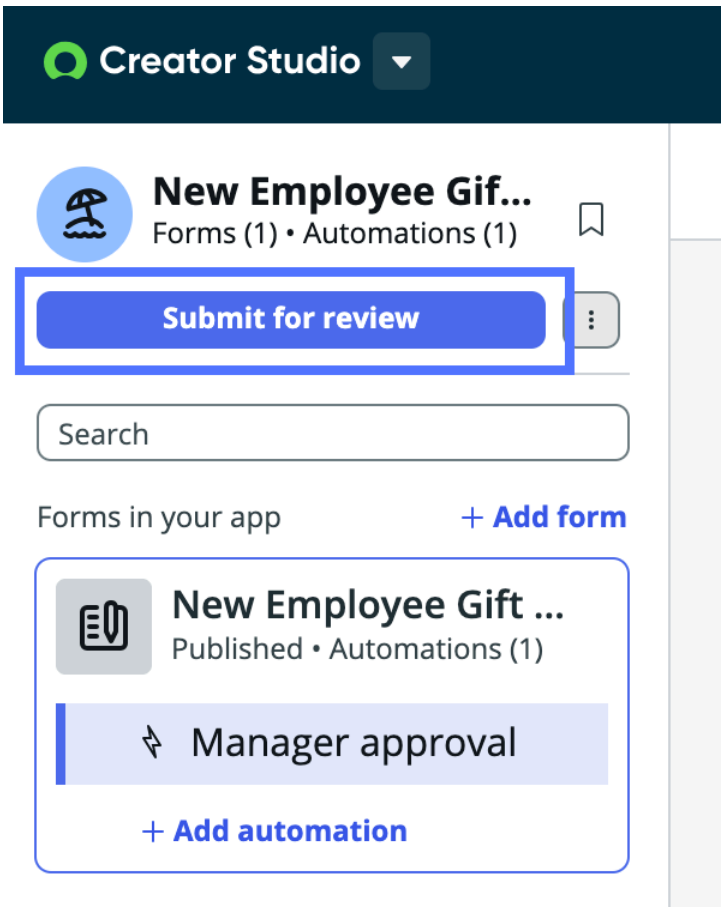
About this task

You can't cancel the review cycle after you submit an app for deployment review, so make sure that your app is ready to go.

Procedure

1. Go to **All** > **App Engine** > **Creator Studio** to see all the apps on the Creator Studio home page.
2. Open the app that you want to deploy to production.
3. In the navigation panel, select **Submit for review**.

Submit for review

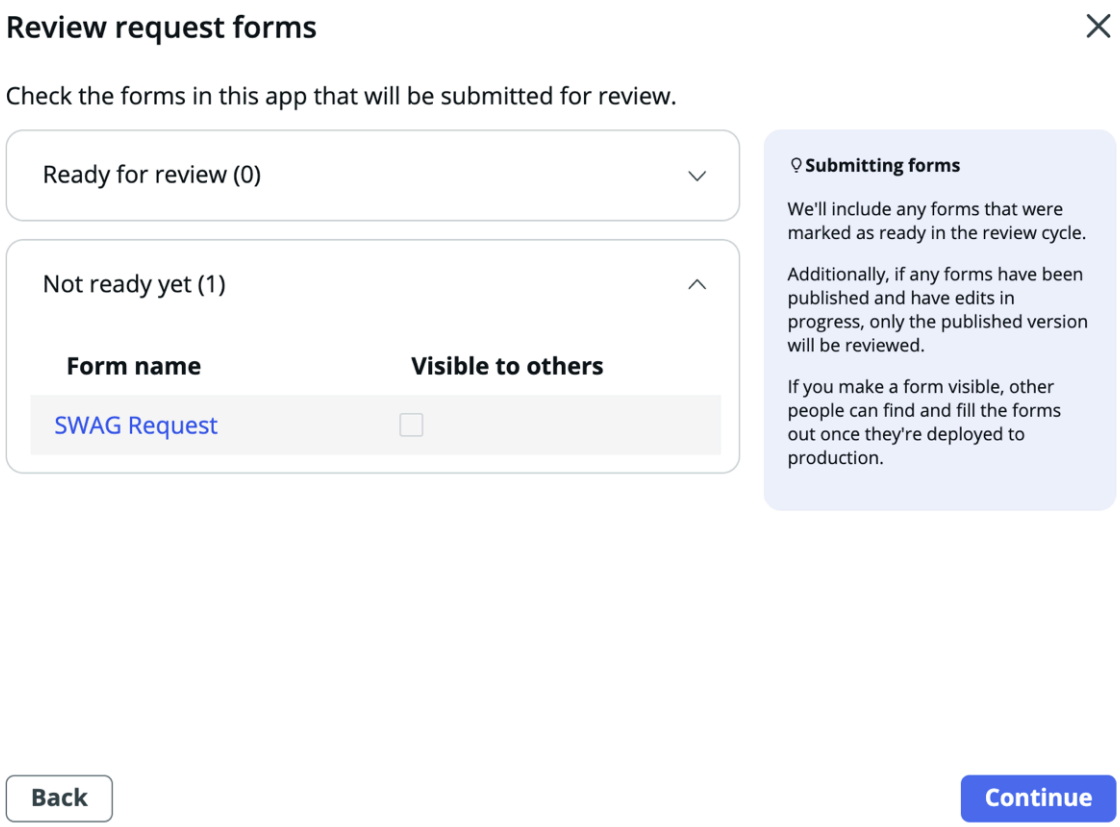


4. Select **Continue** on the Submit app for review modal.
5. Now you need to choose which published forms are visible to users in the catalog. In the Ready for review section of the Review request forms modal, select which of the app's

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

published forms that you want to be available after the app is deployed by selecting the **Visible to others** option.

Review request forms for deployment



6. Select **Continue** when you're happy with the forms being deployed to production.
7. Next, you must decide which of the app's activated playbooks will run on production after the app is deployed. In the Review playbooks modal, select the **Run on production** option for each playbook that you want to run on records that the app generates.

   Note: If you can't select a playbook, you need to go back to the **Automations** section of Creator Studio and activate it. If you need a refresher on that, check out Activate a playbook in Creator Studio.

   Unactivated playbooks still get deployed to production with the app, they just won't run on the records unless your admin activates them on production or redeploys the app with the playbook activated.
8. Select **Continue** when you're happy with the playbooks being deployed to run on production.
9. Finally, make sure that all the release details for the published app are correct.

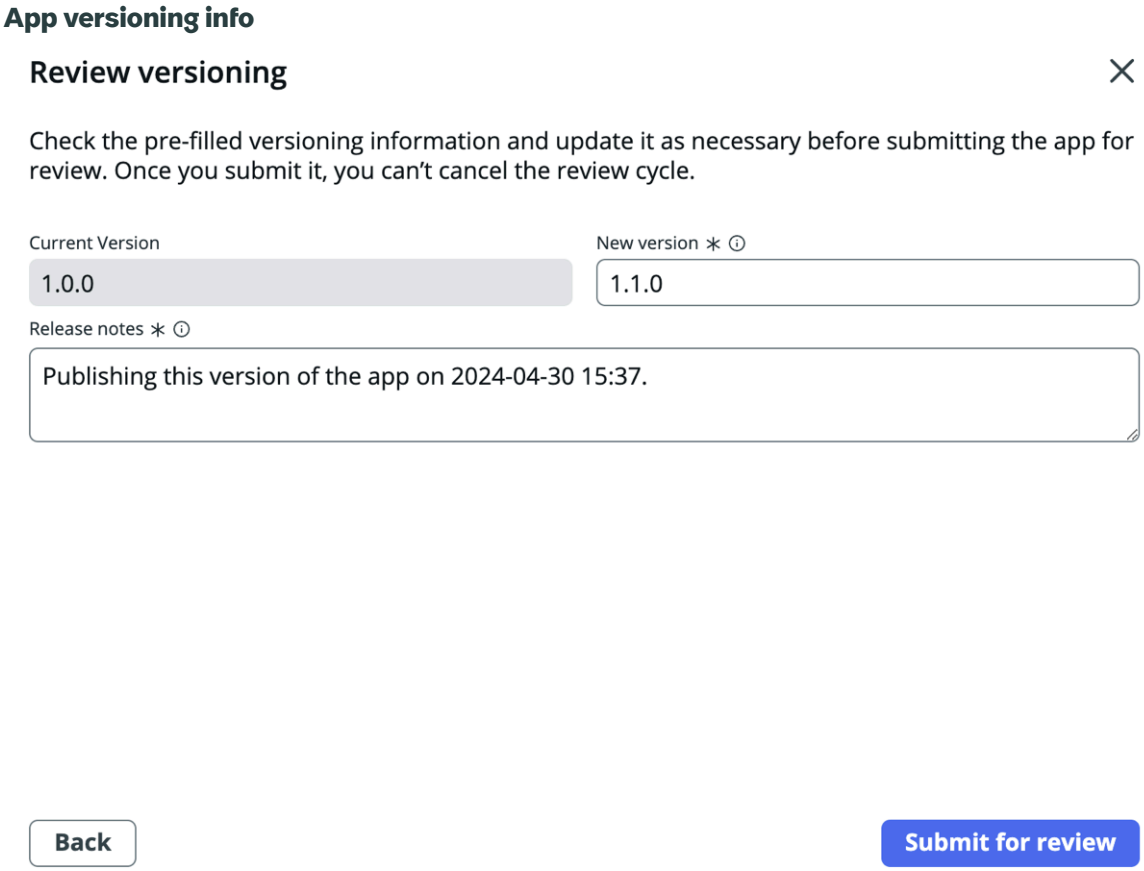
Versioning options for deployment

Field	Description
New version	Version number of the app you're requesting for deployment. Creator Studio automatically generates an updated version number, but you can change it.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
	Follow your organization's versioning guidelines, or use the x.y.z format, where x = major update, y = minor update, and z = patch.
Release notes	Details on what's changed in this new version of the app, or a general description of what the app does if this is its first version.

Read more about this step of requesting deployment in App versioning and release notes for Creator Studio apps.



10. Select **Submit for review** when everything is correct and ready for your admin to review and deploy.

Result
Woohoo! Your app is ready to be reviewed for deployment.

What to do next
Your admin uses Pipelines and Deployments to deploy the app to production. You can still make changes to the app after it's been deployed, you'll just need to request re-deployment when it's ready.

If you're an admin, check out Managing deployments using Pipelines and Deployments for the scoop on deploying apps.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

I've built my app in Creator Studio, now what?

You've added forms and automation to your app, and customized the workspace list configurations. Now what?

What happens after I submit my app for review?

When you finish creating your app and select to **Submit for review**, your admin gets a deployment request. They then review your app deployment request and move the app to production.

After the app is deployed to production, the forms are available on the ServiceNow AI Platform.

Where do I find my app after it's deployed?

After deployment, your app lives as forms in the Service Catalog and categories you specified when creating the forms.

Users can access those forms directly in a Service Catalog, as well as Service Portal and an Employee Center.

If you associate the app's form with one or more topics, the form will appear in the relevant, dynamically created topic pages in Employee Center. Find out more about topics in Associate a catalog item with a taxonomy topic in Employee Center, and more about taxonomy, which is a categorization method, in Unified Taxonomy for Employee Center.

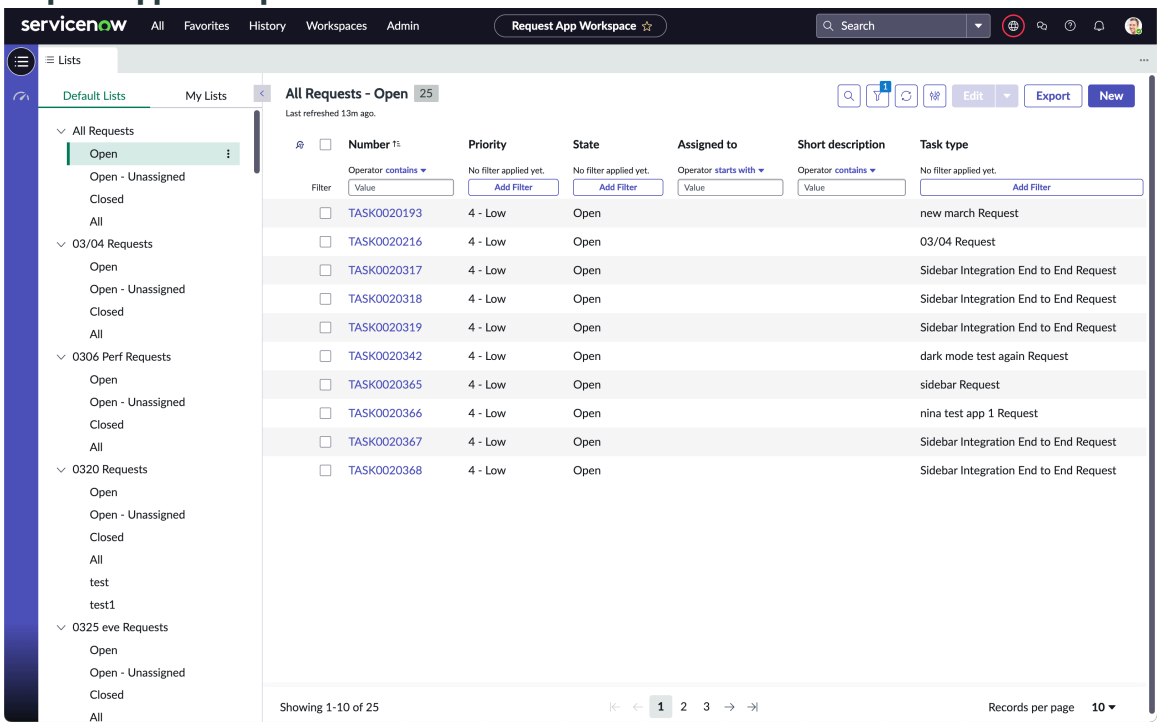
Where do fulfillers work on submitted forms?

Fulfillers access the default workspace on the ServiceNow AI Platform by going to **All** > **App Engine** > **Request App Workspace**. The Request App Workspace displays a category for each app with filtered lists of records, both default and any extra that you added. Fulfillers then select a record to view the details of the request and make updates as needed.

Note: You can't access the Request App Workspace as the owner of an app, though you can access it if you also have an admin or the applicable agent role. The app's agent role is <app scope>.agent, for example, x_snc_app_name.agent.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Request App Workspace



You can preview and interact with how records appear in the workspace category directly within Creator Studio by navigating to the **List categories** section of the navigation panel and selecting a record.

Can I continue developing my app in other ServiceNow AI Platform builders?

You can open Creator Studio apps in other builders. For example, open it in ServiceNow Studio to add more experiences or complex automations. An **Open in ServiceNow Studio** link, available from the app's tile on the Creator Studio home page, takes you directly to the app in ServiceNow Studio.

Get help and check out the FAQs

Got more questions? Take a look at the Creator Studio FAQs on the Community site !

You can also watch a quick video on how to contact your admin and check out some helpful resources.

Closing requests and app notifications in Creator Studio

The records that your app creates when a form is completed are not automatically closed, and some notifications are not automatic and must be configured.

Closing requests generated by Creator Studio apps

After an app's automation is done and the request is fulfilled, users must manually close the request record in the Request App Workspace. When fulfillers manually close the request, the ServiceNow AI Platform emails the person who made the request to notify them that the request is closed.

Additionally, you can create a playbook in the app to automatically email the requester when their request is closed. For more information, see Add activities to an app's playbook in Creator Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Default notifications for apps built with Creator Studio

Apps created in Creator Studio send the following notifications by default.

Note: Admins can customize the default notification emails for events listed in the following table. For details, see Administering notification emails in Creator Studio.

Default app notifications

Event	Recipient	Trigger
Request opened	User who opened the request from the app	Record in the app's Task table or associated table is created.
Request assigned to user	User the request is assigned to	Record in the app's Task table or associated table is created or updated.
Request assigned to group	All users in the group that the request is assigned to	Record in the app's Task table or associated table is created or updated.
Request closed*	User who opened the request	Record is updated any closed state, such as Complete or Cancelled.

*Remember, request records must be manually closed.

For more information on notifications from the ServiceNow AI Platform, see Create an email notification.

Notifications Creator Studio sends about building apps

Default app notifications

Event	Recipient
A Creator Studio Restricted User submits a new app request	App Engine Admin group
The requested app is approved and created by the admin	Creator Studio Restricted User who requested app creation
The app creation request has been fulfilled	App Engine Admin group

Creator Studio reference

Find details on the parts of building apps in Creator Studio, such as descriptions of all possible types of questions or layout options for forms.

Creator Studio form settings

You can edit the settings for an app's form in Creator Studio at any time. For example, to change who can access it.

General form settings

The **General** tab defines the form's basic information and how things appear on it.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

General tab for app settings

Field	Description
Form name	Name of the form. Apps can have multiple forms. If you want to edit the settings for a different form, select that form in the navigation panel.
Short description	Brief description of what the form is used for. For example, "Laptop request form."
Image	Image to appear on the form. Select the logo and navigate to a JPG, PNG, or SVG file to upload.
Description	Informative reason why people should use the form. You can include content like videos, images, and links to other sources of information, such as knowledge base articles.
Make form visible to others	Option to have the form available outside of the app, for example, in the specified catalogs. You can edit this option only for forms that have been published. **Note:** You can also specify whether the form is visible after deployment when you're ready to deploy your app. Find out how in Request deployment for your app from Creator Studio to production.
Hide "Add to wish list" option	Option to prevent users from favoriting the form on their wish list. Favoriting a form makes it easier to find.
Hide attachment option	Option to prevent users from adding attachments to the form. **Note:** You can hide the attachment option for the whole form and instead add an attachment question that appears when users provide a specific answer to a question. For more information, see Available question types in Creator Studio.
Mark attachment as mandatory	Option to require attachments on the form. If you select the **Hide attachment option** setting, you can't select this setting.

Uneditable form settings

After you create the app, its **Template** can't be edited. If you want to change it, create a new form using a different catalog template.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Catalog location form settings

Location settings define where the form appears in a catalog, how it's categorized, and what topics it appears for. You must select the edit icon Edit to make changes to the catalog and topic settings.

Select the catalog that represents the business area the app will use. For example, you could select a service catalog that contains software and laptop cables for an IT fulfillment app. Expand the caret for each catalog to see its sub-catalogs. Then select as many items in the catalogs as you need.

If you associate the app's form with one or more topics, the form will appear in the relevant, dynamically created topic pages in Employee Center. Find out more about topics in Associate a catalog item with a taxonomy topic in Employee Center, and more about taxonomy, which is a categorization method, in Unified Taxonomy for Employee Center.

User access form settings

Select the edit icon Edit to define access for roles and groups for the following:

- **Available for**: Search for and select roles and groups that can access the form.
- **Not available for**: Search for and select roles and groups that can't access the form.

Available question types in Creator Studio

Build apps using different types of questions on a form in Creator Studio.

You get to pick the questions to ask requesters. Creator Studio provides two types of questions you can ask:

- **General questions** are where you give requesters an empty box to write a short answer (think name or email) or a list of choices to pick from (like a dropdown menu or those little circles you fill in).
- **Reference-based questions** are where you provide a list of choices that you define in a table.

The type of question you select from the **Content type** field in the Question details panel changes the question's settings that appear.

Types of general questions

General question type form elements

Content type	Description
Short answer (Single-line text)	Single-line question.
Multi-line text	Longer option that provides space for a question multiple lines in length.
Dropdown	Question with a list of pre-defined answers that users select from a dropdown menu. Users can select only one answer.
Radio button	Question with list of pre-defined answers that users select from radio buttons. Users can select only one answer.
Check box	True or false question that users answer by selecting the relevant check box.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

General question type form elements (continued)

Content type	Description
	Note: If you put two checkbox questions side-by-side on a form, they make a section. You can't add other types of questions to that section.
Yes/No	Question where users select Yes or No as their answer.
Date	Question where the answer is a date that users must select from a calendar.
Date and time	Question where the answer is a date and time that users must specify.
Question set	Preconfigured questions that you can't change. Admins create question sets in Service Catalog. For more information, see Service catalog variable sets. **Note:** • You can't change an existing question into a question set. To include a question set on a form, you must newly add it to the form. • If you put two checkbox questions side-by-side on a form, they make a section. You can't add other types of questions to that section.
Attachment	Question where users add an attachment. **Note:** The **Attachment** question differs from the **Add attachment** option for the overall form. You can use the **Add attachment** question to include in dynamic behavior, for example appearing when a user answers a **Photograph on badge?** question with `Yes`, enabling them to upload a headshot. For more information on dynamic behavior, see Make a form change based on responses in Creator Studio.
Duration	Question where users specify a length of time. When you add a **Duration** question, you then specify the **Days**, **Hours**, **Minutes**, and **Seconds** that an event will last.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

General question type form elements (continued)

Content type	Description
	Note: You can also add duration for dynamic behavior questions. For more information, see Make a form change based on responses in Creator Studio.

Types of reference-based question types

Reference-based questions pull their possible values from the table that you must designate as the **Source table**.

Reference-based question form elements

Content type	Description
Record choices	Question where the answers are retrieved from an additional specified table when the filter conditions, which you must also define, are met. Use this question on a form that references data from another form, such as a form that uses the Incident table. "Choose the related incident" is an example Record choices question, which could use a filtered list of incidents as possible answer selections. If you have a Record choices question on a form, you'll see an **Add auto-fill** option when you configure other questions on the form. Select the **Add auto-fill** option to choose a value from the Record choices question's table to automatically populate the question's answer. For example, a form can enable someone to select a user for the **Choose the gift box recipient** field, which retrieves its values from the User table. An additional **Preferred name** field can be automatically populated from the **First name** field of the User table selection, but then edited if someone prefers a nickname.
Field choices	Question where the answers are retrieved from a field on the specified table when the filter conditions, which you must also define, are met. For example, you could create a Field choices question for a table with a "Country" field, which the user could then use to select their country.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Reference-based question form elements (continued)

Content type	Description
Multi-select	Question where users can select multiple answers that are retrieved from the specified table when the filter conditions, which you must also define, are met. Use this question on a form that references data from multiple other forms, such as a form that pulls answers from multiple separate Incident tables.

Question details

Depending on the type of question you add to a form, you must specify some of the following details.

Question type details

Field	Description	Available for
Question label	Question or statement that says what the question is asking.	All question types
Help text	Additional information that appears as a tooltip to guide the user when they hover over the information icon.	All question types
Show questions on form	Hide the question from the form by leaving this option unselected. You can use this option to associate data with the form behind the scenes by adding it to the app's table.	All question types
Mark as required	Option to make the question required before the user can submit the form.	All question types
Mark as read-only	Option to make the question read-only. For example, if you don't automatically pull the user's role into the form's table, you could create a single-line text question where the default value is their role (such as Sales) and then hide it. You could even hide the question	All question types

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Question type details (continued)

Field	Description	Available for
	from the form to make the data exchange invisible to the user.	
Default value	Value that's auto-populated in the question's answer when the form is opened. For reference-based questions where you designate a source table, you must use an asterisk (`*`) as a wildcard when searching for values in the **Default value** field. For example, if the question references the User table and you're looking for Abel Tuter, entering `Tuter` won't find that user but entering `*Tuter` will.	All question types
Choices	Customizable answers to the question, which you can drag to rearrange in a new order.	• Dropdown • Radio button
Include "None" as a choice	Option to include **None** as a possible answer to the question.	• Dropdown • Radio button • Field choices
Default to the first choice	Option to display the first choice as the possible answer when the user sees the question.	Radio button
Source table	Table location that provides data for answers to the question.	• Record choices • Field choices • Multi-select
Field	Field in the specified source table that possible answers are retrieved from.	Field choices
Filter conditions	Conditions that must be met to view certain choices. Select **Add condition** and use the condition builder to create the filter. For more information, see Condition builder.	• Record choices • Field choices • Multi-select

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

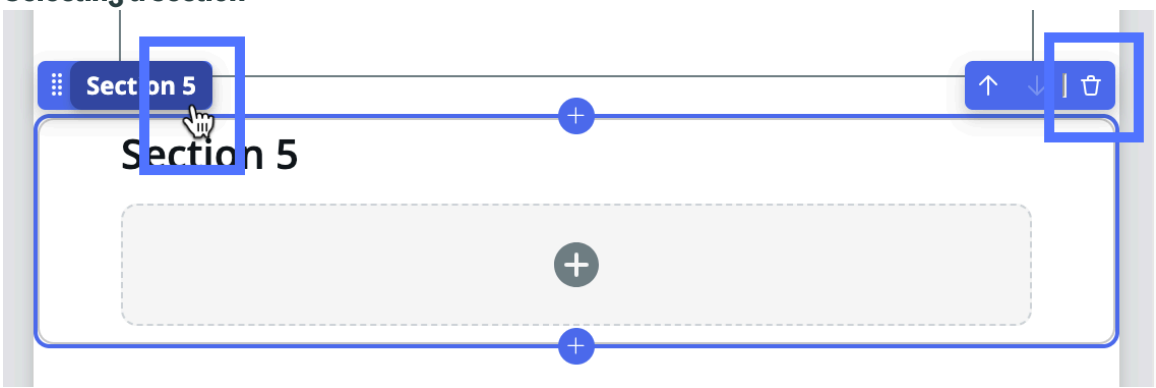
Layout options for forms in Creator Studio

Adjust how your form, which creates a catalog item, appears in Creator Studio using layout options.

Layout options enable you to design a form by dragging them from the Form elements panel to the form canvas. You can also drag layout options to rearrange them.

Note: To edit or delete a section, you must hover over the section name and then select **Section** to see the section details in the properties panel, as well as the delete icon.

Selecting a section



Layout options

Element	Description	Details
One-column section	Section with optional header (label) where all questions in that section of the form appear in a single column.	• Specify a **Section label** to give the section a header. • Select **Show section label on form** to display the section header.
Two-column section	Section with optional header (label) where all questions in that section of the form appear in two columns.	• Specify a **Section label** to give the section a header. • Select **Show section label on form** to display the section header.
Divider line	Displays a line between questions or sections.	N/A
Heading	Displays header text, such as a question or a statement based on what type of information you're asking the user to provide on the form.	Specify header text in the **Name** field.
Rich text	Displays longer text that you can enhance with options like links, different fonts, and bullets and tables.	Use the rich text editor to enhance any supplemental text with the following typographic formatting:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Layout options (continued)

Element	Description	Details
		• Bold/Italics/Underlining • Font face and size • Font and background color • Tables • Links • Images • Source code to include scripting • Alignment • Bulleted list • Numbered list

Creator Studio glossary

Learn about the terms and concepts used in Creator Studio.

activity

Activities in a playbook complete automated tasks, such as obtaining approvals, assigning records, and sending emails.

automation

Tasks or requests moving through a workflow without manual intervention.

board view

A traditional layout of viewing a playbook, which shows each activity as a card in the navigation panel. When you select an activity, for example a task, its editable details appear in the canvas for you to work with them.

catalog

A collection of items that represent all the items a requestor can ask for.

collaboration descriptor

Level of collaborative access assigned to a developer. The collaboration descriptor that someone is assigned determines if they can assign, manage, and monitor delegated development permissions. For example, people who are Owners of an app can do more than people with the Editor collaboration type.

delegated development

A collaborative process that enables developers to invite other people into apps so they can co-create and develop the app together.

deployment

The process of an app moving from a non-production instance like development or QA through to production. Production is the instance that your customers see, or your “live” instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

diagram view

A visually intuitive diagram that shows you the order that activities occur in the playbook. When you select an activity, such as the trigger or a decision, a modal pops up where you can edit its details.

dot-walking

Describes the structure of a variable that contains one or more reference fields, separated by a dot for each table involved.

dynamic behavior

A form for a catalog item can use dynamic behavior to automatically update based on how users answer a question. For example, if a user says they don't want lunch for an event they're attending, you can make a **Meal preference** field disappear.

Note: Dynamic behaviors are also called Catalog UI Policies.

extend a table

When you create a table based off an existing table. Extending a table means the new table inherits the parent (extended) table's columns, as well as its business logic. Apps built in Creator Studio generally extend the Request Task table.

field

A field in Creator Studio is a column on the app's underlying table. If you're using condition builder, a field is either a column on the table or a question from the form.

form

A content page that displays fields and values for a single record from a database table.

form template

A pre-defined set of questions arranged on a form provided by Creator Studio.

fulfiller

Someone who works on requests. Fulfillers may also approve or deny requests, depending on any approval automation for the app.

item

Something a requestor can ask for, such as new computer equipment or time off work. Each item requires its own form in Creator Studio.

playbook

A set of automated activities that occur based on a trigger. For example, sending an email when a request is approved. Think of playbooks as pre-programmed workflows that kick in after a form is submitted. They are like if/then statements: If this happens, then that will happen.

record

A task or request that needs to be acted on. All rows in that table for an app are individual records. Records have different states depending on what actions have been taken.

reference-based question

A question on a form that retrieves its possible answers from the table that you define.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

release notes

The information about what's included in each new version of the deployed app. Release notes help administrators to track what the changes are between app versions.

requestor

Someone requesting something, like a piece of equipment or permission to do something, such as take time off work.

target table

The table where apps created in Creator Studio store their requests. Admininstrators can change the table that an app saves its data to. For more information, see Administering an app's associated table.

trigger

The event that makes an automated playbook begin running. For example, submitting a request automatically assigns it to a manager.

version number

A number assigned to the app that increases with each subsequent time it's deployed.

workflow

Defined sequence of steps automating processes in Sales Customer Relationship Management.

Build apps using App Engine Studio

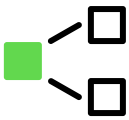
ServiceNow® App Engine Studio (AES) is a guided, low-code tool for developing rich web applications to store information, automate business processes, and solve business problems. Work that was once assigned to administrators can now be delegated to employees with little to no training using AES.

Get started

By delegating development to citizen developers in your organization, administrators are freed up to address more strategic, system-wide priorities. AES can help you do the following:

- Build powerful apps quickly to fit any business need by using a suite of development tools, including guided workflows and templates.
- Create experiences users love with Catalog Builder, Workflow Studio, Workspace Builder, Mobile App Builder, and other integrated tools.
- Scale apps without sprawl with straightforward governance and tools that support both citizen and pro developers, such as App Engine Management Center (AEMC).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Explore  Learn about App Engine Studio concepts and features.	**Install**  Install App Engine Studio and its associated apps.	**Configure**  Configure environments, tools, and user access.
Build  Build applications using App Engine Studio.	**Deploy**  Deploy applications to the instances in your pipeline, from development to production using App Engine Management Center.	**Reference**  Get details about App Engine Studio components such as fields, tables, and properties.

Additional resources for App Engine Studio

Learn more about App Engine Studio	ServiceNow resources
App Engine Studio is a guided, low-code tool for developing rich web applications to store information, automate business processes, and solve business problems.	 Create Apps Fast with App Engine Studio
	 Build Apps Fast with App Engine
	 Getting Started on App Engine Studio

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Request AES on the store

Visit the ServiceNow Store to view all the available apps, and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the ServiceNow Store version history release notes.

Before you can use AES, you must first download it from the ServiceNow Store. When you have completed the download, you may need to clear your local cache before it appears on your instance.

Get help with App Engine Studio

- Contact your company's Customer Admin to unlock or add user accounts, perform restores or zBoots, and more.
- Ask or answer questions in the App Engine Studio community forum
- Search the Known Error Portal for known error articles
- Learn more about how to create your own apps on the developer site.
- Contact Customer Service and Support

Exploring App Engine Studio

Learn how you can use the low-code app development tool, App Engine Studio (AES), to build powerful applications using guided setup, predefined templates, and workflows.

App Engine Studio users

Users

User	Description
Citizen developer	Citizen developers without programming experience can efficiently create applications on a simplified low-code, no-code platform that uses graphical interfaces and templates. The lower barrier to entry means developers of all skill levels can build applications.
Pro-code developer	Pro-code developers are familiar with using programming languages to build applications.
App Engine Studio admin	App Engine Studio admins manage the processes related to application development in AES. They review new application ideas, manage application development and deployment, and manage collaborator access, usually in the App Engine Management Center.

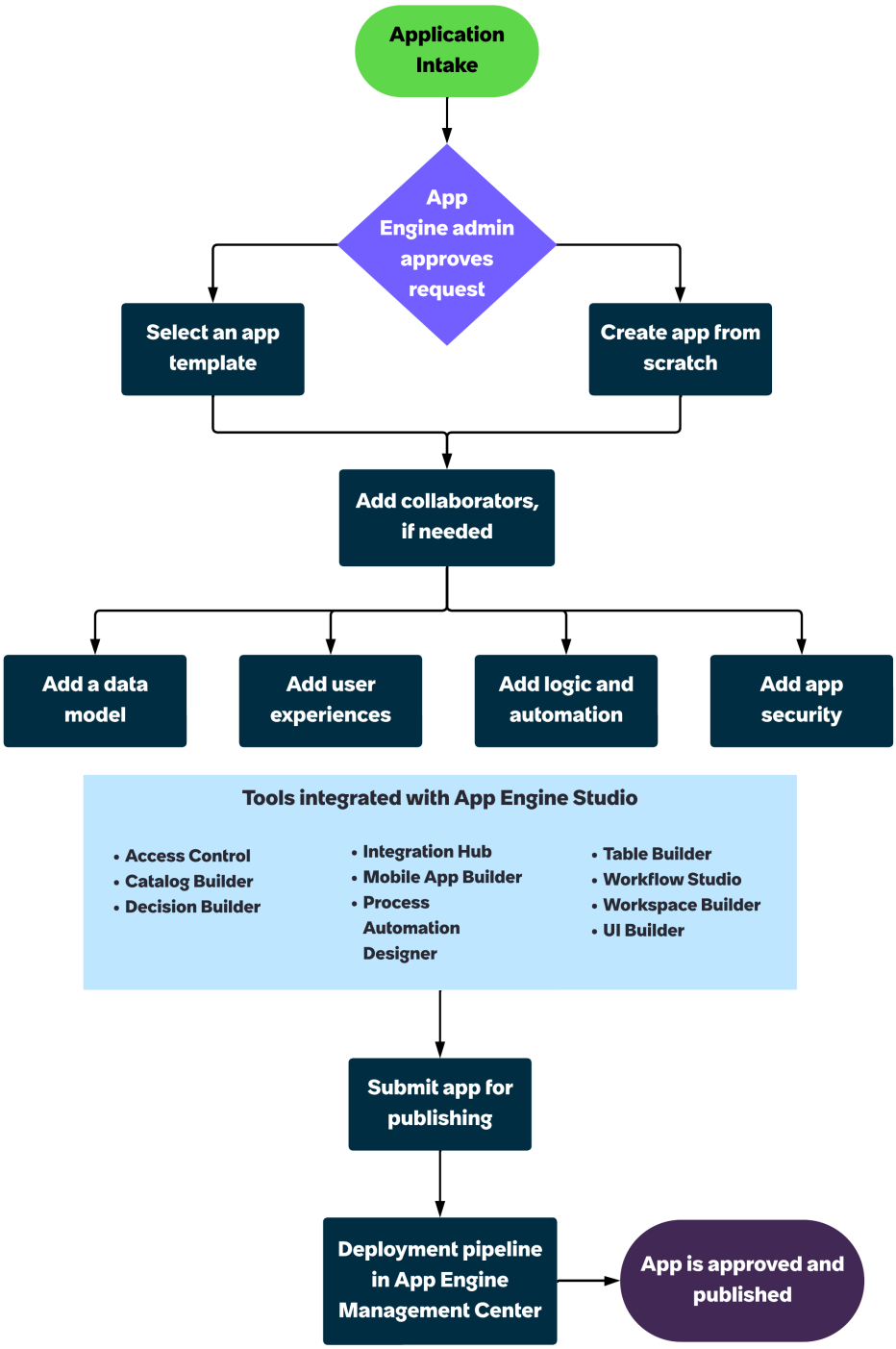
Workflow to create apps using App Engine Studio

The following illustration describes the basic tasks involved in creating an app using App Engine Studio. For detailed instructions, see Building apps in App Engine Studio.

Note: Select the image to enlarge it.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Application creation process using AES



The workflow for creating apps in App Engine Studio is as follows:

1. As a citizen developer, you submit your plan for an app through Application Intake for approval and development in AES.
2. The AES admin reviews the intake request. If it's rejected, that closes out the request. But you can always update your idea and submit a new request.
3. When the request is approved, you must decide whether to create your app using a predefined template, or to create it from scratch. You should explore the available templates. They can save you lots of time.
4. Then you must determine whether you have the resources to build the app on your own. If you need help, you can request collaboration, or the AES admin can

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Additional resources for App Engine Studio

Learn more about App Engine Studio	ServiceNow resources
App Engine Studio is a guided, low-code tool for developing rich web applications to store information, automate business processes, and solve business problems. **Note:** AES requires an App Engine subscription or product packaged with one. For more information, refer to Installing App Engine Studio. Contact your account representative for details.	 App Engine Studio Playlist  App Engine Studio vs. Studio  ServiceNow Community site  Build My First App Engine Studio Application - ServiceNow Developers site  App Engine Migration Assessment

Get to know AES

AES is designed to address enterprise low-code development needs by putting the power of the ServiceNow AI Platform in the hands of low-code developers. Guidance-driven flows and easy-to-customize templates help citizen developers to come up to speed quickly.

Regardless of their skill level, citizen developers become more productive using AES. Whether you're building a custom app from scratch or using a template, the simplified low-code experience in AES speeds up development on the ServiceNow AI Platform.

Key features of the AES include:

- **Guided experience for first-time users:** AES features a guided experience that orients developers, making it easy to navigate the environment. The first time you use the guidance, you're stepped linearly through the entire process of creating an application. After that, the guidance displays assistance related to what you're working on. You can hide the guidance by selecting **Don't show me this again** and exiting the pop-up.
- **App templates:** App templates simplify the app development process. Using templates based on proven use cases gives developers the ability to create powerful apps without the learning curve.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- **Guardrails:** Citizen developers receive tooltips and recommendations along the way, keeping them on track. Submitting apps for approvals also ensures quality.
- **Collaboration:** Developers of all skill levels can seamlessly collaborate on app development in AES's low-code environment.
- **Integrated tools:** Why work with multiple low-code tools when AES integrates with several? From designing tables to building catalogs and creating custom flows, AES has it all.

App Engine Studio benefits

App Engine Studio provides the following benefits.

Benefit	Feature	Role
Submit ideas for an app, which admins then approve.	Submit your idea for app development	Citizen developer
Use templates for standard application types, such as procurement portal or time off request, as a starting point to create and customize new apps quickly.	Create your app using an application template	Citizen developer
Access multiple development tools in one interface, such as Table Builder and Workspace Builder.	Integrated development tools for AES	Citizen developer
Scale development while maintaining process to review and test applications before deployment.	Managing app development using the App Engine Management Center	Administrator

Accelerate your low-code development with AES

Using App Engine Studio's low-code development environment, citizen developers can quickly build high-quality apps. Citizen developers are individuals in your organization who aren't experienced programmers, but who are subject matter experts with ideas for creating apps. With an approved idea in hand, a citizen developer chooses an app template, fills in a few parameters, and AES builds the app. The resulting application includes data, experiences, logic and automation, and security, all of which can be customized to your organization's needs.

Apps created from a template can be easily modified to meet the requirements defined by the citizen developer, or new apps can be created from scratch.

You can perform the following tasks to customize apps created using AES.

Task	Description
Create your data foundation easily	App Engine Studio templates contain default data tables that include everything you must deliver base system app functionality. Using the Table Builder app included with AES, you can edit or add new tables, add and delete columns, change table types, and more.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Task	Description
	You can also import existing data from spreadsheets, making it easy to automate existing manual spreadsheet-driven workflows.
Deliver great experiences	Experiences are role-based user interfaces that end users access to interact with your app. AES includes several distinct experience types, including configurable workspaces, portals, catalog items, and mobile experiences. You can add an experience by following a few simple steps. For example, you can quickly create a workspace by adding a name and description, a URL, and the roles that can access the workspace. Then you can tailor the workspace experience to the needs of your organization using the Workspace Builder tool within AES. More advanced configurations can be edited in UI Builder.
Automate your apps with workflow templates	AES templates include a set of powerful digital workflows that you can use to simplify the user experience, increase productivity, and strengthen your business processes. Simply choose from a list of common automation activities, and let the wizard quickly guide you through configuring the workflow. To adapt template-generated workflows to your needs, launch the Workflow Studio app to edit workflows.
Keep your apps secure	You can maintain control over who uses your apps by adding roles in AES. The app templates include predefined roles, giving you a head-start on security. You can add or delete existing roles, or create roles to meet your security needs.

Scale development while maintaining oversight

App Engine Studio lets you scale development across your organization while establishing and maintaining guardrails. You can control access to AES. You can also review apps created by your citizen developers before they're moved through your pipeline and published to your production environment. This review process gives you confidence to ensure the quality of the apps you create, release, and deploy.

What to explore next

To learn more about configuring and using App Engine Studio, see:

- Configuring App Engine Studio and related apps
- Building apps in App Engine Studio
- App Engine Studio reference

AES user interface

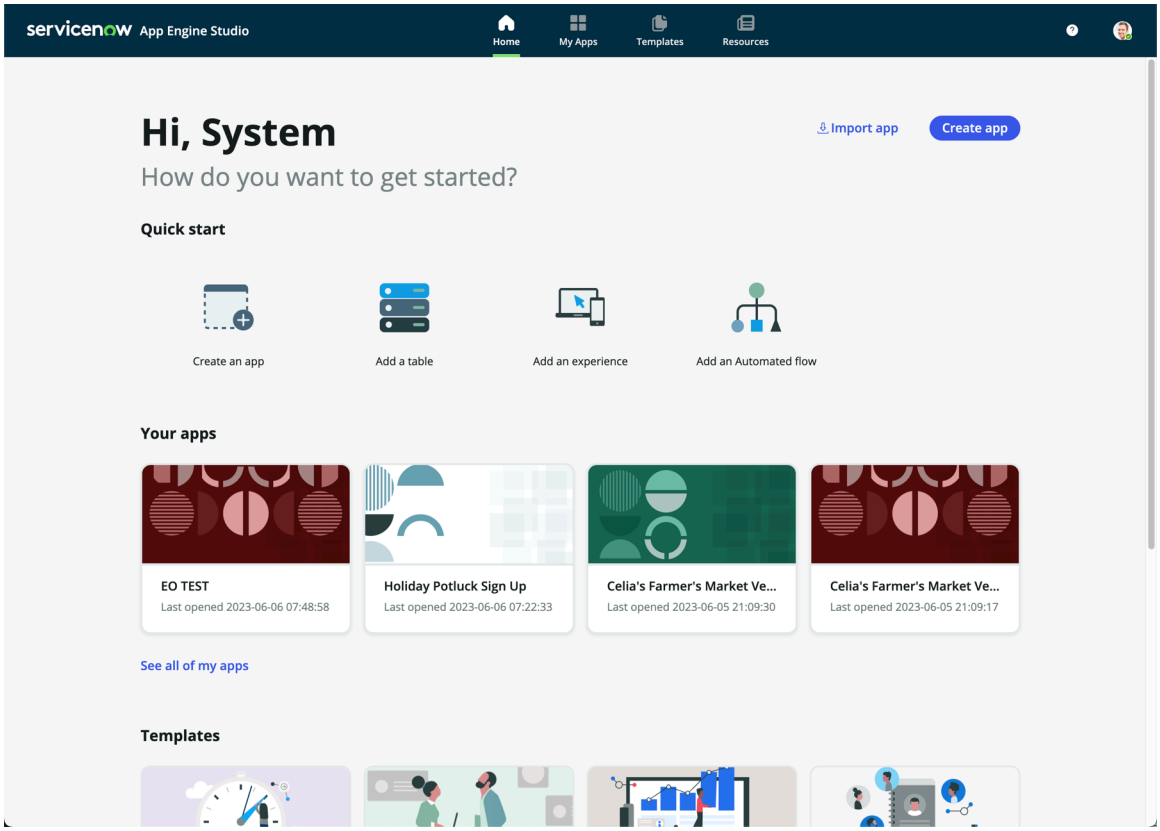
Learn about the App Engine Studio (AES) user interface.

AES is a web user interface that lets you build custom applications for your organization. You can build applications using templates which can be customized to fit your organizations needs.

App Engine Studio home

The App Engine Studio home page provides easy access to creating apps, viewing your app, quick start actions, and templates.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Use the top navigation bar to view your apps, access app templates, and find help resources. You can also access the help center by selecting the in the top right.

Users can access the following in AES:

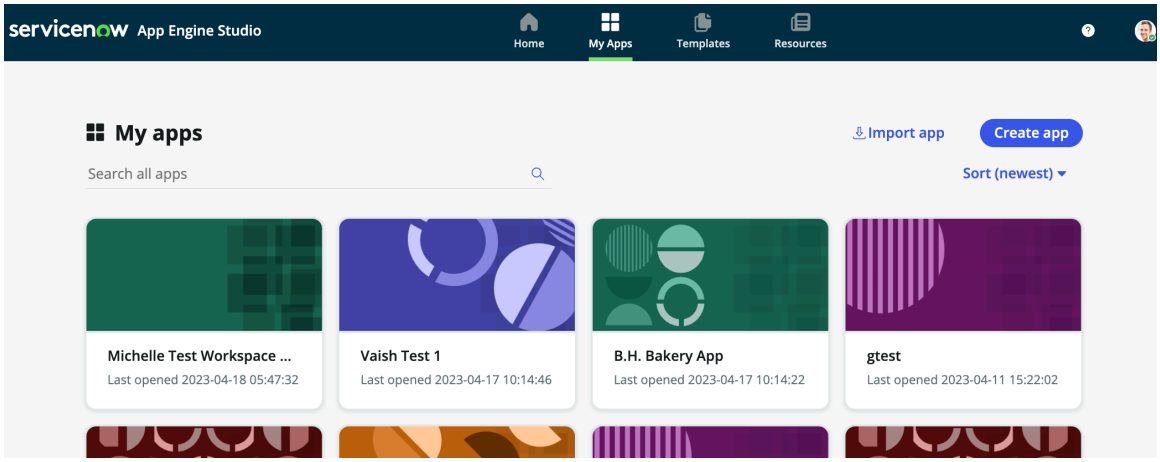
- Create app — Create an application from scratch.
- Quick start — Links to add objects to apps, browse app templates, and learn more about the App Engine Studio tools.
- My recent apps — A list of applications you have accessed recently using App Engine Studio.
- Templates — A list of templates you can use to create an application with preconfigured data, experience, logic and automation, and security.

> **Note:** If you're in the App Engine Studio User Limited group, you won't see any templates on this tab.

My Apps tab

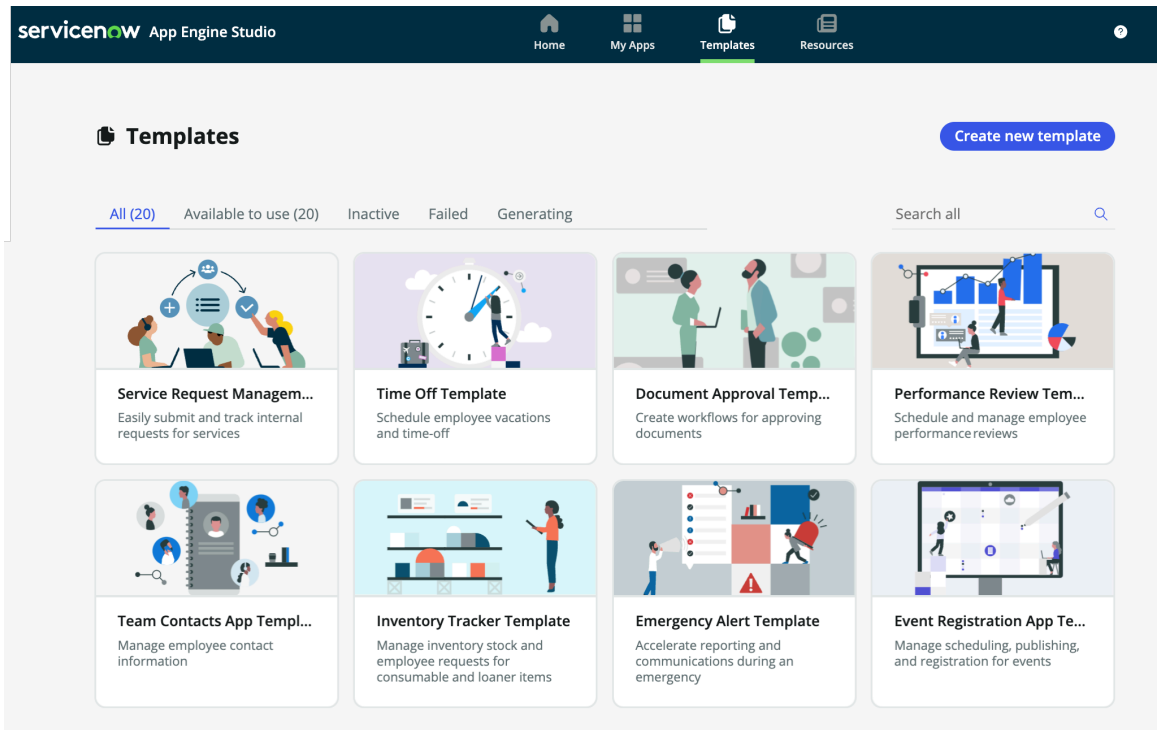
View and search the applications that you created.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Templates tab

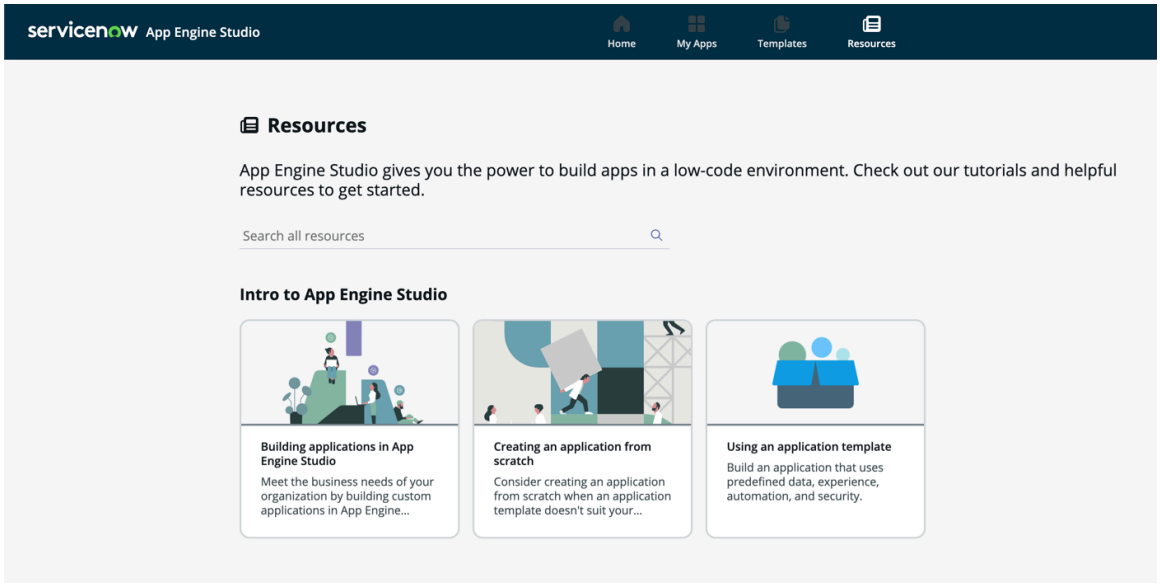
Provides a list of available app templates.



Resources tab

View tutorials and helpful resources to get started in AES.

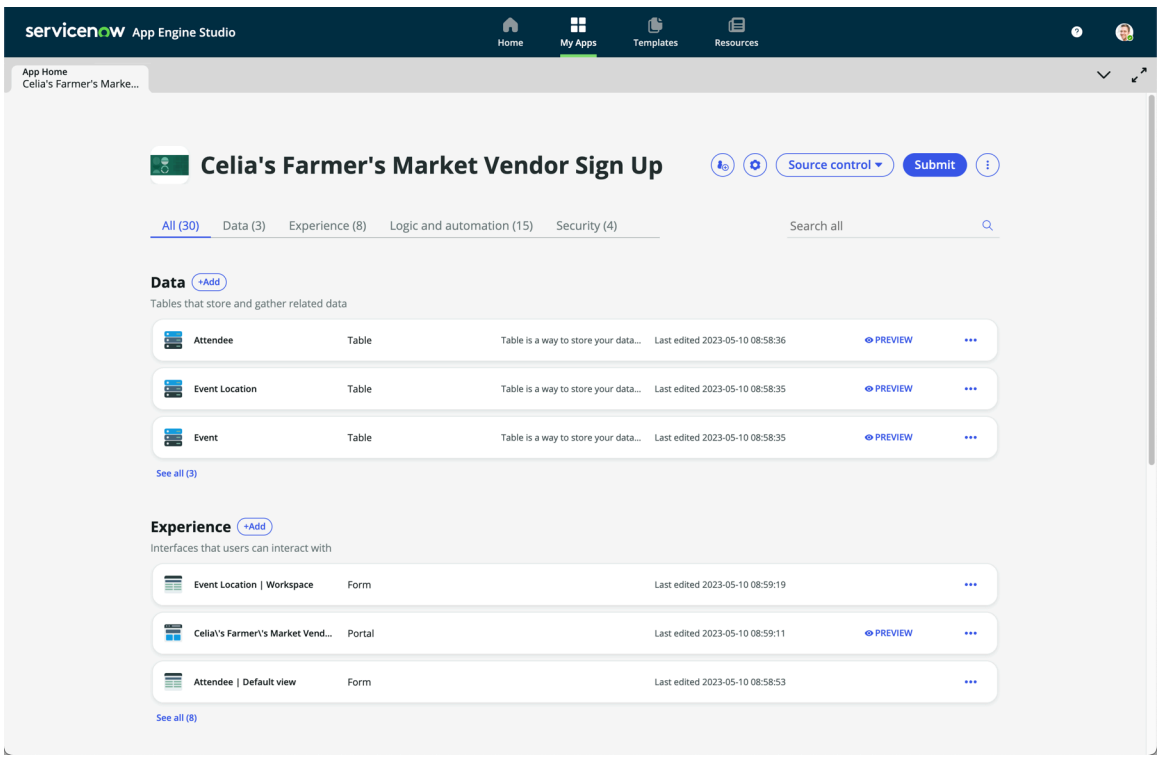
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



App home

Add data, experience, logic and automation, and security to your app from the app home.

Note: When viewing an app home page in AES, if you tab to a section (such as tables or experiences) with more than three items, the **Sort by** option enables you to display them alphabetically or temporally, by **Newest** or **Oldest**.

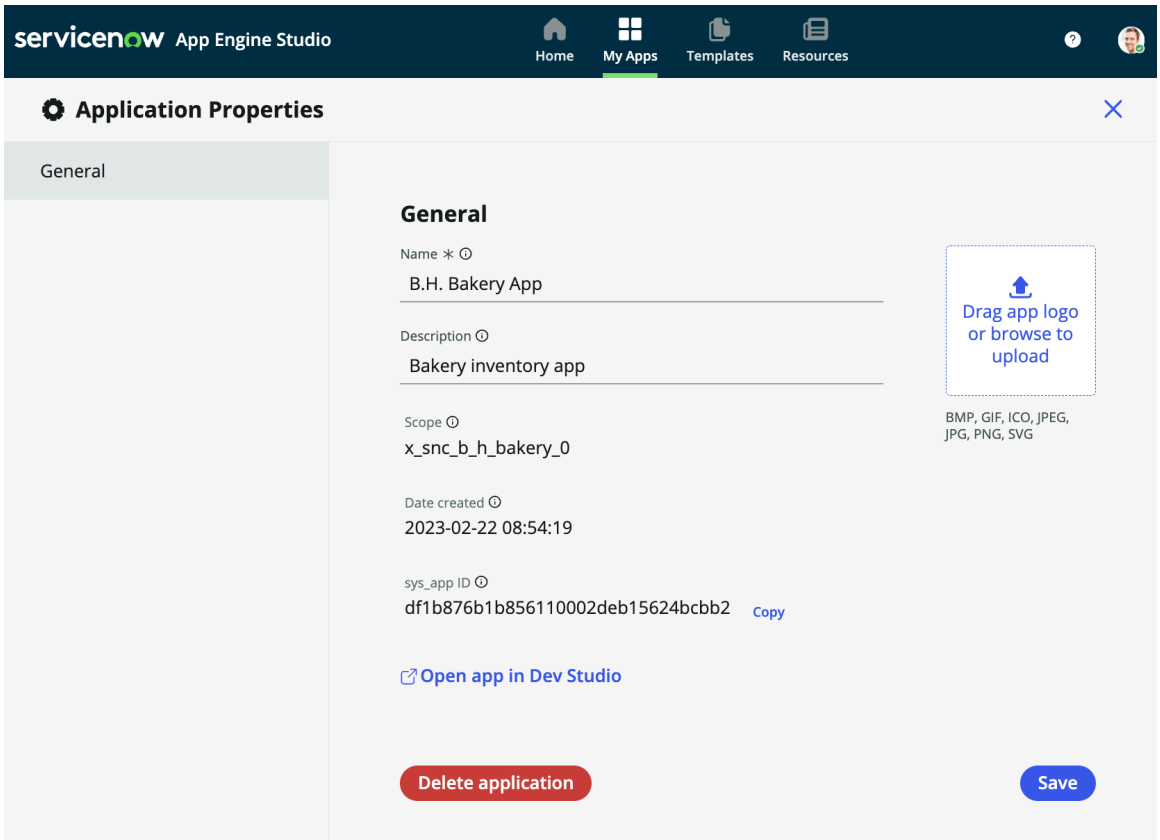


Some objects require access depending on your role. You can request access by selecting **Contact your system administrator** to become a delegated developer. For more information on the delegated developer role, see Delegate developers using AES.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

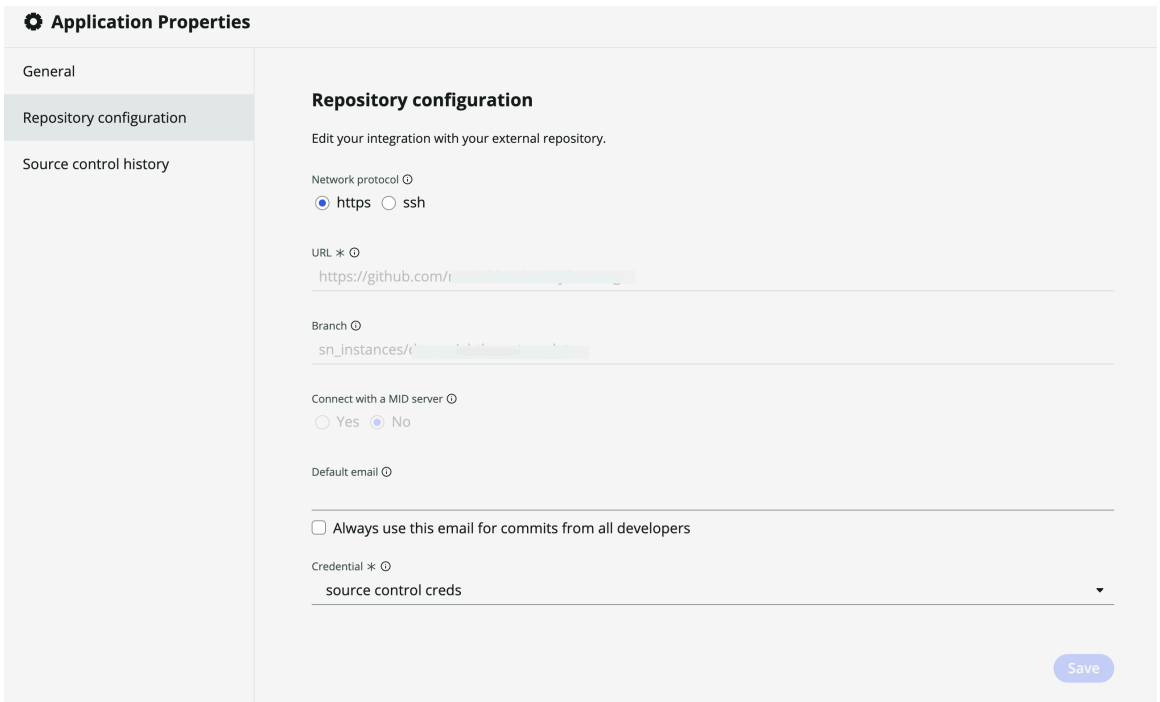
Application properties

View general settings of your application such as the name, description, and logo. You can also open your application in ServiceNow Studio or delete your application.



Repository configuration

View and edit the source control repository settings for your application.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Migrating to Build Agent

Build Agent is the agentic development experience on the ServiceNow AI Platform. Migrate your applications from App Engine Studio to Build Agent to expand your app's capabilities and take advantage of the agentic developer experience.

When to migrate to Build Agent

When building apps with App Engine Studio (AES), you may reach a point where your development needs exceed the scope of what AES can offer. You might need to add custom logic to an app, add scripting, or build global apps instead of scoped ones. When this occurs, you need to migrate your application to a development environment that supports more complex capabilities. Build Agent, the agentic developer experience on the ServiceNow AI Platform, supports these advanced capabilities, and the process of migrating apps to Build Agent happens in the background.

Build Agent is embedded within the same development environments (ServiceNow Studio and the ServiceNow IDE) that AES you already use when your work exceeds AES capabilities. The agent generates full-stack applications and artifacts using natural language prompts. To learn more about Build Agent, see Build Agent.

Whether you have an AES app that you want to extend, or you want to build something new, Build Agent offers the full suite of developer tools and the agentic development experience on the ServiceNow AI Platform.

What migration involves

Migrating to Build Agent is more of a development workflow adjustment than an actual migration process. When you're ready to migrate to Build Agent, the agent converts the the AES app to ServiceNow Fluent, enabling you to work in an integrated development environment (IDE). When your app is converted to ServiceNow Fluent, you can also work in ServiceNow Studio. Existing applications and app artifacts, such as tables, flows, workspaces, ACLs, and playbooks, are already accessible in ServiceNow Studio and available to be modified with Build Agent.

What does change is how you complete tasks. AES developers are accustomed to thinking in terms of wizard steps and form completion, which guides them through completing specific tasks. Development using Build Agent, however, operates using natural language prompts and iterative development. So if you plan to migrate to Build Agent, you must adjust to the cycle of prompting, reviewing, and accepting or rejecting Build Agent output.

You also gain access to scripting, global scope applications, and modification of base system applications with Build Agent. For some, this might mean learning new systems or developing new practices for a different scale of work. For more information about adjusting to the Build Agent development workflow, see Getting started with Build Agent.

Key differences between App Engine Studio and Build Agent

In addition to the workflow adjustment, there are several differences between AES and Build Agent that affect the developer experience.

Interface

AES uses a guided wizard interface, while Build Agent uses a conversational prompt interface inside ServiceNow Studio or the ServiceNow IDE.

Scope

AES supports scoped applications only and creates only the artifacts enabled by its integrated builders, such as tables, experiences, flows, workspaces, and ACLs. Build Agent supports scoped and global applications and can create the full set

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

of development artifacts, including business rules, Script Includes, Client Scripts, ACLs, and UI Builder pages.

Script access

To add scripting to AES applications, you must convert the application to ServiceNow Fluent and open the application in an IDE, such as ServiceNow Studio. Because Build Agent is already embedded within an IDE, the agent generates and modifies scripts.

Base system application modification

With AES, you can modify only custom, scoped applications. Build Agent supports the modification of any application, including base system applications.

For a full side-by-side comparison between App Engine Studio and Build Agent, see App Engine Studio and Build Agent feature comparison.

Getting started with Build Agent

Use the following guidelines to help you transition from using App Engine Studio to Build Agent.

From wizard completion to intent-driven prompts

App Engine Studio (AES) development is a series of guided wizard steps that result in building an application and app files. Each step corresponds to a discrete decision, such as adding a table, adding a field, or configuring a form. The wizard interface constrains what you can do and sequences the choices for you.

With Build Agent, development occurs via a conversational interface. In either ServiceNow Studio or the ServiceNow IDE, you describe the application that you want, who it's for, and what it should contain. From that description, Build Agent produces the artifact. Unlike AES, the Build Agent interface doesn't constrain your choices. Instead, Build Agent interprets your intent and generates output accordingly.

This process means learning how to write prompts that are clear and effective. Prompts should lead with the purpose of the application and the key workflows it needs to support, rather than describing a request for a specific artifact. For guidance on effective prompting, see General guidelines for Build Agent and Example prompts.

Owning the review step

With AES, the constraints imposed by the wizard interface meant that you were limited in what you could create accidentally. For example, you were unlikely to create apps or files that affect entire instance performance by accident.

However, with Build Agent, you have access to create the full set of artifacts available on the ServiceNow AI Platform. With this greater range of creation abilities, the necessity of reviewing your development output becomes of even greater importance and impact to your instance performance and security. So you must shift to owning the review step when migrating applications to Build Agent.

Before accepting any Build Agent output, read the generated code and description. Treat the output as a strong first draft that might not contain specific business rules, data volumes, or performance requirements for your organization.

The core workflow of prompting, reviewing, and accepting or rejecting is consistent across every Build Agent session. For more information, see Build Agent workflow.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Iteration over prompt perfection

When creating apps and files in AES, modifying selections that you made during the creation process could sometimes prove to be difficult or require restarting the process entirely. But with Build Agent, the agent supports iteration by default. From prompt to prompt, Build Agent can process corrections and adjust the output accordingly. You can also create checkpoints with Build Agent, which help you distinguish one artifact version from another across sessions. For more information, see Checkpoints and conversation change log.

Prioritize iteration over a perfectly articulated prompt when developing with Build Agent. Attempting to write a single, catch-all prompt that describes an entire application and all of its artifacts rarely produces better results than the iterative approach. If needed, you can also revert your changes with Build Agent. For more information, see Revert app changes with Build Agent.

Session continuity

Development in AES is stateless, meaning that each app a file you create is discrete and independent of the existence of other files. AES also doesn't store data about your work from one session to the next.

Build Agent accumulates information about your data model, business rules, and design decisions during each session. Sessions are then stored, enabling you to choose between rejoining an earlier session or starting a new one.

Learning how to structure your work between and across sessions is helpful when developing with Build Agent. Determine whether your work would benefit from the accumulated context of an existing session. For less complex builds or single artifact extensions, you can start a new session. For more information about sessions and limitations, see Build Agent limitations.

Build Agent as a development partner

AES is a development tool. When you're ready to build an application or file, you enter the AES development environment and create or modify app files.

With Build Agent, you can use the chat panel as both a development tool and a design/planning partner. For example, you can use Build Agent to:

- Brainstorm application architecture before writing any code
- Analyze an existing AES application and identify where scripting or additional artifacts would improve it
- Generate README documentation for an application
- Add comments to existing scripts
- Refine requirements before committing to a data model
- Identify scope conflicts in applications that originated globally

These activities don't produce deployable artifacts on their own, but they can help to reduce the uncertainty that leads to rework. Using Build Agent to think through a build before prompting can result in higher quality output.

Related topics

Build Agent use cases

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

App Engine Studio and Build Agent feature comparison

App Engine Studio (AES) and Build Agent differ across interface, artifact scope, scripting access, application scope, and governance. Use this feature comparison to understand what changes when you migrate to Build Agent.

AES and Build Agent feature comparison

	App Engine Studio (AES)	Build Agent in ServiceNow Studio
Primary interface	Guided wizard interface	Conversational prompt interface inside ServiceNow Studio
Developer persona	Citizen and low-code developers	All developer experience levels, from citizen to pro-code
Artifact types	Tables, forms, flows, and workspaces within wizard constraints	Full platform set of artifacts: business rules, Script Includes, Client Scripts, flows, ACLs, UI Builder pages, packaging, and README
Script access	Not directly available. Requires converting to ServiceNow Fluent and opening app in an integrated development environment (IDE)	Built-in script writing and reading
Application scope	Scoped applications only	Scoped and global applications
Existing application modification	Applications you created only	Any application, including base system applications
Session context	Stateless	Persistent within a session and accumulates over the course of a session
External integrations at build time	None	Model Context Protocol Server Console (MCP) client connections. For a full list of MCP connections, see MCP connections and Build Agent
Governance	App Engine Management Center (AEMC) optional deployment path	Several deployment options, including App Engine Management Center (AEMC). SeeDeploying what you built with Build Agent
AI capability embedding	Manual post-build step	In-app AI agents and skills can be embedded in workflows at build time

What does not change

The following remain consistent across both tools:

- App files built in AES are already accessible in ServiceNow Studio. No re-creation or export is required.
- Workflow Studio flows created in AES are the same records in ServiceNow Studio.

Use App Engine instead of customization

App Engine development tools, such as App Engine Studio, offer an excellent alternative to customizing existing applications on the ServiceNow AI Platform.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

When your company needs to add new functionality to the ServiceNow AI Platform, you can customize existing applications, such as IT Service Management (ITSM), or create a new application using App Engine developer products, such as App Engine Studio, Creator Studio, or ServiceNow Studio. A simple guideline for which path to choose is:

- If the customization extends the intended purpose of the application, it works better to customize. For example, you can add IT functionality to ITSM.
- If the customization doesn't extend the intended purpose of the application, it works better to create a new application using App Engine developer products. For example, do not repurpose the ITSM workflow to add a travel request workflow.

Examples of when to use App Engine

ServiceNow products work best when they're used as they were intended. If you find yourself heavily customizing an application to repurpose it, a better plan is to create a new application using App Engine developer products.

The following scenarios demonstrate where creating a new application works better than heavily customizing an existing ServiceNow application:

- Your company has a business process that augments existing product functionality but doesn't follow the same workflow exactly.
- You have a novel use case for an app that doesn't align with any product workflow.
- You have a use case that could be built by heavily customizing an out-of-the-box application, but it doesn't align with what the existing application was intended to do.

Let's dive deeper into the last use case.

Issues with repurposing existing products

ServiceNow applications come with roles, processes, and flows that are specially tailored to their use case. For example, ITSM apps help with IT users, IT issues, IT reports, and IT cases.

You might have an idea for an app that's similar to but doesn't exactly align with ITSM. Because ITSM gives you a starting point, you might be tempted to customize ITSM to add the new functionality. For example, ITSM tracks IT issues, and a travel app you want to create might track travel requests. While the workflows sound similar, in fact, they use very different data, different user interfaces, and the details of each workflow vary greatly. Rather than heavily customize ITSM to repurpose it, a better plan is to use App Engine developer products for the following reasons:

- Combining two workflows creates conflicts.
- Customizing applications has implications.

Combining two workflows creates conflicts

In the ITSM example, the repurposing of ITSM to include a travel workflow uses different data, different tables, different roles, and different workflows than ITSM. As ITSM, ITSM customizations, and the travel workflow grow over time:

- Their features will continue to diverge.
- Adding new functionality or fixing problems in one workflow might adversely impact the other.
- The performance of ITSM may suffer.
- The code base will grow and the two purposes of ITSM will make troubleshooting more difficult.
- Quality engineers will require two different testing frameworks.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

All these issues may cause unnecessary complications, poorer performance, upgrade delays, and software problems.

Customizing applications has implications

The ServiceNow AI Platform is built to embrace customization and configuration. The ServiceNow AI Platform is flexible enough to fit your company's business needs. How you customize ServiceNow applications, however, can have significant impacts on ServiceNow support, upgrading to future ServiceNow AI Platform versions, and the functionality of the platform.

Let's start by differentiating customization and configuration:

- Customization is any change made to the code that is part of the baseline installation of a ServiceNow instance. You use code to customize applications.
- Configuration is any change you make to the behavior of a product that does not touch the code in the baseline installation of a ServiceNow instance. You can use system properties, ServiceNow products, or code to configure an application.

The following are some of the implications that result from customizing applications:

- If you add code to an application, you own it whether or not it modifies the code in the baseline installation on a ServiceNow instance.
- The platform marks all customizations and skips them when you update to a new version of the platform. That means you are responsible for manually updating the customizations. This can have a significant impact on the time and resources required to update to new platform versions.
- The ServiceNow AI Platform uses a framework that supports applications in how they process tasks, how forms are rendered in multiple browsers, and the overall user experience. Introducing customizations can have unintended consequences on this framework.
- You own the burden of testing custom code and determining if it impacts platform functionality.
- ServiceNow Customer Support cannot troubleshoot custom code or issues caused by custom code.

Customization is one of the key features of the ServiceNow AI Platform. However, over-customizing an application to repurpose it is likely to generate technical debt, lengthen your upgrade cycle, and complicate future platform upgrades because the custom code may not easily migrate to new platform versions.

Conclusion

Customization and configuration are hallmarks of the ServiceNow AI Platform that enable your company to customize workflows to fit its specific needs. Proceed with these tasks in the following order:

1. Configure ServiceNow applications as much as you can before customizing them.
2. Customize an application only when it extends the intent of the application.
3. Use App Engine developer products, such as App Engine Studio, Creator Studio, and ServiceNow Studio, to create applications rather than customizing an application to create functionality that doesn't align with its original purpose.

Customization vs. configuration with App Engine Studio

There are important differences between customizing and configuring ServiceNow applications. The ServiceNow platform is built to embrace customization and configuration, but how you do so can have significant impacts on ServiceNow support, upgrading to future ServiceNow platform versions, and the functionality of the ServiceNow platform.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The general rules around customization are:

- Customize an application only if it extends the original intent of the application. For example, add IT functionality to ITSM but don't add a travel workflow. Instead of over-customizing an application, create a new application using App Engine products, such as App Engine Studio, Creator Studio, or ServiceNow Studio.
- Configure as much as you can before customizing an application.
- If you add code or make other modifications to out-of-the-box functionality, you own them.

What is configuration

Configuration is the process of using ServiceNow built-in tools and features to modify an application's behavior without making changes to flows or the code that is part of the baseline installation on a ServiceNow instance.

Configuration can take the form of using ServiceNow built-in tools to add tables and more, setting instance-wide parameters, as well as using code to extend an application's functionality to meet business needs as long as the code does not modify the baseline code installation. The entire platform is designed for you to add configuration code.

If you add code, such as workflow scripts, you own it, even if it doesn't alter the baseline code installation. That includes owning the impact it has on the entire ServiceNow platform. Issues that arise from added code are beyond the scope of ServiceNow support to debug.

Reverting a configuration should not require any change to the baseline code.

Configuration examples include:

- Forms: Configure tables, fields, data types, default values, and field dependencies to configure the data you capture and display.
- UI elements: Modify layouts, add related lists, add buttons, and change field names.
- Service Catalog: Configure portals where your customers can request catalog items such as service and product offerings.
- ACLs: Restrict unauthorized users from accessing forms and data.
- System property values: Modify the application's experience for all users.

What is customization

Customization is any change made to the flows or the code that is part of the baseline installation on a ServiceNow instance. You use ServiceNow products or code to customize applications.

If you add code, you own it, even if it doesn't alter the baseline installation. That includes owning the impact it has on the entire ServiceNow platform.

Customization examples include:

- Scripting: Customize ServiceNow through scripting using JavaScript. This includes creating client scripts, server-side scripts, and business rules with intricate logic that modify the baseline code.
- Custom tables: Develop custom tables to accommodate specialized data that doesn't fit within standard tables.
- Integration: Customize integration with external systems, such as APIs and web services, for seamless data exchange.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Widgets and portals: Create custom widgets and portals to provide unique features and user experiences.
- Workflows: Create and modify workflows using Workflow Studio. Create and manage playbooks, flows, actions, decision tables, and integrations from one design environment to automate tasks. Upgrading to a new version of a flow requires reapplying your customizations.

Tools for customization and configuration

ServiceNow offers many tools and features, such as business rules, to modify the out-of-the-box behavior of ServiceNow applications. Whether they customize or configure an application depends on how they're used. Using these tools to modify the installed code base constitutes customization. Using these tools to add code that does not modify the flows, or the installed code base constitutes configuration. In both cases, you own the code you add as well as the impact it has on the ServiceNow platform.

ServiceNow tools include:

- UI Policies: Dynamically modify the visibility of fields and attributes on a form according to user inputs.
- Business rules: Automatically trigger actions based on specified conditions.
- UI Actions: Extend and customize forms and lists by adding buttons, context menu items, or other UI elements that perform specific actions when clicked.
- Client-side scripts: Scripts that execute within the user's browser when certain actions occur on a form or a UI page.
- Server-side scripts: Scripts that execute on the ServiceNow server or database, for example, to update record fields when a database query runs.

What is personalization

Personalization is when users use out-of-the-box application tools to modify an application's look and feel only for themselves. Admins can change the look and feel for all users and that is considered configuration. Personalization examples include a user choosing to use dark mode or choosing which table columns to display.

Personalization does not change the baseline code installation on a ServiceNow instance. So, personalization does not impact customer support or interfere with upgrades to new ServiceNow versions.

Ramifications of customizing ServiceNow products

The ServiceNow platform is extremely flexible and built to embrace customization and configuration to fulfill a wide range of business requirements. How you customize ServiceNow applications, however, can have significant impacts on ServiceNow support, upgrading to future ServiceNow platform versions, and the functionality of the platform. Instead of customizing ServiceNow applications, consider using App Engine development products, such as Creator Studio and ServiceNow Studio to create new applications.

The ServiceNow platform uses a framework that supports applications in how they process tasks, how forms are rendered in multiple browsers, and the overall user experience. ServiceNow relies on the framework's integrity to develop and provide support in a consistent manner. Customizations can impair this framework, change platform functionality, and impair workflows and upgradeability.

Customizations trigger the platform to create sys_update_xml records, which are stored in the Customer Update table. The platform marks all customizations and skips the customized records

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

when you update to a new version of the ServiceNow platform. That means you are responsible for manually updating the customizations. This can have a significant impact on the time and resources required to update to new platform versions.

Note: The Customer Update table also contains modifications or additions to configuration metadata, for example, creating a new catalog item or a new workflow.

For more information, see the Customer Updates table. Note that the complexity of maintaining customizations dramatically increases as the number of customizations increases.

Customizing the installed code base can be costly, generate technical debt, lengthen your upgrade cycle, and complicate future platform upgrades because the custom code may not easily migrate to new platform versions. Custom code can change the standard functionality of the ServiceNow platform in unintended ways. Evaluate demands for customization carefully and only resort to customization where there is clear business value and no alternative. Wherever possible, avoid customization by using configuration instead.

If you customize a product:

- You are responsible for maintaining the customization going forward.
- Customer Service and Support will not support problems caused by custom code. If it is the cause of the problems, the support team will likely advise you to revert to the out-of-the-box code.

What is the Customer Service and Support stance on supporting customization

The ServiceNow Customer Service and Support stance on customization is if you add code, you own it and its consequences. Why? Customer Support is not privy to your custom business logic, doesn't know what the expected behavior should be, is unable to reproduce the issue on an out-of-the-box instance, and customer support engineers are not certified implementation specialists, so they are not certified to review customized code logic.

Alternatives to customization

If you have requirements and ideas for enhancements, instead of customizing the installed code base, you can:

- Use configuration instead of customization.
- Submit an Enhancement Request to the ServiceNow development team. Each request is evaluated and, if approved, will be incorporated into a future release.
- Create an app using App Engine developer products to handle desired functionality.

When to use App Engine developer products instead of customizing

When your company needs to add new functionality to the ServiceNow platform, you can customize existing applications, such as ITSM, or create a new application using App Engine developer products, such as Creator Studio or ServiceNow Studio. A simple guideline for which path to choose is:

- If the customization extends the intended purpose of the application, it works better to customize. For example, you can add IT functionality to ITSM.
- If the customization doesn't extend the intended purpose of the application, it works better to create a new application using App Engine developer products. For example, do not repurpose the ITSM workflow to create a travel request workflow.

For example, ITSM is designed to handle IT issues. To customize it to handle travel requests goes beyond the original intention of ITSM. Because IT and travel requests have different workflows,

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

it's better to create a travel request app using App Engine developer tools, such as Creator Studio and ServiceNow Studio, instead of customizing ITSM.

For more information, see Build apps using App Engine Studio.

Examples of when to use App Engine developer products

ServiceNow products work best when they're used as they were intended. If you find yourself heavily customizing an application to repurpose it, a better plan is to create a new application using App Engine developer products.

The following scenarios demonstrate where creating a new application works better than heavily customizing an existing ServiceNow application:

- You have a novel use case for an app that doesn't align with any product workflow.
- You have a use case that could be built by heavily customizing an out-of-the-box application, but it doesn't align with what the existing application was intended to do.
- Your company has a user group or business process that should be separate from the OOTB product workflow.

Guidelines for customizing ServiceNow products

If you must make a customization, consider the following suggestions:

- Maximize configuration options first.
- Avoid copying objects. Instead, update objects in place wherever possible, except for Service Portal widgets and other items designated to be reused.
- Default to "add before edit." This means that you should, for example, add fields to forms rather than change the type of an existing field. Avoid using the same names as out-of-the-box objects, methods, or classes when adding them.
- Minimize the number of fields you add to a form. The more fields you have on a form, the longer it may take to load.
- Export original records as backups before customizing them. Keep track of their out-of-the-box sys_id's in case you must restore them in the future.
- Use scoped applications as your default for any new custom development.
- Document all customizations. Add comments explaining why you customized (including business justification). Review all comments before upgrading to determine if you can revert to out-of-the-box code.
- Create tests for all customizations. Write Automated Test Framework (ATF) tests for all customizations where possible.
- Use HealthScan regularly to identify unnecessary customizations.
- Customizations should be made to baseline objects where necessary so that conflict resolution and decision-making can be appropriately recorded in the updates. Hidden customizations may cause administrators to overlook updates in future assessments in case reverts or merges are necessary.
- Test your customization for all use cases. Include performance testing and the introduction of unintended consequences.
- Administrators are responsible for verifying that their customizations work after a ServiceNow platform upgrade, and for keeping track of what customizations are made.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Handling customizations when you upgrade

Customizations trigger the platform to create sys_update_xml records, which are stored in the Customer Updates table. These records are not updated during platform version upgrades. ServiceNow marks them as skipped records in the ServiceNow Upgrade Monitor. To make sure they're successfully ported to the upgraded instance, you must manually process the skipped changes. For more information, see the Customer Updates table.

Assuming you've documented all your customizations—including the business justification—take your documented inventory and compare it with the skipped records identified in the Upgrade Monitor. After filtering out low-risk changes that have resulted in skipped records (for example, field labels or form layouts), you'll need to decide whether to:

- Retain each customization
- Revert to out-of-the-box
- Merge your customization with the base system to resolve conflicts

Integrated development tools for AES

App Engine Studio (AES) provides instant access to several tools available in the ServiceNow AI Platform development suite.

As you create apps using App Engine Studio, you can extend their capabilities with these powerful development tools.

App Engine Studio development tools and features

Tool	Description	Reference
Access Control	Work with access control and roles for the application.	See Add application security.
Catalog Builder	Add record producer catalog items in a guided tool for building a Service Catalog.	See Add a standard catalog item.
Decision tables in Workflow Studio	Add decision rules to decouple decision logic from code using decision tables.	See Add a decision.
Email notifications	Create and edit email notifications that are automatically sent when a record is created or updated. Email notifications are also used as a reference in flows or can be triggered by an event.	See Add an email notification.
Zero Copy Connector for ERP	Build an ERP (Enterprise Resource Planning) data model and extract data from your ERP system.	See Zero Copy Connector for ERP.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

App Engine Studio development tools and features (continued)

Tool	Description	Reference
Flows in Workflow Studio	Add flows to automate application processes.	See Add a flow from scratch.
Integration Hub - Import	Import and map data into existing tables.	See Create a data integration.
Mobile App Builder	Add mobile experiences.	See Add a mobile experience.
Process Automation Designer	Edit cross-functional processes and consolidate them into automated task-oriented views.	See Edit a process.
Table Builder	Work with data models in a tabular format. Table Builder is exclusive to App Engine Studio and necessary for creating apps.	See Create a blank table.
UI Builder	Edit workspace and portal experiences using a highly configurable UI development tool.	See Add a workspace and Add a portal.
Workspace Builder	Quickly build custom workspaces to give users a service desk-like experience to manage tickets.	See the following topics: • Using Workspace Builder • Edit a workspace in Workspace Builder • Configure workspace settings in Workspace Builder • Customize a workspace home page in Workspace Builder • Add a record page for a workspace in Workspace Builder • Create lists for a workspace in Workspace Builder • Enable analytics for a workspace in Workspace Builder

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Get help with App Engine Studio

To get help with App Engine Studio, your ServiceNow instance, plugins, permissions, and more, watch a short video to contact the ServiceNow admin who works in your company.

Get help now from your Now Support administrator

The ServiceNow admin in your company can help you with things like activating plugins, kicking off upgrades, unlocking accounts, updating user permissions, and much more.

https://player.vimeo.com/video/1056112088?h=1a5cd3b245&badge=0&autopause=0&player_id=0&app_id=58479

Helpful resources

Some ServiceNow resources that can provide you with helpful information are:

Video

Watch Build apps fast with App Engine Studio.

ServiceNow Community

App Engine Studio Community

App Engine and App Engine Studio

Learn about the differences between App Engine and App Engine Studio.

App Engine is a ServiceNow AI Platform® suite of products to help boost productivity by automating workflows. ServiceNow® App Engine Studio (AES) is a development tool for creators of varying skill levels to build applications that meet the immediate needs of your organization.

For more information, see App Engine products and offerings.

How App Engine and App Engine Studio work together

App Engine provides the ServiceNow AI Platform infrastructure, governance, and application life cycle management capabilities. App Engine Studio is the development environment you use to build applications within that platform. App Engine serves as the platform infrastructure while App Engine Studio provides the development tools.

App Engine	App Engine Studio
Platform suite and entitlement	Development environment within App Engine
Manages deployment, governance, and application lifecycle	Enables you to design, build, and test applications
Platform administrators manage this environment	Developers and citizen developers use this environment

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Additional information about App Engine

Learn more about App Engine	Additional ServiceNow resources
ServiceNow provides several additional resources on App Engine.	App Engine product page App Engine Enterprise data sheet (PDF) Entitlements for subscription products Custom table guide (PDF)

Learning plan for getting started with App Engine Studio

ServiceNow provides citizen developers with several quick training modules for getting started building low-code apps in App Engine Studio (AES).

The ServiceNow Developers site provides numerous resources for developing apps in AES.

Use the following learning modules to quickly explore ways to build apps.

ServiceNow Developers site learning modules

Course	Description
Build My First App Engine Studio Application	Create an application to track a collection using AES. The application will be secure, automated, and provide an engaging user experience.
Create an App and Data Model	Create an application from scratch with AES, including the data model used by the application. The data model is a foundation for learning how to add user experiences, logic and automation, and security to your application in other learning modules.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ServiceNow Developers site learning modules (continued)

Course	Description
Create Record Producer User Experiences	Create record producer catalog items in AES. Users can submit a request or create a task using Service Catalog items in Service Portal.
Create Workspace User Experiences	Create a workspace to report on and manage data in an application created with AES.
Create Mobile User Experiences	Create and test a mobile experience in AES that enable users to access applications from anywhere.
Automate Processes with Flows	Create flows in AES to automate business processes. Flows reduce human error, improve efficiency, and make apps work better.
Secure Apps and Data	Use AES to secure an application by creating roles, assigning the roles to users, and testing user access to application data.

Additional learning resources

Learn more about application development
You can learn about How citizen developers cook up innovation on the ServiceNow blog.
For more information about defining cross-scope access to an application resource and approving or denying requests, see Define cross-scope access to an application resource.

Configuring App Engine Studio and related apps

App Engine Studio (AES) configuration involves installing the application and completing guided setup, which walks you through the implementation process, one step at a time. After you finish each setup procedure, you're prompted to begin the next one.

Configuring App Engine Studio

Configure App Engine Studio (AES) by installing the application on your development instance (and any other non-production instances on which users will develop applications) and completing guided setup. See the following resources for more information:

- Installing App Engine Studio
- Configure App Engine Studio

Note: If you plan on cloning your production instance to one or more non-production instances, you should also install the AES product on your production instance prior to cloning. For more information, see Instance Clone and Cloning instances with AES.

Configuring related applications

After you finish configuring AES, you have the option to configure other applications related to AES. Each of these applications has its own guided setup. If you have already configured one or more of the applications and you're prompted to configure it again, you can simply ignore that instruction. For more information about the applications related to AES and configuring them, see the following list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Related applications

App Engine Management Center (AEMC)

App Engine Management Center AEMC contains all of the features that you need to manage application development within your organization, from idea submission to deployment. Using AEMC, admins can manage requests for new apps and delegate development and collaboration permissions. Admins can also oversee the entire deployment process with insights into individual deployment requests, test suites, and scheduled deployments.

Application Intake

Application Intake enables citizen developers to submit ideas for applications. Application Intake requests can be viewed in AEMC, where admins can approve or reject the requests.

Pipelines and Deployments

Pipelines and Deployments enables release managers and App Engine admins to build custom pipelines across multiple instances. You can also connect the testing suites and frameworks that your organization uses to verify that applications are stable and compatible before they reach production.

ReleaseOps

ReleaseOps improves the existing pipelines deployment process by supporting the deployment of update sets, leveraging the automation capabilities of ServiceNow Playbooks, and enabling both scheduled and on-demand releases.

AES and domain separation

Domain separation is unsupported for App Engine Studio (AES). Domain separation enables you to separate data, processes, and administrative tasks into logical groupings called domains. You can control several aspects of this separation, including which users can see and access data.

Support level: No support

- The domain field may exist on data tables but there is no business logic to manage the data.
- This level is not considered domain-separated.

For more information on support levels, see Application support for domain separation.

Related topics

Domain separation for service providers

Installing App Engine Studio

You can install the App Engine Studio (AES) application (com.snc.app-engine-studio) if you have the admin role. The application installs related ServiceNow® Store applications and plugins if they are not already installed.

Trial version of AES

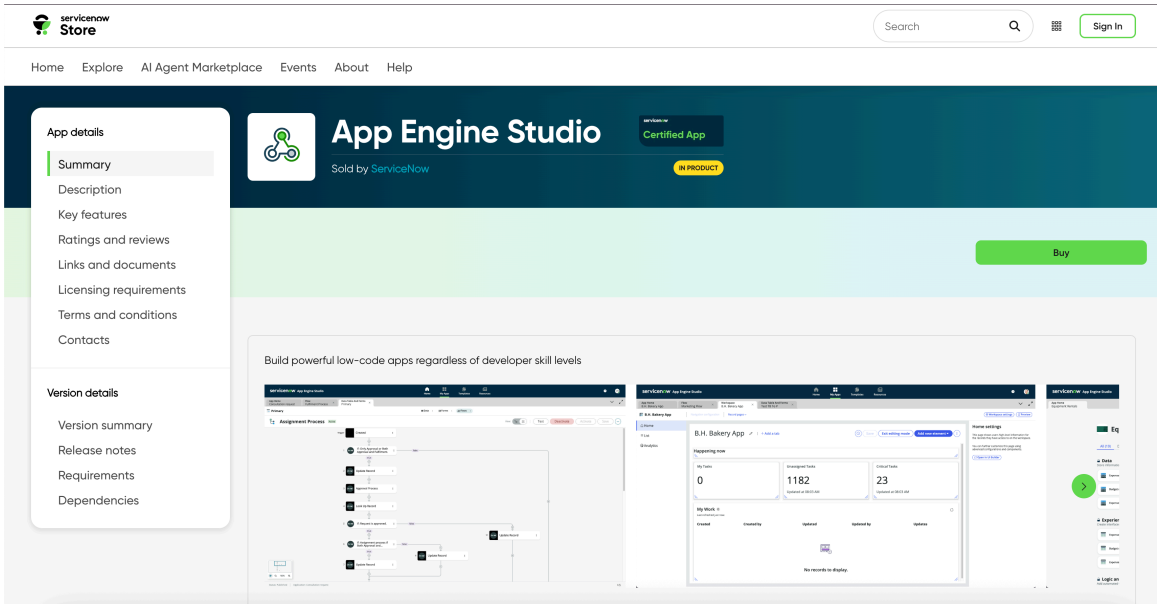
To try AES before purchasing a license, you can install it on a personal developer instance (PDI). For more information about PDIs, see Personal developer instance guide. To obtain a PDI, see the ServiceNow Developer Site.

Note: An App Engine Studio subscription is required to install on a production or non-production instance. For more information, contact your ServiceNow representative.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Using the ServiceNow Store to install AES

The ServiceNow Store enables you to download core products and applications. A product contains one or more applications that are licensed as a group. For example, App Engine Studio is a product that contains numerous tools and applications within it, such as Table Builder and Mobile App Builder. If you install the basic AES application and not the full AES product, you won't get its full suite of features and tools.



Check that the application and all of its associated ServiceNow Store applications have valid ServiceNow entitlements. For more information, see Get entitlement for AES apps.

When you install AES, you also install the following related items:

- Roles and groups
- Tables
- Plugins

For more information, see Components installed with AES.

Installation workflow for AES

The following is an overview of the tasks that you must perform to install AES for the first time:

1. Develop your instance strategy for AES.
2. Get entitlement for AES to ensure you have a license.
3. Install AES (the full product, not the application) from the ServiceNow Store. App Engine Studio is a product that contains numerous tools and applications within it, such as Table Builder and Mobile App Builder. If you install the basic AES application and not the full AES product, you won't get its full suite of features and tools.
4. Install the AES integrations and plugins.

Alternatively, you can update existing AES installations by doing the following:

- Update a previously installed AES version
- Upgrade your AES instance to the next family release

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Learn more about AES installation	Additional ServiceNow resources
ServiceNow provides several additional resources on installing and administering App Engine Studio.	 ServiceNow Community site
	 ServiceNow University Introduction to App Engine Studio for Citizen Developers course

AES instance strategy

You should install App Engine Studio (AES) on all ServiceNow instances where users will develop applications.

You will need to establish your company's instance strategy for AES. Some of the decisions you will need to make include:

- Whether you want to allow anyone within your company to have access to AES to start building apps.
- Whether you want to grant access to a select group of people.
- Whether you want to grant access on a case-by-case basis by setting up a form where individuals can complete information about the app that they are looking to build. IT then decides whether to give those individuals access to build that app.

A non-production instance that is similarly configured to your production instance may be the best candidate for your test environment. You can then more accurately find issues that may arise if the application is deployed to production.

Note: If you plan on cloning your production instance to one or more non-production instances, you should also install the AES product on your production instance prior to cloning. For more information, see Instance Clone and Cloning instances with AES.

After you have established your instance strategy, you must also establish and automate your approval or review process. AES is a product that runs on your non-production environment. For an organization with multiple non-production environments, you will need to decide which non-production environment AES will run on. You must also determine which pipeline to use for promoting apps from a particular non-production instance to your test instance, and then finally to production where the app will be running live. For more information, see Pipelines and Deployments.

AES and the ServiceNow Store

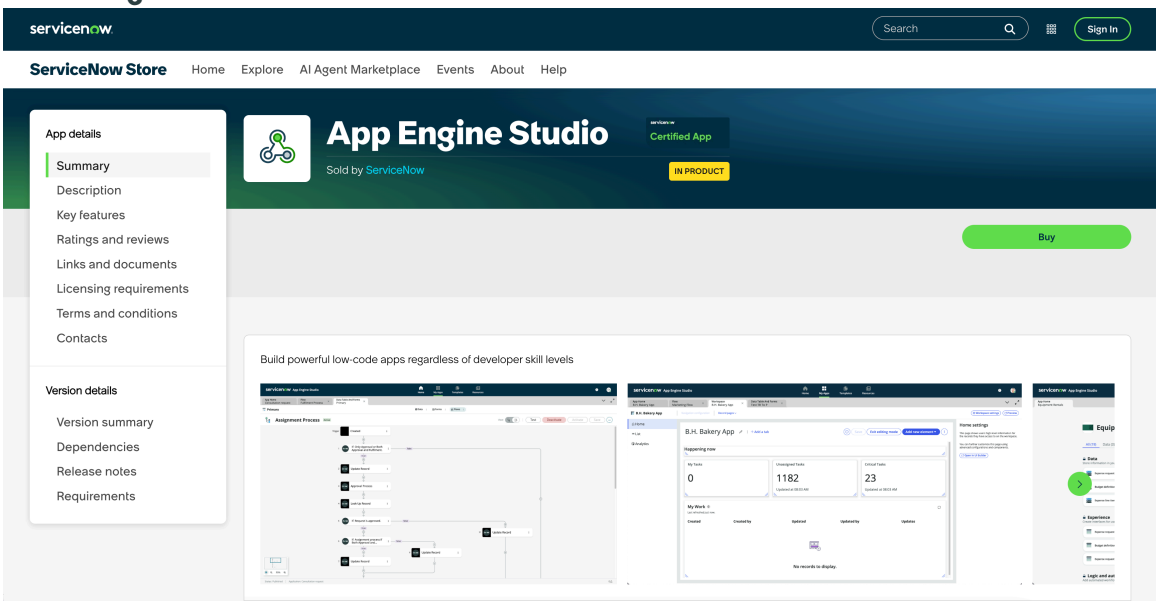
App Engine Studio (AES) enables you to obtain new and updated features more rapidly. Before you can use AES, you must verify that you have entitlement to it, meaning that you have valid licenses to use it. Then, you can install AES and its plugins from the ServiceNow Store.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Request apps on the Store

Visit the ServiceNow Store to view all the available apps, and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the ServiceNow Store version history release notes.

AES listing in the ServiceNow Store



Methods for installing AES

The process you use to obtain AES depends on whether you're installing AES for the first time, updating a version of AES that you previously installed from the ServiceNow Store, or upgrading from one family release to the next one.

Get entitlement for AES apps

Install App Engine Studio (AES) by verifying that the application or the product and its associated applications have valid ServiceNow entitlements. An entitlement means you have a license to install the product.

This video shows you how to perform the following procedure.

Before you begin

Note: An App Engine Studio subscription is required to install on a production or non-production instance. For more information, contact your ServiceNow representative.

Individuals performing entitlement must have a Now Support account and have permission to request applications for the instances under consideration.

Role required: admin

Procedure

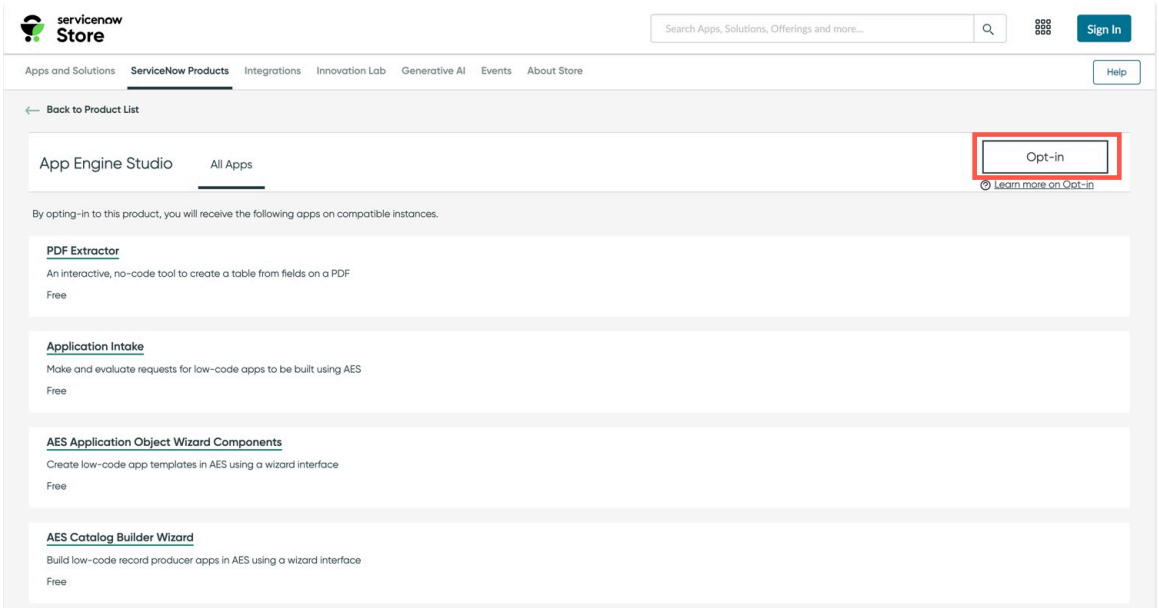
1. Navigate to the ServiceNow Store.
2. Select **Sign In** and log in using your Now Support credentials.

   Integrations and other types of content appear on the **Certified Apps** tab.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

3. Select the **ServiceNow Products** tab to view all available ServiceNow products.
4. Select **App Engine Studio**.

   Note: When you have obtained entitlement to the core product, the applications listed under it are also entitled.



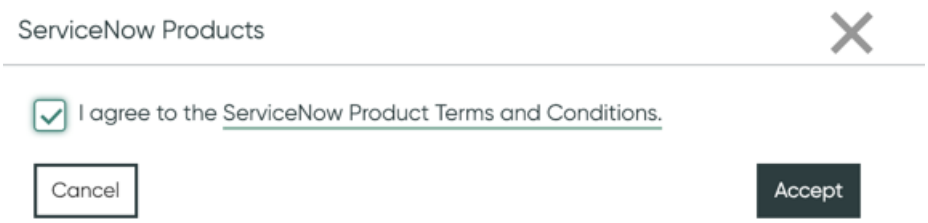
   The page lists all the applications that are associated with the AES product you're entitling.

5. Verify that you have entitlement to the product and applications by selecting **Opt-in**.

   Note: The **Opt-in** button is not available unless you purchased an AES subscription.

6. When the ServiceNow Store prompts you to read and accept the ServiceNow terms and conditions, select the check box and select **Accept**.

   Opting in to entitlement is a one-time process for each product.



   A check mark appears next to the product name, and the **Manage Entitlement** button appears. The button indicates that you are subscribed to the product and its associated applications.

7. Change the instances affected by the entitled applications by selecting the **Manage Entitlement** button and then specifying one of the following options:
   - **Remove all existing entitlements**
   - **Entitle all instances**
   - **Entitle selected instances**

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Result
Install and activate AES. Navigate to **All** > **My Company Applications** and select App Engine Studio.

Install AES from the ServiceNow Store for the first time

Install App Engine Studio (AES) from the ServiceNow Store for the first time. The installation involves several easy steps. Some steps are performed on the ServiceNow Store and some in your ServiceNow AI Platform instance.

Before you begin
Role required: admin

About this task
The ServiceNow Store enables you to download core products and applications. A product contains one or more applications that are licensed as a group. For example, App Engine Studio is a product that contains numerous tools and applications within it, such as Table Builder and Mobile App Builder. If you install the basic AES application and not the full AES product, you won't get its full suite of features and tools.

For a list of all dependencies and tools installed with the AES product, see the listing in the ServiceNow Store.

Procedure
1. In the ServiceNow Store, ensure that you have entitlements (or licenses) to the product and its dependent applications.
2. Install AES from the ServiceNow Store.
3. Run the installation on your instance, and accept all the plugins.

Install the AES product and integrations

Install the App Engine Studio (AES) product to ensure you get all the dependencies, integrations, and tools, such as templates and builders.

Before you begin
Store installation requires a Now Support account. You must also have permission to request applications for the instances you're supporting.

Role required: admin

Procedure
1. Navigate to **All** > **System Applications** > **All Available Applications** > **All**.
2. Locate the App Engine Studio product (not the application), and select the product tile.

   Visit the ServiceNow Store to view all the available apps, and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the ServiceNow Store version history release notes.

   Note: You must select the AES product, not the application.

3. On the App Engine Studio product page, select **Install**.
4. In the Review Installation Details modal, review the list of applications and dependencies to install.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Dependent plugins and applications are listed if they will be installed, are currently installed, or need to be installed. If any plugins or applications must be installed, you must install them before you can install App Engine Studio.

5. **Optional:** If demo data is available and you want to install it, select the **Load demo data** check box.
   Demo data are the sample records that describe application features for common use cases. Load the demo data when you first install the application on a development or test instance.
6. Select **Install**.

Result
App Engine Studio is installed on your instance.

Update a previously installed AES version

If you previously installed App Engine Studio (AES) from the ServiceNow Store and if a new version is available, then update the version in your instance.

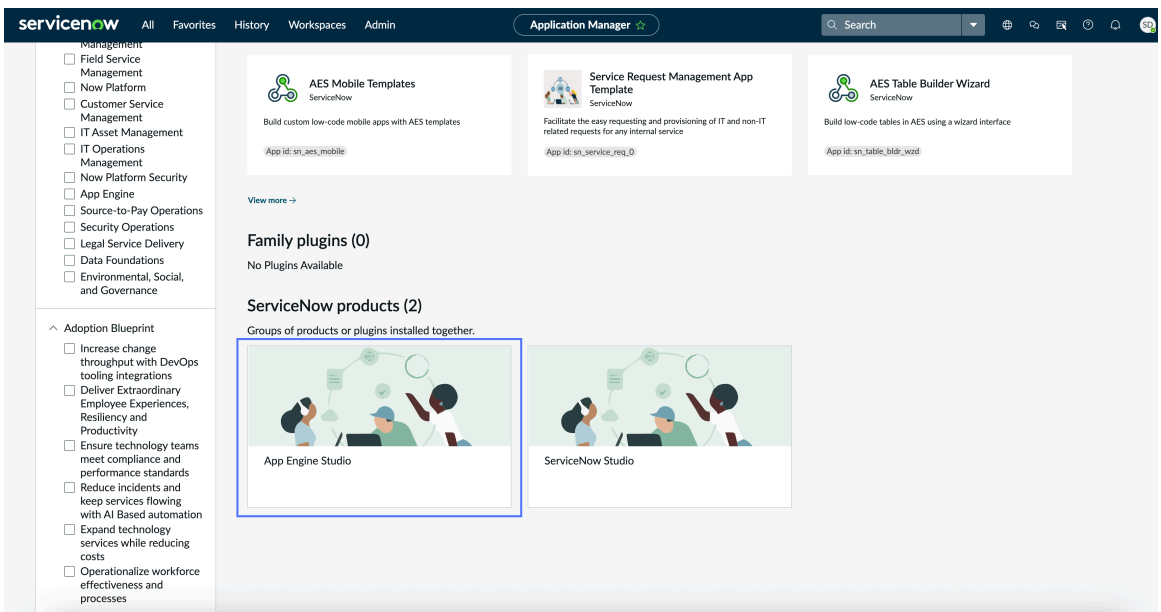
Before you begin

Note: There should be no changes to the underlying applications or code bases when updating AES.

Role required: admin

Procedure

1. Navigate to **All** > **System Applications** > **All Available Applications** > **All**.
2. Search for `App Engine Studio`.



3. Select the **Proceed to update** button.
4. In the Review Installation Details modal, locate the select the App Engine Studio application and select the version that you want to update to.
5. Select **Proceed**.

Result
App Engine Studio automatically updates on the instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Upgrade your AES instance to the next family release

If you are currently running a family release (Brazil, for example) and want to upgrade App Engine Studio (AES) to the next release, you don't need to acquire the applications from the ServiceNow Store.

The application is automatically updated when the platform is updated to the minimum required version.

Components installed with AES

Several types of components are installed with activation of the App Engine Studio (AES) plugin, including tables and user roles.

Note: The Application Files table lists the components that are installed with this application. For instructions on how to access this table, see Find components installed with an application.

Roles installed

Role title [name]	Description	Contains roles
App Engine Studio admin [app_engine_admin]	Administers AES, including users, app configuration, and other management tasks. This role is assigned automatically to users in the App Engine Studio Admins group.	• sn_request_read • sn_request_write • sn_app_eng_studio.user
App Engine Studio app template admin [app_template_admin]	Administers the use, sharing, and activation or deactivation of templates. This role is assigned by an admin to individual users. For more information, see Manage template access.	• app-template-author • app-template-runner • flow-designer • flow-operator
App Engine Studio user [sn_app_eng_studio.user]	Builds applications in App Engine Studio. This role is assigned automatically to users in the App Engine Studio Users group. For more information, see Grant user access to AES.	• catalog_builder_editor • app_template_runner • sn_g_app_creator.app_creator

Tables installed

Table	Description
App Details	Details about the operations that a developer used to create an application in App Engine

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Table	Description
[sn_app_eng_studio_app_details]	Studio. This table is updated automatically as developers build applications in App Engine Studio.
Applications in Projects [sn_app_eng_studio_project_application_m2m]	Applications that a developer creates in App Engine Studio. This table is updated automatically as developers build applications in App Engine Studio.
Environment [sn_pipeline_environment]	Instances that you use for the developing, testing, or launching an application. You update this table as you define environments for App Engine Studio.
Deployment Request [sn_deploy_pipeline_deployment_request]	Requests to review and publish an application that a developer created in App Engine Studio. From the deployment request form, a reviewer can deploy the application to different environments, accept or reject an application, and send feedback to a developer. For more information, see Deployment Request form in the Pipelines and Deployments app.
Pipeline [sn_pipeline_pipeline]	Configurations for deploying applications to different environments. There can be only one active pipeline at a time. You update this table as you create a pipeline for the deployment of applications from App Engine Studio. For more information, see Managing deployments using Pipelines and Deployments.
Project [sn_app_eng_studio_project]	Details about App Engine Studio development sessions. A project is created automatically when a developer creates an application in App Engine Studio. This table is updated automatically as developers build applications in App Engine Studio.
Resources Content [sn_app_eng_studio_resources_content]	Help resources that a developer can access in App Engine Studio. This table includes configurations to support the default user experience for App Engine Studio.
Resources Content Topic [sn_app_eng_studio_resources_content_topic]	Categorizations of help resources in App Engine Studio. This table includes configurations to support the default user experience for App Engine Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Table	Description
Taxonomy [sn_app_eng_studio_taxonomy]	Application files that a developer creates in App Engine Studio. This table is updated automatically as developers build applications in App Engine Studio.
Taxonomy Category [sn_app_eng_studio_taxonomy_category]	Categorizations for application files in App Engine Studio. By default, application files are categorized as data, experience, logic and automation, or security. This table includes configurations to support the default user experience for App Engine Studio.
Taxonomy Details [sn_app_eng_studio_taxonomy_details]	Details about application files that a developer creates in App Engine Studio. This table includes configurations to support the default user experience for App Engine Studio.

Note: The following data preservers are added for tables related to pipelines:

- Environment
- Pipeline
- Pipeline Environment Order
- Pipeline Types

The data preservers prevent records in these tables from being overwritten during cloning on a non-production instance. For more information, see Data preservation on cloning target instances.

Configure App Engine Studio

App Engine Studio (AES) guided setup provides a sequence of tasks that help you configure AES on your ServiceNow instance.

App Engine Studio must be installed on the instances where you expect to be developing your applications. In earlier versions, you were required to install AES on all instances in your pipeline.

If you plan to clone your production instance to one or more non-production instances, you should also install the AES product on your production instance prior to cloning. For more information, see Instance Clone.

Learn more about AES configuration	Additional ServiceNow resources
ServiceNow provides several additional resources on configuring and administering App Engine Studio.	https://www.servicenow.com/docs/access?context=app-engine-studio-rn&version=brazil&pubname=brazil-release-notes&ft:locale=en-US

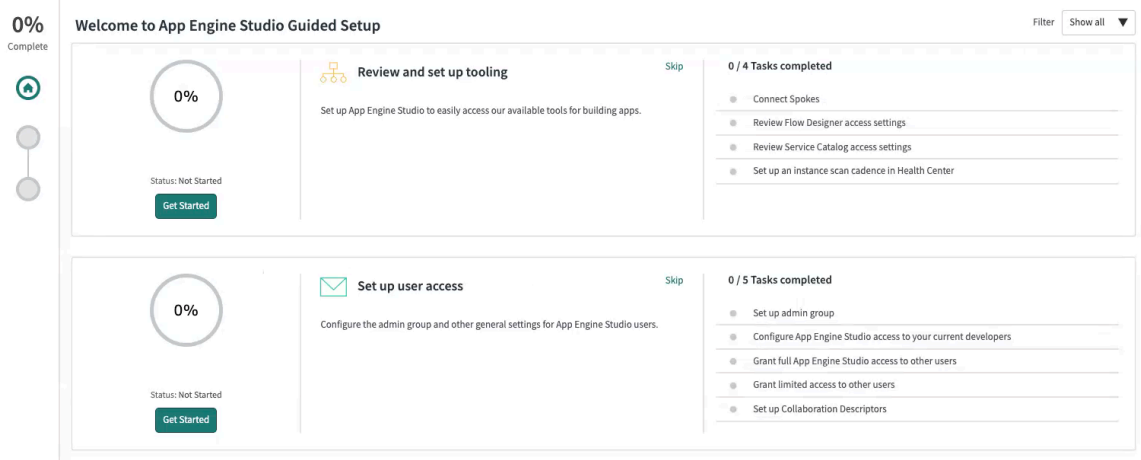
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Learn more about AES configuration	Additional ServiceNow resources
	ServiceNow Community site App Engine Studio video on adding security

Guided setup to configure AES

To open App Engine Studio guided setup, navigate to **All** > **App Engine** > **App Engine Studio** > **Configuration** > **Guided Setup**.

The landing page provides information on the different tools and user access. Select the **Get Started** button in the top, right corner to start your configuration.



The App Engine Studio Guided Setup page provides a list of different categories. Select the **Get Started** button under each category to start configuring App Engine Studio.

Note: If you have previously started any of the guided setup tasks, and then exited without completing them, the **Get Started** button is labeled **Continue**.

For detailed instructions on any of the tasks initiated from guided setup, see the Perform AES configuration tasks.

Configure AES personas and roles

The responsibilities of your staff are controlled by roles assigned to each member. Personas aren't explicitly part of App Engine Studio (AES) but administrators assign roles to give team members permission to configure or use AES.

Low-code/citizen developer

Low-code/citizen developers are tech savvy and interested in creating apps. Though they might not have formal coding or app development training, citizen developers can submit ideas for new apps and, if approved, build them using AES.

App Engine Studio admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

App Engine Studio admins manage all processes related to app development in AES. They review new app ideas, manage app development and deployment, and manage collaborators, usually in the App Engine Management Center.

App template admin

The app template admin makes sure that the right people have access to appropriate predefined and custom templates for app development in App Engine Studio.

App template author

The app template author creates and edits custom app templates in App Engine Studio and can share them with other users or groups.

Security admin

The security admin creates and modifies roles and access control lists for apps. This role is set on the platform level, and it is required for making updates to roles in App Engine Studio.

System administrator

The system administrator has access to all system features, functions, and data, regardless of security constraints. Grant this privilege carefully. If you have sensitive information, such as HR records, that you must protect, create a custom admin role for that area and train a person who is authorized to see those records to act as the administrator.

Professional ServiceNow developer

Professional ServiceNow developers can work in App Engine Studio, usually as collaborators with citizen developers. Because of their coding knowledge and development background, though, they're more likely to build and customize apps on the ServiceNow AI Platform, using more complex building tools.

Personas for AES

Persona	Roles required	Responsibilities
Low-code/citizen developer	App Engine Studio Users group (sn_app_eng_studio.user)	• Submit requests for new custom applications to build in App Engine Studio. • Understand ServiceNow and application development best practices. • Build and test applications in App Engine Studio. • Submit App Engine Studio applications for IT review. • Maintain and change the application during its life cycle if determined during the intake process.
	App Engine Studio User Limited group (sn_app_eng_studio.user)	• Collaborate with other users on applications they've been given permission to see.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Personas for AES (continued)

Persona	Roles required	Responsibilities
		• Submit App Engine Studio applications for IT review. • Understand ServiceNow and application development best practices. **Note:** Users in the App Engine Studio User Limited group can't create new apps.
App Engine Studio admin group	• atf_test_designer • scan_admin • app_engine_admin • sn_app_eng_studio.admin • sn_collab_request.collab_request_admin • sn_table_bldr_wzd.table_bldr_wzd_admin	• Manage App Engine Studio processes and properties. • Review and approve/reject submitted application requests based on intake guardrails defined by the system administrator. • Provision App Engine Studio user access. • Review submitted App Engine Studio applications. • Manage deployment requests. • Manage promotion of App Engine Studio applications across instances. • Execute ATF tests/suites in testing instances. • Ensure instance scans run with proper definitions. • Collaborate with the system administrator to resolve application conflicts that arise on the platform.
App template admin	app_template_admin	• Control access to custom and predefined templates. • Activate and deactivate templates. • Share templates globally or with users and groups.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Personas for AES (continued)

Persona	Roles required	Responsibilities
App template author	app_template_author	• Create and edit custom templates in App Engine Studio. • Share custom templates with users or groups.
Security admin	security_admin	• Create and modify roles and access control lists. • Required for making updates to roles in App Engine Studio.
System administrator	Admin	• Install and configure App Engine Studio and its dependent apps. • Define environments. • Configure pipelines. • Provision access to the App Engine Studio administrator group. • Upgrade the App Engine Studio application, when applicable. • Define guardrails for which AES administrators can approve or reject application intake requests. • Collaborate with professional ServiceNow developers to create instance scan definitions for the platform. • Collaborate with AES administrators to resolve application conflicts that arise on the platform.
Professional ServiceNow developer	• atf_test_admin • delegated_developer • scan_admin • sn_app_eng_studio.user • sn_collab_request.collab_request_admin	• Build and test applications in ServiceNow Studio and App Engine Studio • Collaborate with citizen developers on an as-needed basis to deliver applications in App Engine

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Personas for AES (continued)

Persona	Roles required	Responsibilities
		Studio, including the following tasks: ○ Ensuring application development and organizational best practices are followed by citizen developers. ○ Helping with review and testing of submitted App Engine Studio applications. ○ Building complex configurations involving UI Builder, Mobile App Builder, Workflow Studio, or other builder tools. • Create ATF tests/suites for applications. • Collaborate with a system administrator to create instance scan definitions for the platform.
AES admin	sn_app_eng_studio.admin	Granular admin role that provides access to App Engine Studio and enables administrative access to AES tables. Can be combined with other granular admin roles or delegated development for additional permissions. For more information about granular admin roles, see Platform security granular admin roles.
Collaboration request admin	sn_collab_request.collaboration_request_admin	Granular admin role that provides administrative access to collaboration tables.
AES Table Builder Wizard admin	sn_table_bldr_wzd.table_bldr_wzd_admin	Granular admin role that provides access to manage permissions during table creation in AES.

Important: Granular admin roles cannot manage tables that depend on ServiceNow AI Platform tables, such as the Experience Visibility Control table (sn_udc_experience_visibility_control) or Preferred Tables (sn_preferred_table). To manage such tables, additional granular admin roles are necessary. For more information, see Platform security granular admin roles.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Perform AES configuration tasks

As you work through the App Engine Studio (AES) guided setup, you must perform different configuration tasks.

Setting up environments, tools, and user access

The tasks for configuring App Engine Studio are listed below, along with links to detailed instructions for completing them.

App Engine Studio configuration tasks

Task	Description
Install App Engine Studio in your development instances.	You can install the App Engine Studio application (com.snc.app-engine-studio) if you have the admin role. The application installs related ServiceNow® Store applications and plugins if they are not already installed. For detailed instructions, see Installing App Engine Studio.
Set up instance credentials.	For each instance that you're using, create a Connections and Credentials alias. Ensure that the alias type is **Credential**, not **Connection and Credential**. For detailed instructions, see Set up instance credentials.
Activate Playbooks.	You can edit processes in App Engine Studio, but you must activate Playbooks for App Engine Studio and enable the **Process Automation Designer for App Engine [com.glide.pad.license]** plugin to get started. For more information, see Activate Playbooks.
Set up IntegrationHub spokes.	Activate spokes to enhance your Workflow Studio and App Engine Studio experience with integration-specific content. The following spokes are available for use with App Engine Studio. • Gmail spoke • Google Sheets spoke • Jira spoke • Microsoft 365 Excel spoke • Microsoft Teams spoke • Slack spoke • Twilio spoke • X Spoke (formerly Twitter Spoke) • Zoom spoke
Review access settings for Flow Designer in your development instance.	Enable your developers to use the editing capabilities that best suit them.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

App Engine Studio configuration tasks (continued)

Task	Description
	For detailed instructions, see Review Flow Designer access settings.
Review access settings for Catalog Builder.	Enable your developers to add items to the appropriate catalogs and categories. For detailed instructions, see Review Catalog Builder access settings.
Set up an instance scan cadence in Health Center.	Schedule regular scans of your instance so that you can identify possible issues that arise from application development in App Engine Studio. For detailed instructions, see Set up an instance scan cadence in Health Center.
In your production instance, add users to the App Engine Studio Admin group.	Add members to the group so they can review app- and deployment-related requests. Also, define a contact email address for the group. For detailed instructions, see Add users to the App Engine Admin group.
In your development instance, add developers to the App Engine Studio Users group.	Enable developers in your organization to build applications in App Engine Studio. For detailed instructions, see Grant user access to AES.
In your development instance, add users to the App Engine Studio User Limited group.	Enable developers in your organization to collaborate on applications that someone else created in App Engine Studio. For detailed instructions, see Grant user access to AES.
Set up collaboration descriptors.	Set up custom collaboration descriptors in the global scope for use in App Engine Studio. For detailed instructions, see Set up custom collaboration descriptors.
In your non-production instance, specify an app template admin.	Grant the app_template_admin role to users who will manage template activation, deactivation, and sharing in App Engine Studio. For detailed instructions, see Manage template access.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Learn more about AES configuration	Additional ServiceNow resources
ServiceNow provides several additional resources on configuring and administering App Engine Studio.	https://www.servicenow.com/docs/access?context=app-engine-studio-rn&version=brazil&pubname=brazil-release-notes&ft:locale=en-US ServiceNow Community site App Engine Studio video on adding security

Add users to the App Engine Admin group

Add users to the App Engine Admin group in your production instance to give them administrative rights to App Engine Studio (AES). You must also identify a contact email address for the group. After you configure the email address of the App Engine Admin group, members can receive notifications for app development-related requests, including app intake requests.

Before you begin

Role required: admin

About this task

If you are not using pipelines, you must also add users to the App Engine Admin group in your development instance.

Procedure

1. Navigate to **All** > **User Administration** > **Groups**.
2. Open the App Engine Admins group.
3. Select the **Group Members** related list and select **Edit**.
4. Select one or more names in the **Collection** list.
5. Enter a contact email address for the group.
6. Select **Add**.
7. Select **Save**.

Grant user access to AES

Control who has access to build applications in App Engine Studio (AES) by adding users to AES Users group or the AES User Limited group.

Before you begin

Role required: user_admin or admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

About this task

Grant access to build applications in AES by adding users to the App Engine Studio Users group in your organization's development instance. Each group member is automatically assigned the sn_app_eng_studio.user role.

To restrict access so that users can work in App Engine Studio but not create apps or see templates, add users to the App Engine Studio User Limited group.

For more information about the differences between the App Engine Studio Users group and the App Engine Studio User Limited group, see Configure AES personas and roles.

Procedure

1. Navigate to **All** > **User Administration** > **Groups**.
2. Add users by selecting either the **App Engine Studio Users** group or the **App Engine Studio User Limited** group.
   a. In the Group Members related list, select **Edit**.
   b. On the Edit Members page, move each developer from **Collection** to **Group Members List**.
   c. Select **Save**.
   d. On the group record, select **Update**.

Related topics

Components installed with AES

Manage template access

Control who has access to templates in App Engine Studio (AES) using the app_template_admin role. App template admins activate and deactivate templates and grant other users access to share their templates.

Before you begin

Role required: admin

About this task

An administrator grants the app_template_admin role to users who will manage template activation, deactivation, and sharing. This role should be granted on non-production instances where App Engine Studio users author and use templates.

To create apps and work with templates in AES, users must be an admin or in the App Engine Studio Users group. If they are in the AES User Limited group, they can only edit existing apps, not create new ones.

Procedure

1. Navigate to **All** > **User Administration** > **Users**.
2. Select the user you want to give the app_template_admin role to.
3. On the Roles related list, select **Edit**.
4. Move **app_template_admin** from the **Collection** list to the **Roles list**.
5. Select **Save**.

Set up custom collaboration descriptors

Customize the set of development collaborators you list in App Engine Studio (AES) using Guided Setup.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin
Role required: admin

About this task
Owner and Editor collaboration descriptors are provided out of the box in AES and have special capabilities. Development permissions included in each descriptor can be configured to your platform needs. You can create custom descriptors as well, in the global scope, so they can be used in AES.

Note: You should create collaboration descriptors in addition to Owner and Editor in the global scope. If you want collaboration descriptors to appear and be used in AES, you should also set them to `standard = TRUE`. AES doesn't support collaboration descriptors that are created in custom scopes, and non-standard collaboration descriptors don't render in AES.

Admins or delegated developers who invite collaborators can use these descriptors to delegate different types of work.

Procedure
1. Navigate to **All** > **App Engine** > **App Engine Studio** > **Guided Setup**.
2. Select **Get Started**.

   Note: If you have previously started any of the guided setup tasks, and then exited without completing them, the **Get Started** button is labeled **Continue**.

3. In the Set up user access section, select **Set up Collaboration Descriptors**.
4. In the Set up Collaboration Descriptors section, select **Configure**
5. Select **New**.
6. Give the new app collaboration descriptor a name and description.
7. Select **Submit**.

What to do next
Further customize collaborator permissions in App Engine Studio. For more information, see Change collaborator permissions.

Delegate developers using AES

Delegated development enables designated users without a system admin role to develop or deploy applications on the ServiceNow AI Platform. This enables administrators and delegated developers to work together to deliver custom applications through App Engine Studio (AES).

After an administrator reviews a request, they delegate the development work by providing a link to the developer that opens App Engine Studio in your organization's development environment. The administrator can:

- Send the link directly to a developer via email
- Post the link on a portal page
- Create a catalog item for delegated developers to request App Engine Studio access

Delegated developer role in AES

The delegated developer role enables an admin to elevate user permissions to a dedicated role with specific access for App Engine Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

A delegated developer has more permissions than a user, but less than an admin. The delegated developers each have their own permissions.

An admin may have several delegated developers, each with their own set of permissions. For example, you may want a designated developer for developing a time-off request, and another developer with separate permissions for creating support tickets.

Some AES features may be inaccessible to the delegated developer, depending on permissions. Your role enables you to access, edit, and add objects. If a feature of App Engine Studio is unavailable to you, contact your system administrator.

For more information about delegated development and permissions, see Delegated development and deployment.

Restricted ability to create apps

In addition to delegated development, admins can define whether AES users can create apps or only work on existing apps.

- Grant access to build applications in AES by adding users to the App Engine Studio Users group in your organization's development instance. Each group member is automatically assigned the sn_app_eng_studio.user role.
- To restrict access so that users can work in App Engine Studio but not create apps or see templates, add users to the App Engine Studio User Limited group.

AES integration with a Git source control repository

Enable application developers to integrate App Engine Studio (AES) with a Git source control repository to save and manage multiple versions of an application from a non-production instance.

Linking an application to source control enables all application developers on a non-production instance to:

- Import applications from a Git repository.
- Pull and apply remote changes from a Git repository.
- Commit all local changes on the instance to a Git repository.
- Create tags to permanently link to a given version of an application.
- Create branches to maintain multiple versions of an application simultaneously.

Integration requirements

To link an application to source control:

- The user must have the admin role.
- The non-production instance must have network access to the Git repository.
- Each application must be within its own Git repository.
- The repository user credentials must grant read and write access.

Note: All application developers on the instance share a single set of credentials per repository.

Options available from App Engine Studio

After linking an application to source control, application developers can use App Engine Studio to manage the repository. From App Engine Studio, developers can:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Edit the application repository credentials.
- Commit all local changes on the instance.
- Apply remote changes from the repository.
- Create a branch.
- Switch branches.
- Import an application from a remote repository.

Source control integration does not support managing applications on a production instance. Instead, you can manage applications on a production instance using the application repository, an update set, or App Engine Studio. For more information about managing applications on a production instance, see Application sharing.

Options available from a Git repository

The ServiceNow platform offers limited support for modifying linked application files outside of an instance. From Git, developers can:

- Move application files to a different Git directory structure.
- Edit application files outside of App Engine Studio.

The system generates a properties text file called `sn_source_control.properties` at the root level of the repository. To move application files to a different Git directory structure, application developers can set the *path* parameter to specify the subfolder path containing their application files. For example, if you moved your application to the `src/app` subfolder, set the *path* to `path=src/app`.

The system generates a `checksum.txt` file in the Git repository to determine if any application files have been changed outside of App Engine Studio. When the checksum value from the file matches the current checksum value, the integration skips the validation and sanitization process. When the checksum values do not match, the integration validates and sanitizes the application files as part of the source control operation. The sanitization process:

- Creates upgrade log entries for each sanitization action taken.
- Removes unsupported folders and files from the repository.
- Aborts all source control operations when a system application file fails XML schema validation. For example, if a database dictionary record fails XML schema validation, the system aborts all operations.
- Skips the current source control operation when a non-system application file fails XML schema validation.

The Git integration sanitizes only content within the application path listed in the `sn_source_control.properties` file. Repository content outside the application path is ignored.

MID Server support

Use an existing MID Server to connect to a source control repository. Connecting an application through a MID Server enables access to repositories behind a firewall.

Source control role permissions

For more information on roles and collaborators, see Application collaboration.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Link an application or application-customization to source control

Linking an application or application-customization to source control allows application developers to manage changes in App Engine Studio (AES) from a Git repository.

Before you begin

- Learn more about Manage customizations to applications.
- Create a dedicated Git repository for the application. For increased security, enable multi-factor authentication for the Git repository.
- Generate an access token that the source control integration can use instead of a password and multi-factor authentication passkey while creating a Credential record. Search for personal access token on GitHub or GitLab.
- Restrict permissions on the access token to allow read and write access to the Git repository.
- Verify that the non-production instance has network access to the Git repository.
- Ensure that users add the email address to their respective Users Table (ServiceNow sys_user) records that they use in their commits to the Git repository.
- Role required: admin

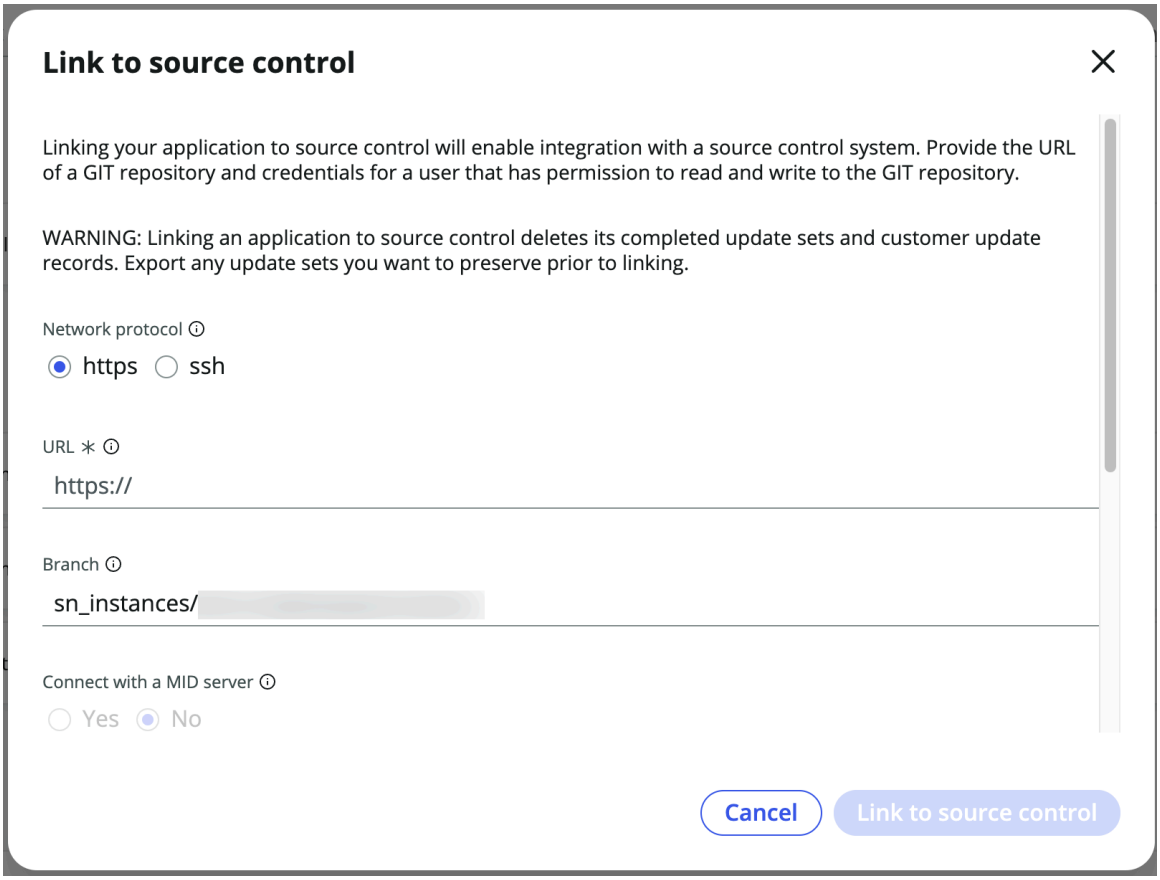
About this task

The source control integration does not support linking to an application or customization on a production instance. Instead, install applications on a production instance from the application repository, an update set, or the App Engine Studio.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. Select **Source control** > **Link to source control**.
   App Engine Studio displays the Link to source control dialog box.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



4. Enter the connection details for the Git repository.

Source control connection details

Field	Description
Network protocol	Https or SSH credential type that enables secure channel data exchange.
URL	The URL to the Git repository where you want to save application files. For SSH protocol, use command to generate private key `ssh-keygen -t rsa -m PEM -b 4096 -C "email@address"`. **Note:** If the Git repo URL for SSH provided by your Git server does not work, check with your Git server owner or provider for the correct URL. There may be additional specifications such as scheme protocol prefixes, port numbers, and so on, required for your Git repo URL to function.
Branch	The repository branch to work on within the application. **Note:** This branch is used for commits. The default branch is set to "main" if it is not already set in the remote repository. If there is no default branch on the remote git repository, the instance creates a new default branch with the name "main". This is configured using the `glide.source_control.git_default_branch` system property.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
MID Server Name	The name of the existing MID Server to link through. **Note:** Use a separate MID Server to prevent conflicts with Discovery activities. Be sure that the MID server user can create files to the sys_attachment table and that the table can accept files of the "bundle" type. Linking or an application through a MID Server enables access to repositories behind a firewall. See MID Server for more information.
Default email	The committer email address is defined by the sys_user record if available. But if a committer's sys_user record email field is empty, the system generates an alternate email (username@instancename.service-now.com). You can also enter a default email address and change it later. To use that default email address in all cases, select the check box.
Credential	The credential to be used with the selected protocol. See Get started with credentials to learn more about creating credentials. **Note:** If you select the SSH network protocol, enter a valid credential of the SSH private key type. If you select the https protocol, enter a valid credential of the Basic Auth credentials type.
Commit Comment	An optional description of the repository or application.

Note: All application developers on the instance share a single set of repository credentials.

5. Select **Link to source control**.
   The system validates the connection and user credentials and displays a success message.

   All application developers on the instance can use the linked Git repository to manage changes.

Related topics

MID Server

Getting started with credentials

Edit a Git repository configuration

You can edit a Git repository's integration with App Engine Studio (AES) to change the network protocol selection, credentials or other field entries.

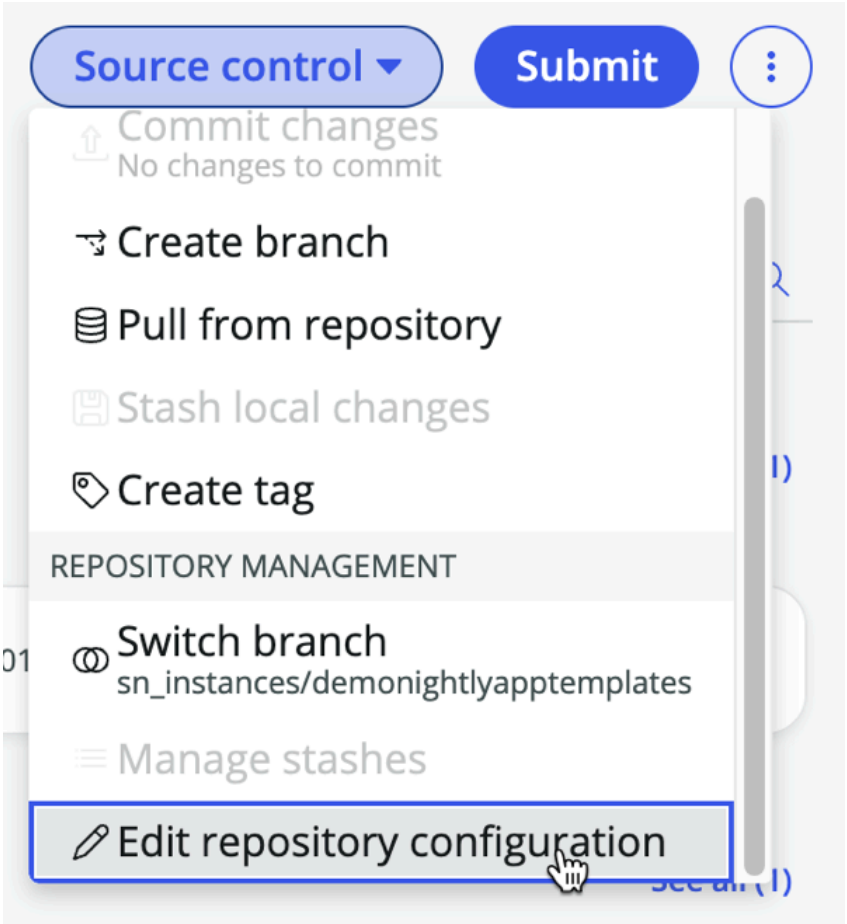
Before you begin

Role required: admin

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. Select **Source control** > **Edit repository configuration**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



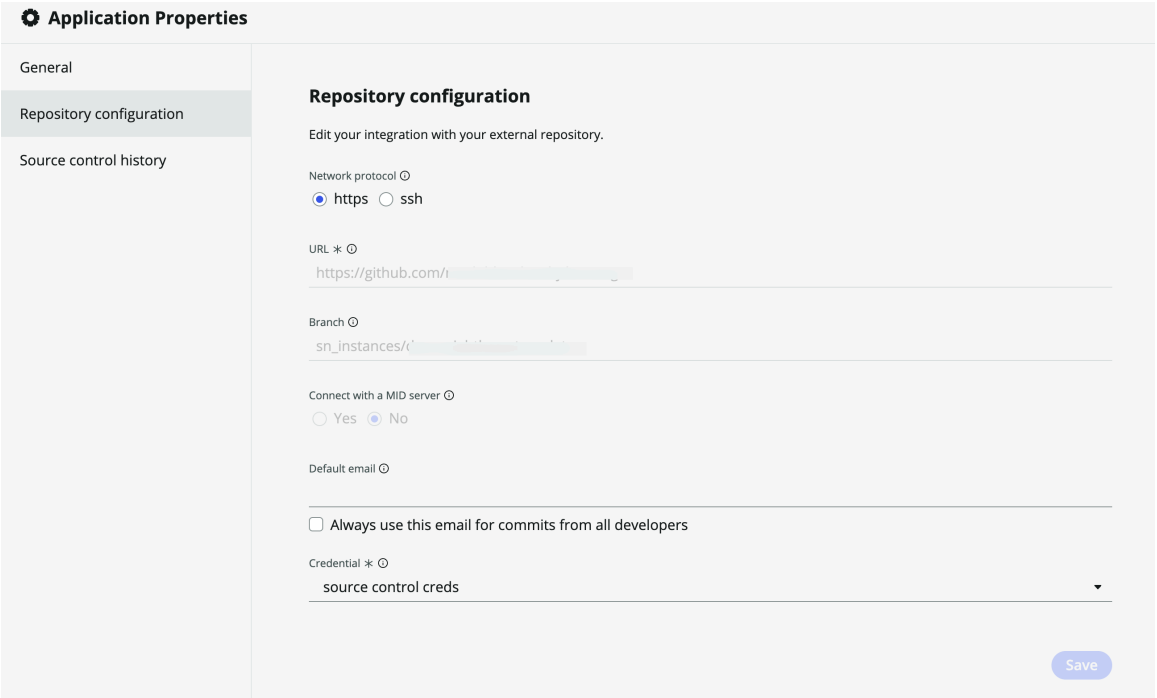
4. On the form, fill in the fields.

Repository configuration fields

Field	Definition
Network protocol	Protocol for source control, either **https** or **ssh**.
URL	URL address of your repository.
Branch	Current branch for the repository.
Connect with a MID server	Whether or not to use a MID server connection, which enables access to repositories behind a firewall. **Note:** If you have no MID server name, you can select a new one from the drop-down list. If you choose a new MID server, in the **Source control** menu before making any further source control operations to avoid errors.
Default email	Email address for the committer, which is defined by the sys_user record, if available.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Definition
	If a committer's sys_user record email field is empty, the system generates an alternate email (username@instancename.service-now.com). You can also enter a default email address and change it later by selecting the **Always use this email for commits from all developers** check box.
Credential	Saved credentials to use for the source control connection. All application developers on the instance share a single set of credentials per repository. For information on working with credentials, see Get started with credentials.



5. Select **Save**.

Cloning instances with AES

Learn how to protect the data, tables, and templates you've created in App Engine Studio when using System Clone to copy instances from production to non-production.

Preserving data and tables when cloning

The following are requirements for cloning instances with AES:

1. Verify that all of your AES plugins are installed across all instances.
2. If you're cloning a production instance, verify that you've set up a data preserver on the target instances to preserve the Automated Test Framework (ATF) and Instance Scan properties. For

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

more information about data preservers, see Create a clone preserver and Preserve data during an instance clone.

Important: By default, the ATF system property is disabled to prevent you from accidentally running these tests on a production system. Running ATF on a production instance is neither recommended nor supported due to the potential for data corruption or outages.

3. If you're collecting development and deployment data, the App Engine Management Center (AEMC) plugin must be installed on all instances.

Cloning data and tables from a production instance over a non-production instance can overwrite data in your non-production tables. To ensure that data isn't lost in development environments, create a cloning strategy for collaboration.

1. The following tables have data preservation to ensure that the tables are correctly cloned between instances:

   Note: For the following tables, preservation is for global scope only.

   - Collaboration Descriptor tables:
     - App Collaboration Descriptors (sys_appcollab_descriptor)
     - App Collaboration Descriptor Permissions (sys_appcollab_permission_m2m)
   - Collaboration Users and Groups tables:
     - App Collaboration Users (sys_appcollab_user)
     - App Collaboration Groups (sys_appcollab_group)

   The data preservation ensures data is retained on the tables in the development instances.
2. The following tables have clone exclusions:
   - Collaboration Descriptor tables:
     - App Collaboration Descriptors (sys_appcollab_descriptor)
     - App Collaboration Descriptor Permissions (sys_appcollab_permission_m2m)
   - Collaboration Users and Groups tables:
     - App Collaboration Users (sys_appcollab_user)
     - App Collaboration Groups (sys_appcollab_group)

   Clone exclusions ensure data from production instances isn't copied to development instances.
3. If AES is the only application using the Credentials table, consider creating data preservers for Credential Alias, Basic Auth, and Discovery credentials. Otherwise, you must ensure that these tables aren't overwritten when the production instance is cloned to non-production instances.
4. The following users must be reassigned their roles after cloning:
   - Users in the AES Users group
   - Users in the AES User Limited group
   - Users who have the sn_app_eng_studio.user role in non-production instances
5. After cloning, a **ReSync Collaborations Permissions** post-clone clean-up script runs automatically, so any applications that were the same on production and development instances are automatically have collaborators synced. Developers can resume development on them immediately.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: The cloned instance must have the collaboration plugin enabled.

6. If some applications were backed up prior to cloning and retrieved after cloning, you can use the **Resync collaboration permissions** related link on the sys_app record to reassign users and groups to their appropriate delegated development permissions.
7. If a collaboration descriptor is no longer associated with a user or group after cloning (in the event that development apps were wiped out during cloning as they were not in the source instance), select the **Clean up records with empty references** related link to remove the unreferenced user or group from the collaboration table. You should run this UI action after the cloning is done and all preserved applications have been retrieved (with **Resync collaboration permissions** already run on them).

The following tables have data preservation to ensure that the tables are correctly cloned between instances:

- Pipeline Instance
- Request Authorization Key
- Deployment Request
- Deployment Environment Request

Preserving app templates when cloning

Admins must protect custom templates from being overwritten during the cloning process. Without protection, templates created in AES (both from existing applications and from scratch) are in danger of disappearing during a clone.

When you create a template in AES, a scoped app is automatically generated on the Custom Applications [sys_app.list] table in your instance. Though they have different contents, template applications and standard custom applications are treated similarly on the ServiceNow AI Platform. So, preserving app templates during a system clone works the same way as preserving an application.

To protect app templates on your non-production instances, follow the process in Create a clone preserver or Preserve data during an instance clone.

More information on cloning and data preservation

See the following topics for more information on cloning and data preservation:

- For more information on using the System Clone tool, see Instance Clone.
- For more information on data preservation, see Create a clone preserver.
- For more details on cloning instances with AES, see the App Engine Studio System Administrator Guide on ServiceNow University.

Learn more about cloning instances with AES	Additional ServiceNow resources
ServiceNow provides several additional resources on cloning instances with App Engine Studio.	 Cloning basics knowledge article

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Learn more about cloning instances with AES	Additional ServiceNow resources
	 Cloning instances tips and tricks knowledge article
	 Extensive FAQ knowledge article on cloning instances
	 App Engine Enterprise - Data Preservation During System Clone Whitepaper **Note:** You must log in to ServiceNow University to access this resource.

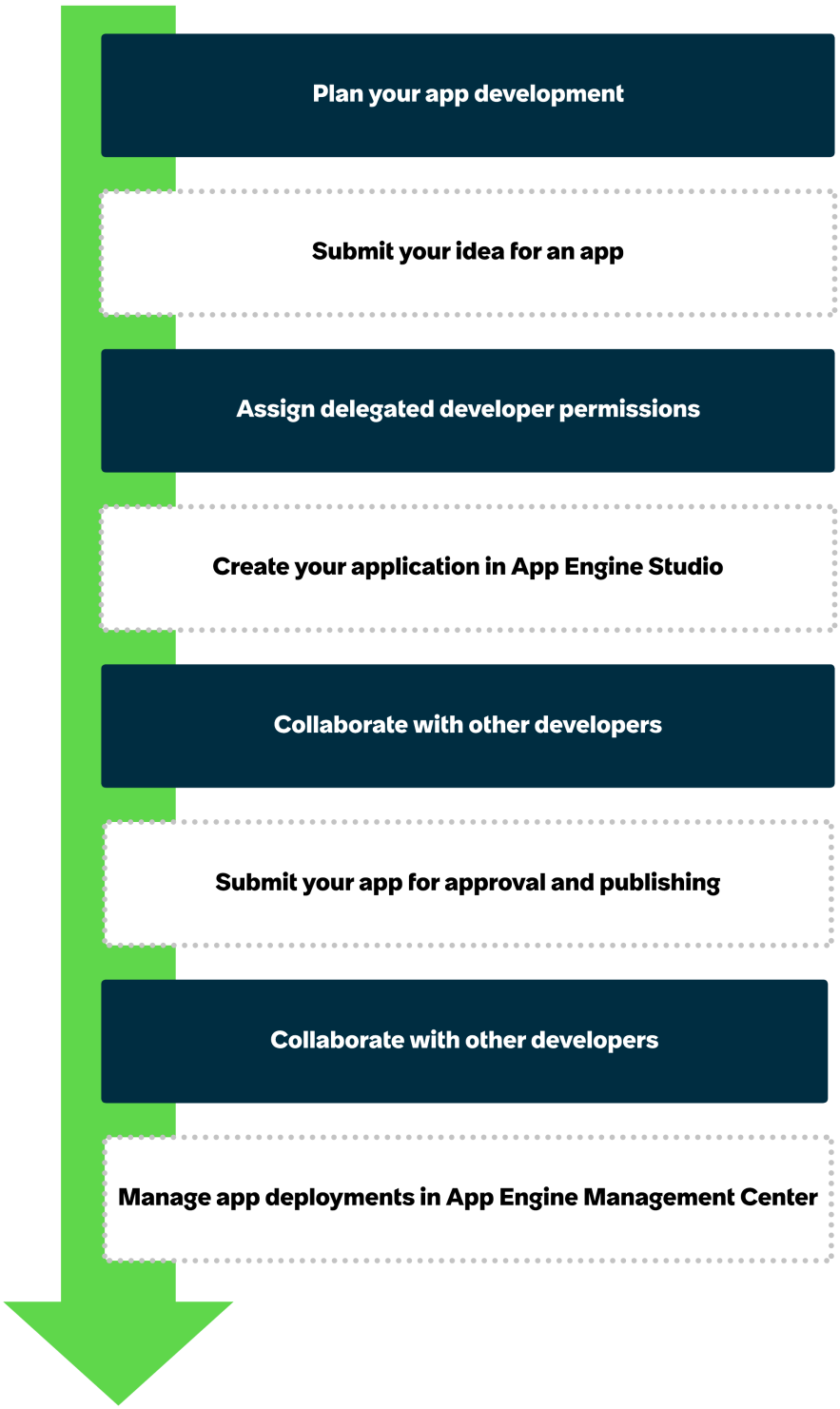
Building apps in App Engine Studio

Meet the business needs of your organization by building custom applications in App Engine Studio (AES).

App Engine Studio build process

At a high level, there are seven main steps for building a new application in App Engine Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



1. Plan your app development
2. Submit your idea for app development
3. Delegate developers using AES
4. Create your app
5. Collaborate with other developers

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

6. Submit your app for approval and publishing
7. Manage deployments using App Engine Management Center (AEMC)

Considerations

There are several considerations to keep in mind as you build applications in App Engine Studio:

- You access App Engine Studio by navigating to **App Engine Studio** > **App Engine Studio** or with the following web address: `<instance-name>.service-now.com/now/appenginestudio`. If you don't have a working link, contact your administrator.
- When you navigate back and forth between different applications in App Engine Studio, the application scope changes accordingly.

Working with data, experiences, automation, and security

App Engine Studio enables you to add as many data tables, types of experiences, logic and automation, and security roles as needed. However, you aren't required to add all four categories to your app if the app doesn't need them.

App creation tutorial

This App Engine Studio tutorial series guides you through the entire process of building an application using the low-code tools available in the ServiceNow AI Platform.

If you want to build an application that can solve your business problems, the App Engine Studio app creation tutorial is your guide to building an application from start to finish. The tutorial walks you through each step of app development, from planning to building and testing. You also learn to use the powerful low-code app development tool, App Engine Studio, and its suite of integrated tools to build and enhance applications easily.

The use case for the tutorial is simple: We want to create an application that automates the employee travel request process for an organization. To do so, we build the application step-by-step, including the data model, user experience, logic and automation, and security for the application.

Each step in the tutorial demonstrates one aspect of app development. You can follow along with the tutorial sequentially or skip to the steps most relevant to you. Use the app creation tutorial and related resources to build your own application.

Complete the tutorial

Step 1	Planning your application
Step 2	Create an app
Step 3	Building a data model
Step 4	Creating user experiences
Step 5	Adding logic and automation
Step 6	Test your application

Planning your application

Plan out your application before building it in App Engine Studio to streamline the development process.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

When you want to build an application, the first thing you should do is develop a plan for the app. Planning involves outlining what the application does and identifying who will use the application. By planning your application, you can better understand the purpose of your application and address potential issues before development begins.

This is the first step in the app creation tutorial. Follow along to develop a plan for an application that manages employee travel requests for an organization.

Step 1	○	Planning your application
Step 2	○	Create an app
Step 3	○	Building a data model
Step 4	○	Creating user experiences
Step 5	○	Adding logic and automation
Step 6	○	Test your application

https://player.vimeo.com/video/964898255?badge=0&autopause=0&player_id=0&app_id=58479

Video sections

Timestamp	Section
0:17	Identify the use case for your application.
0:21	Visualize the workflow for your application.
0:42	Outline the functional requirements for your app.

Step 1: Identify the use case for your application

Start by identifying the use case for your application. In our tutorial, we're creating an application to manage employee travel requests for an organization. The app automates the entire travel request process and involves users such as employees, managers, regional heads, and travel desk agents as needed.

Consider the problem that you aim to solve with your application. To determine if your use case is a good fit for the tools featured in this tutorial, see Plan your app development.

Step 2: Visualize the workflow for your application

Once you have identified the use case for your application, visualize the end-to-end workflow for the application. The end-to-end workflow is the sequence of actions an application follows to complete a task or process.

To visualize the end-to-end workflow for your application, map out what your application does from start to finish. Include any loops that lead back to earlier steps in the workflow and any actions that cause you to exit the workflow.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Step 3: Outline the functional requirements for your app

It's also important to consider the functional requirements of your application. The functional requirements of an application include who uses the application, what tasks they perform in the app, and what data the app must collect to support these tasks.

Use the following prompts to outline the functional requirements for your application:

1. Identify the personas who use your application and consider what tasks they perform.

   In our tutorial, the personas who use the employee travel request application and their associated tasks are:

Persona	Tasks
Employees	Submit travel requests
Managers and regional heads	Approve or deny travel requests
Travel desk agents	Book travel and complete follow-up tasks

2. Determine the data that your application must collect.

   For example, the application in our tutorial must collect details about employees' travel requests, including:

   - Where the employee is traveling to and from
   - When the travel occurs
   - How much the travel costs

3. Choose the process or processes that you want to automate in the workflow for your application using logic and automation.

   Review the following table to see how we plan to use logic and automation in our tutorial application.

	What it does	Tutorial application example
Logic	Controls how and when certain actions occur in your application workflow.	Verifies that an employee's travel request is only forwarded to the travel desk after it has been approved by both the manager and the regional head.
Automation	Sets up automatic processes based on the logic that you define.	Updates the employee's travel request case status and sends email notifications whenever a travel request is rejected.

4. Consider the roles and permissions for your application.

   Roles determine whether users have access to application content. Permissions are the individual abilities that a user has within application content, such as creating, reading, updating, and deleting. You can determine the roles for your application and configure permissions for each role at a later step in the tutorial. For more information about roles in App Engine Studio, see Configure AES personas and roles.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

5. Think about how you want to organize the data collected by your application.

   The data that your application collects is automatically stored in tables. You might want to organize the data across different tables, depending on the processes that your application completes. You can decide how you want to organize the data for your application a later step in the tutorial. For more information about data organization in App Engine Studio, see Create a data model for your application.

Next steps

Once you've developed a plan for your application, you can begin building the application in App Engine Studio. Proceed to the next step: Create an app.

Create an app

When you're ready to start building your application, begin creating your application in App Engine Studio.

Before you begin

Note: To create apps in AES, you must be an admin or in the App Engine Studio Users group. If you are in the App Engine Studio User Limited group, you can only edit existing apps, not create new ones.

Role required: admin, sn_app_eng_studio.user

About this task

Once you've developed a plan for your application, you can start building it in App Engine Studio. App Engine Studio is a powerful tool that enables you to develop robust applications in a low-code environment.

This is the second step in the app creation tutorial. Follow along to create an application in App Engine Studio.

Step 1		Planning your application
Step 2		Create an app
Step 3		Building a data model
Step 4		Creating user experiences
Step 5		Adding logic and automation
Step 6		Test your application

https://player.vimeo.com/video/953654187?badge=0&autopause=0&player_id=0&app_id=58479

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Video sections

Timestamp	Section
0:06	Open App Engine Studio.
0:10	Create an application.
0:14	Name your application and provide additional details.
0:24	Review, add, and customize roles.
0:36	Go to your application dashboard.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. On the App Engine Studio home page, select **Create app**.
3. Enter a unique **Name** and **Description** for your application.
   We enter the following details for the application in our tutorial:

Field	Value
Name	`Travel request`
Description	`Enables employees to submit travel requests`

4. **Optional:** Upload a logo for your application by dragging an image into the **Drag app logo or browse to upload** field, or by selecting the field, selecting the image from your file directory, then selecting **Open**.

   Tip: You can add a logo for your application later using the **Application Properties** menu.

5. Select **Continue**.
6. Control who has access to content in your application by reviewing, adding, or customizing roles.
   By default, App Engine Studio provides admin and user roles. To add more roles to your application, select **Add a role**, and enter a name and description for the role.
   In our tutorial, we use the default roles and permissions. We don't add additional roles.
7. Select **Continue**.
8. On the summary screen, select **Go to app dashboard** to continue setting up your application.
   Your application dashboard displays each part of your application, including **Data**, **Experience**, **Logic and automation**, and **Security**. You'll configure these parts of your app in later steps of the tutorial.

Result

You've created an application in App Engine Studio. The application is empty and ready for configurations.

What to do next

Now that your application is created, you can start building the data model for the application.
Proceed to the next step in the tutorial: Building a data model.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Building a data model

After creating an application, start building the data model for the application. Learn about what a data model is and how we build the data model for the tutorial application.

Once you have created an application in App Engine Studio, you can start configuring each part of the application. In our tutorial, we start with the data model for the application.

This is the third step in the app creation tutorial. In this step, you learn about data models and build the data model for the employee travel request application.

Step 1		Planning your application
Step 2		Create an app
Step 3		Building a data model
Step 4		Creating user experiences
Step 5		Adding logic and automation
Step 6		Test your application

Overview of data models

The data model for an application describes the data that the application reads and writes and how the data is stored. To build a data model, you can create tables to store the data for your application. Each table contains rows that represent individual records and columns that define the type of information within each record. You can create as many tables as needed for your application and define relationships between the tables.

For more information about building data models in App Engine Studio, see Create a data model for your application.

Tutorial data model

In our tutorial, the data model for our application stores information about employees' travel requests, such as where the employee is traveling to and how much the travel costs.

We build the data model for the application in our tutorial in three phases:

Phase 1	Create a data table
Phase 2	Configure a data table
Phase 3	Share data between tables

Next steps

Start building the data model for your application. Proceed to the next step Create a data table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Create a data table

Create a table to store the data for your application.

Before you begin

Role required: admin, sn_app_eng_studio.user, or delegated developer

About this task

The first phase in building the data model for the employee travel request application is to create a table. In our tutorial, we create a blank table that extends the Task table.

Phase 1		Create a data table
Phase 2		Configure a data table
Phase 3		Share data between tables

This procedure covers the first part of the task demonstrated in the tutorial video. The rest of the procedure is outlined in the next topic Configure a data table.

Follow along to create a table for the employee travel request application.

https://player.vimeo.com/video/967024604?badge=0&autopause=0&player_id=0&app_id=58479

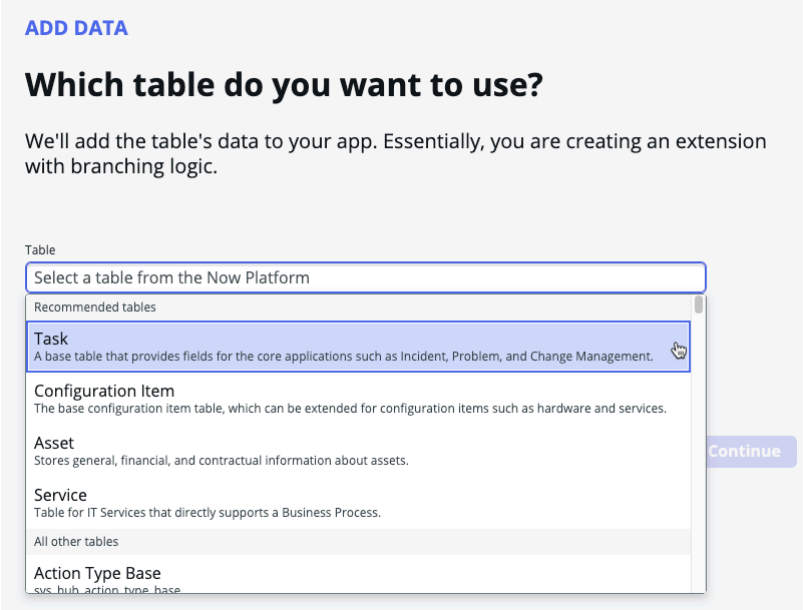
Video sections

Timestamp	Section
0:10	Create a data table.
0:14	Create a table by extending the Task table.
0:21	Define table properties.
0:33	Add permissions to your table.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. Select the travel request application.
3. On the application dashboard, select **+ Add** next to **Data**.
4. Select **Create a blank table**.
   A blank table enables you to create the fields, rows, and columns in your table from scratch. You can import existing data, such as spreadsheets, to create the table for your application.
5. Select **Continue**.
6. Select **Create from an extensible table**.
   Extensible tables are tables with built-in capabilities and functionality that you can use to create new tables. If you don't want to create a table from an extensible table, you can create a table from scratch by selecting **Create new table**.
7. Select **Task** from the **Table** list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



The Task table is one of the commonly extended tables in the ServiceNow AI Platform. The Task table includes built-in functionality for approvals and assignments, which we need for the table in our tutorial application.

8. Select **Continue**.
9. Define the properties of your new table.
   We define the following properties for the table in our tutorial:

Table properties

Field	Description	Tutorial table properties
Table label	Unique label to identify the table.	`Travel request`
Table name	Database name for the table. Auto-populates based on the **Table label** that you provided.	No action needed
Table name prefix	Database prefix for the table. Auto-populates based on the application that you created the table in.	No action needed
Make extensible	Option to enable other tables to share data from this table. For more information, see Table extension.	No action needed

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description	Tutorial table properties
Auto-number	Option to track table records with a unique number.	Select the check mark
Prefix	Abbreviated name of the table to append to the beginning of record numbers.	TRA
Starting number	Number to identify the first record created for your table.	No action needed
Number of digits	Maximum number of digits for the record number.	No action needed

10. Select **Continue**.
11. Control who has access to content in the table by adding permissions to each existing role, or creating new roles.

    Note: At least one role must have read access for you to be able to preview the table.

    In our tutorial, we grant the following permissions to the admin and user roles.

Role	Permissions
Admin	**All**
User	○ **Create** ○ **Read**

12. Select **Continue** to add the roles to your new table.
13. Select **Edit table** to continue setting up your new table.

Result
You have created a table for your application.

What to do next
Complete the setup of your table by proceeding to the next phase: Configure a data table

Configure a data table

Configure a table to capture the necessary data for your application.

Before you begin
Role required: admin, sn_app_eng_studio.user, or delegated developer

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

About this task

The second phase in building the data model for the employee travel request application is to configure the table so that it collects the necessary data for employee travel requests, such as location, dates of travel, and estimated airfare.

Phase 1		Create a data table
Phase 2		Configure a data table
Phase 3		Share data between tables

This procedure covers the second part of the task demonstrated in the tutorial video. The first part of the procedure is outlined in the previous topic Create a data table.

Follow along to configure a table to store the necessary data for the employee travel request application.

https://player.vimeo.com/video/967024604?badge=0&autopause=0&player_id=0&app_id=58479

Video sections

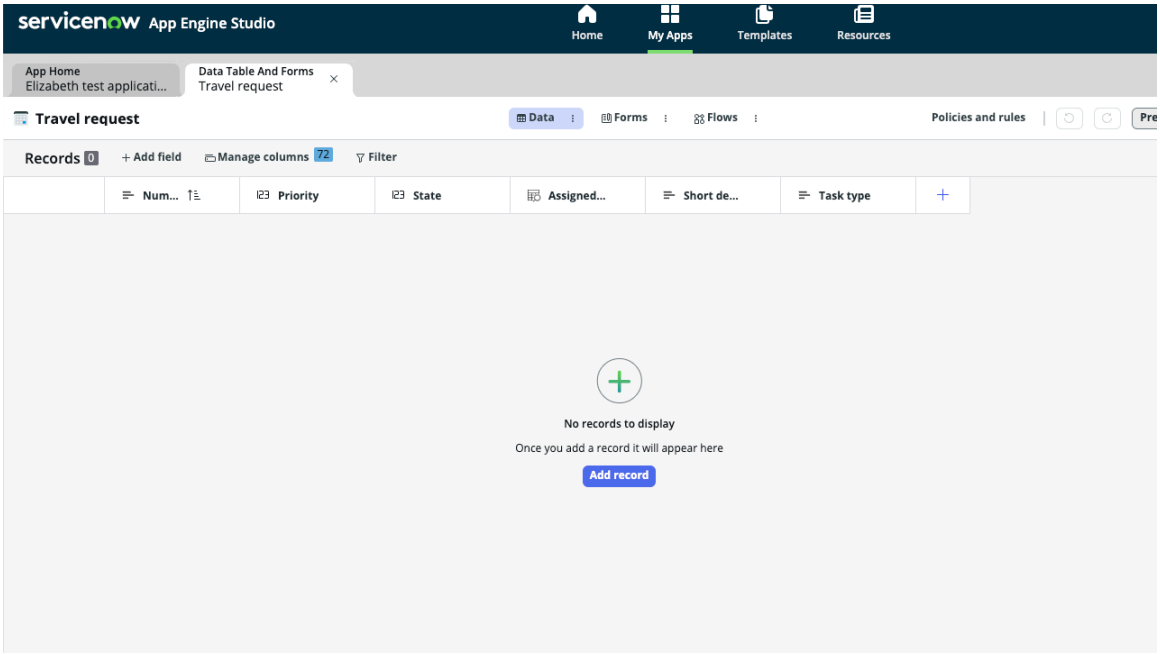
Timestamp	Section
0:48	Begin editing your table.
0:54	Add and configure fields in your table.
2:32	Change the data form layout.
2:39	Save your changes.

Procedure

1. Ensure that the employee travel request table is open.
   - To open a table from your instance home page, navigate to **All** > **App Engine** > **App Engine Studio**. Then select your application and select the table that you want to configure.
   - To open a table that you have just created, select **Edit table**.

   When your table is open, it appears in the canvas space.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



2. Select **Forms** in the top ribbon to change your view to **Forms** view.
   Forms view enables you to create your table as a form. You can choose other views depending on the type of data that your table stores.
3. Remove any unnecessary fields from your table by hovering over the field, then selecting the delete field icon ( ).

   Note: When you create a table from an extended table, the table comes with preconfigured fields. You can choose to keep or remove any preconfigured fields in your own table.

   In our tutorial, we remove the following fields:

   - **Priority**
   - **Configuration item**
   - **Parent**
   - **Assigned to**
4. Select **Save**.
5. Create new fields for your table.
   We must create several fields to collect details about employees' travel requests.

   a. Select **+ Add a field in the table** in the **Add form elements** panel.

   b. In the **Column label** field, enter `Departure Date`.

      Tip: The **Column name** field auto-populates based on the column label that you enter.

   c. In the **Type** field, select **Date**.

   d. **Optional:** To set additional properties for the field, select **Advanced settings**.
      We don't set additional properties for the fields in our tutorial. For more information about field types and advanced settings, see Field configuration in Table Builder.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

e. Select **Add**.

f. **Optional:** Select **Add another one** to create additional fields.

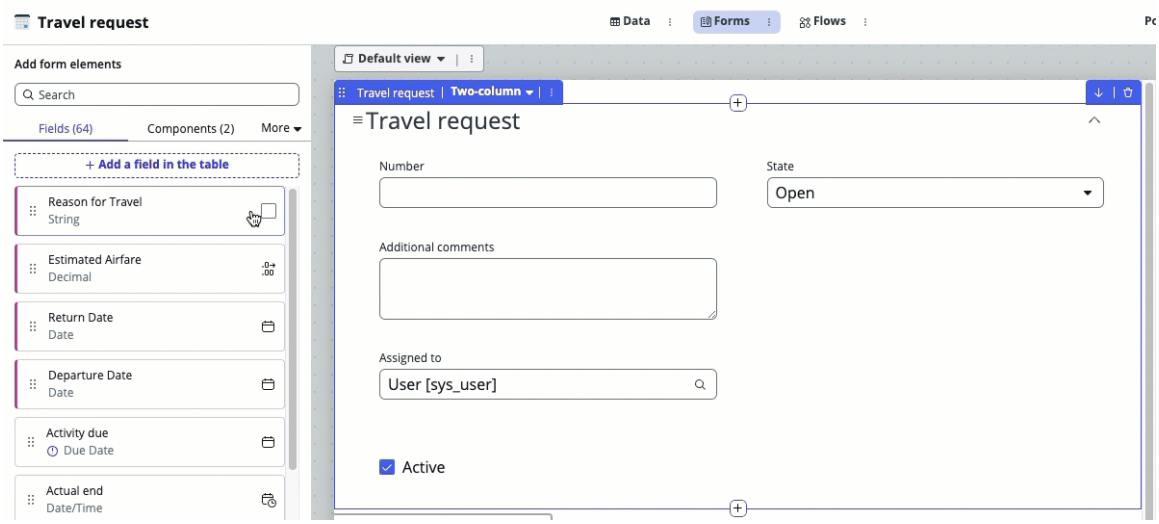
g. Repeat the process in steps a-f to add as many fields as you need.
We add more fields to collect additional information about employees' travel requests. The following table outlines the additional fields that we add to our table:

Column label	Type
`Return Date`	Date
`Estimated Airfare`	Decimal
`Reason for Travel`	String

h. When you have added all of the new fields that you need, select **Done**.
The fields that you have added will appear in the **Add form elements** panel.

i. To add the fields to your form, select the fields, then drag them into the **Default view** panel with your existing fields.

Tip: To add multiple fields to your table at once, hover over the field, select the check mark in each field, then drag the fields into the **Default view** panel.

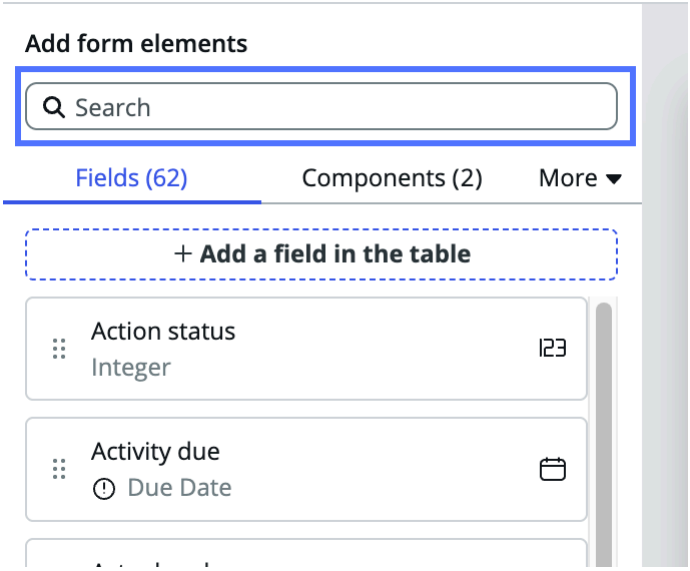


6. Select **Save**.

7. Add pre-existing fields to your table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. Search for the fields that you want to add by entering the names of the fields in the search bar.



In our tutorial, we add the following fields to our table:

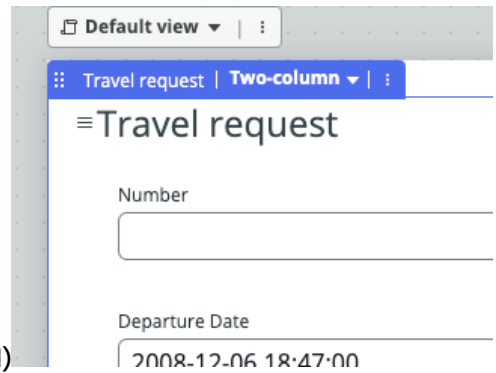
- `Approval`
- `Opened by`

b. Select the field in the **Add form elements** panel, then drag the field into the **Default view** panel.

8. **Optional:** Change the layout of the form.

a. Select your table in the **Default view** panel.
When selected, your table appears with a blue border.

(Optional)



b. In the **Section** customization panel, select **One column**, then select **Two columns** to arrange the form fields evenly into a two-column layout.

> **Tip:** For more information about the customization options available in the **Section** panel, see Customize your form layout in Table Builder.

9. **Optional:** Arrange the table fields into the order that you want them to appear by moving the fields.
In our tutorial, we arrange the fields into the following order:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. Number
b. Opened by
c. Departure Date
d. Return Date
e. Reason for Travel
f. Estimated Airfare
g. Approval
h. Status

10. Select **Save**.
11. **Optional:** Select **Preview** to review how your form appears.

Result
Your table is configured to capture the necessary data for employee travel requests.

What to do next
To complete the data model for the employee travel request application, we must connect the travel request table to a separate table that stores airport data. Proceed to the next step: Share data between tables.

Share data between tables

Share data between tables to avoid data duplication.

Before you begin
Role required: admin, sn_app_eng_studio.user, or delegated developer

Select the following link to download the material needed for this step in the app creation tutorial.

App creation tutorial airport spreadsheet

About this task
The final phase in building the data model for the employee travel request application is to link the travel request table to a separate table that stores airport data. Doing so enables employees to choose which airports they want to use when they complete employee travel requests.

Phase 1	✓	Create a data table
Phase 2	✓	Configure a data table
Phase 3	○	Share data between tables

Storing data in multiple tables enables you to keep your data organized and modular. When you want to share data between tables you can create reference fields, which pull records from one table into another without duplicating the data.

Download the airport spreadsheet and follow along with the tutorial to create references fields between the employee travel request and airport tables.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

https://player.vimeo.com/video/967025312?badge=0&autopause=0&player_id=0&app_id=58479

Video sections

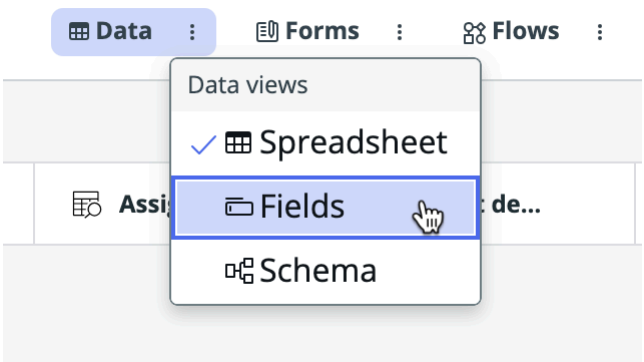
Timestamp	Section
0:03	Create a table to store the data that you want to reference.
0:26	Configure the table.
1:05	Open your existing table.
1:13	Add new fields to the existing table.
1:19	Set the field type to **Reference** and select the table that you want to pull data from.
1:36	Save your changes.
1:38	Add the new reference fields to your table in **Form** view.
1:59	Customize the form layout.
2:10	Save your changes.

Procedure

1. Create a new table to store airport information.
   In our tutorial, we create a new table from a spreadsheet that stores airport information. To follow along with the tutorial, download the airport spreadsheet. Then use the procedures in Import a spreadsheet and Create new table from spreadsheet import to create a new table from the airport spreadsheet.
2. Customize the table so that it displays the information that you want to reference.

   a. Select the airport table from your application dashboard in App Engine Studio.

   b. On the table landing page, select the More actions icon ( ) next to **Data** in the top ribbon, then select **Fields**.



   c. On the **Fields** page, in the **Name** field, toggle on the **Display** switch ( ) to enable other tables to reference airport names.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

> **Important:** You can select only one field to be referenced by other tables. If you attempt to select multiple fields, Table Builder keeps only one of your selections when you save your changes.

d. Select **Save**.

3. Navigate to your application dashboard in App Engine Studio.
4. Select the employee travel request table, where you'll create the reference fields that pull data from the airport table.
5. On the table landing page, select the More actions icon (⋮) next to **Data**, then select **Fields**.
6. Create the reference fields.
   In our tutorial, we create two reference fields to collect airport choices for employee travel requests.

   a. On the **Fields** page, select **+ Add a field in the table**.

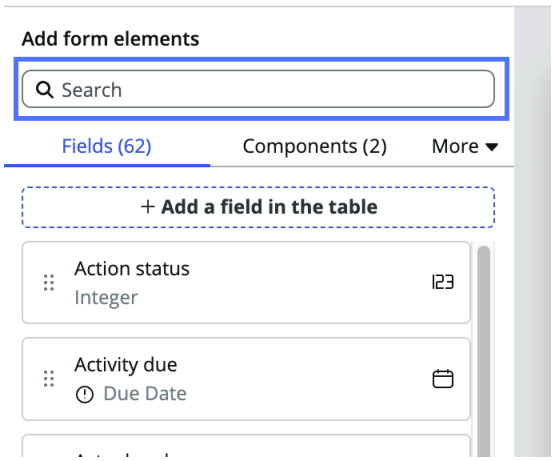
   b. In the **Column label** field, enter `Travel From`.
   The **Column name** field auto-populates based on the label that you entered.

   c. In the **Type** column, select **Reference**, then select the airport table from the list.
   In our tutorial, we create another reference field to capture where the employee is traveling to. We label this field `Travel To`.

7. Select **Save**.
8. Add the reference fields to the form view of the table.

   a. Select **Forms** in the top ribbon.

   b. Enter the label of the reference field in the **Search** bar.
   In our tutorial, the reference fields are labeled `Travel From` and `Travel To`.



   c. Drag the reference fields from the **Add form elements** panel to the **Default view** panel.

9. **Optional:** Repeat the process in steps 6–8 to add as many reference fields as you need.
10. Select **Save**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Result
The data model for the employee travel request application is complete.

What to do next
Continue building the employee travel request application by proceeding to the next step in the app creation tutorial: Creating user experiences.

Creating user experiences

Create the user experiences for your application. Learn about experiences in App Engine Studio and how we build the user experience for the employee travel request application.

We continue building our application by creating user experiences.

This is the fourth step in the app creation tutorial. In this step, you learn about user experiences and add an experience to the employee travel request application.

Step 1		Planning your application
Step 2		Create an app
Step 3		Building a data model
Step 4		Creating user experiences
Step 5		Adding logic and automation
Step 6		Test your application

Overview of user experiences

User experiences, or interfaces, determine the options for users to access and engage with application content. There are several kinds of experiences that you can add to your application, such as record producers and portals. For more information about experiences in App Engine Studio, see Add an application experience.

Tutorial user experience

In our tutorial, we add a record producer, which is one of the application experiences available in App Engine Studio. The record producer creates the form that employees will use to submit travel requests.

We build the experience for our application in two phases:

Phase 1	Add a record producer
Phase 2	Configure a record producer

Next steps

Start building the user experience for the employee travel request application: Add a record producer.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add a record producer

Create a record producer to start building user experiences for the employee travel request application.

Before you begin

Role required: admin or catalog_admin

About this task

The first phase in building the employee travel request application's user experience is to select an experience. In our tutorial, we add a record producer, which enables us to create the form that employees will use to submit travel requests.

Phase 1		Add a record producer
Phase 2		Configure a record producer

This procedure covers the first part of the task demonstrated in the tutorial video. The rest of the procedure is outlined in the next topic Configure a record producer.

Follow along with the tutorial to add a record producer to the employee travel request application.

https://player.vimeo.com/video/953207342?badge=0&autopause=0&player_id=0&app_id=58479

Video sections

Timestamp	Section
0:06	Add an experience.
0:10	Select the record producer experience.
0:14	Enter a name and description for the record producer.
0:20	Create the record producer.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. Select the travel request application.
3. On the application dashboard, select **+ Add** next to **Experience**.
4. Select **Record producer**, then select **Begin**.
5. Define the properties of the record producer.
   a. In the **Name** field, enter `Raise a travel request`.
   b. **Optional:** In the **Short description** field, enter the following description of the record producer: `This request is for employees who are seeking approvals for travel requests`.
6. Select **Continue**.
7. Select **Edit record producer** to continue setting up your record producer.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Result
You have added a record producer to your application. The record producer is ready for configurations.

What to do next
Complete the setup of the record producer by proceeding to the next phase: Configure a record producer.

Configure a record producer

Configure the record producer so that it contains the necessary questions for employee travel requests.

Before you begin
Role required: admin or catalog_admin

About this task
The second phase in building the user experience for the employee travel request application is to configure the record producer. We configure the record producer so that it contains the necessary questions for employee travel requests, such as travel dates, location, and estimated airfare.

Phase 1		Add a record producer
Phase 2		Configure a record producer

This procedure covers the second part of the task demonstrated in the tutorial video. The first part of the procedure is outlined in the previous topic Add a record producer.

Follow along to configure the record producer for the employee travel request application.

https://player.vimeo.com/video/953207342?badge=0&autopause=0&player_id=0&app_id=58479

Video sections

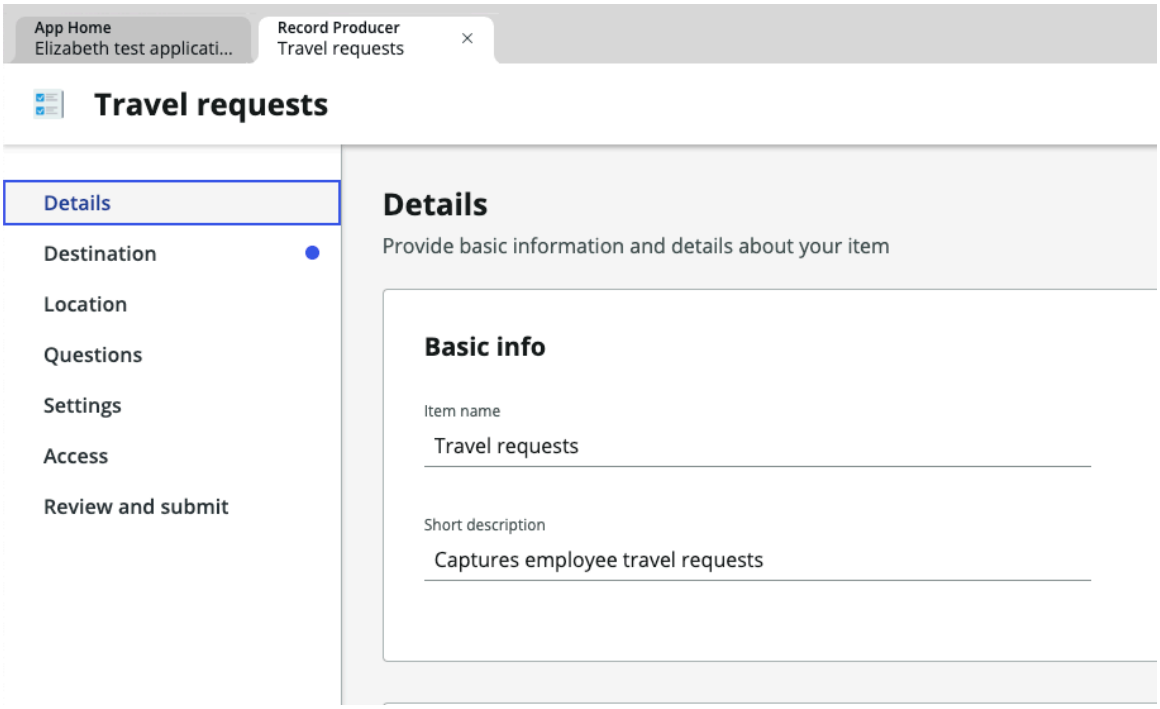
Timestamp	Section
0:21	Specify a destination table.
0:32	Choose the location.
0:42	Create question containers.
1:01	Insert questions.
3:49	Preview the record producer.
3:58	Configure settings.
4:05	Select user access.
4:10	Review and submit the record producer.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Ensure that the travel request record producer is open.
   - To open a record producer from your instance home page, navigate to **All** > *App Engine* > *App Engine Studio*. Then select your application and select the record producer that you want to configure.
   - To open a record producer that you have created, select **Edit record producer**.

   When your record producer is open, it appears in the canvas space.



2. **Optional:** Edit or add details about the record producer in the **Item name** or **Short description** fields.
3. Select **Continue to Destination** to specify the table in which the record producer creates records.
4. Enter `Travel request` in the **Record submission table** field, then select the employee travel request table from the list.
   By setting the travel request table as the destination table, travel requests submitted through the record producer become records in the employee travel request table.
5. Select **Continue to location** to specify the location of your record producer.

   Note: Location determines where users can find the record producer. You can locate your record producer in a catalog, such as the Service Catalog, which is a set of business and technical products, services, and offerings that users can order. You can also locate your record producer with a topic, such as a knowledge base article. For more information about locations for record producers, see Creating or editing catalog item template.

   In our tutorial, we locate our record producer in the Service Catalog, so that employees can access the employee travel request form easily.
6. In the **Catalogs** card, select **Browse**.
7. From the **Available options** list, select **Service Catalog** and use the move icon (>) to add it to the **Selected catalogs** list.
8. Select **Save selections**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

9. Select **Continue to questions** to start building the form questions.
10. Create containers, or sections, for the form questions.
    a. Select the arrow icon () in the **Insert new question** button to expand the menu of available question types.
    b. Select **Single column container** from the list.
    c. In the **Title** field, enter `General Information`.
    d. Select **Submit**.
    e. Repeat the procedure in steps a-d to create as many containers as needed.
    In our tutorial, we create another container for our form questions with the following specifications:

Container type	Title
Two column container	`Travel Details`

11. Add a question to the form.

> **Note:** You can build different types of questions for your record producer. To see the full list of available question types in Catalog Builder, see Create a question for a catalog item in Catalog Builder.

In our tutorial, we create several questions to collect details about employees' travel requests.

a. In the correct question container, hover over the plus icon (⊕), then select **+ Insert**.
In our tutorial, we insert this question in the General Information container.

b. Select **New question**.

c. Select and enter the following values for the new question.

Question type	Question subtype	Map to specific field on table	Table field	Question label	Name	Mandatory
Choice	**Dropdown (fixed values)**	Yes	**Reason for travel**	`What is the reason for travel?`	`reason_for_travel`	Yes

d. Select **Continue to Choices** to complete additional configurations for the question.

e. In the **Available choices** section, hover over the plus icon (⊕), then select **+ Insert**.

f. In the **Display name** field, enter `Customer meeting`.

g. Repeat the process in steps e and f to add additional choices.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

In our tutorial, we add two additional choices. The following list includes the names for each additional choice that we add:

- `Internal meeting`
- `Training`

h. Select **Insert question**.

12. Repeat the procedure outlined in step 11 to create as many questions as needed.
In our tutorial, we add the following additional questions to our form in the Travel Details container:

a. We add a question to determine the employee's departure date.

Question type	Question subtype	Map to specific field on table	Table field	Question label	Name	Mandatory	Additional configuratio
Date / Time	**Date**	Yes	**Departure date**	`When are you leaving?`	`departure_date`	Yes	None, select **Insert question**.

b. We add a question to determine the employee's return date.

Question type	Question subtype	Map to specific field on table	Table field	Question label	Name	Mandatory	Additional configuratio
Date / Time	**Date**	Yes	**Return date**	`When are you returning?`	`return_date`	Yes	None, select **Insert question**.

c. We add a question to determine where the employee is traveling from.

Question type	Question subtype	Map to specific field on table	Table field	Question label	Name	Mandatory	Additional configuratio
Choice	**Record reference**	Yes	**Travel from**	`Where are you traveling from?`	`travel_from`	Yes	Select **Source table** > **Airport**.

d. We add a question to determine where the employee is traveling to.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Question type	Question subtype	Map to specific field on table	Table field	Question label	Name	Mandatory	Additional configuratio
Choice	**Record reference**	Yes	**Travel to**	`Where are you traveling to?`	`travel_to`	Yes	Select **Source table** > **Airport**.

e. We add a question to determine the estimated cost of the travel.

Question type	Question subtype	Map to specific field on table	Table field	Question label	Name	Mandatory	Additional configuratio
Text	**Single line**	Yes	**Estimated airfare**	`What is your estimated airfare?`	`estimated_airfare`	Yes	Select **Text validation** > **Number**.

13. **Optional:** Preview your form by selecting **Preview**.
14. When you're ready to finalize your form, select either **Continue to settings** or the **Settings** tab in the side panel.
15. Select the check box next to any of the following portal settings.
    In our tutorial, we select the check box next to **Hide 'Add to wishlist' button** and **Hide attachment button**. You can select or deselect these settings in your record producer.
    - To prevent users from saving your record producer item to their wish list, select the check box next to **Hide 'Add to wishlist' button**.
    - To hide the component that enables users to provide attachments in the record producer form, select the check box next to **Hide attachment button**.
    - To require users to provide attachments, such as receipts or screenshots for service requests, select the check box next to **Make attachment mandatory**.
16. Select **Continue to access**.
17. **Optional:** Choose from the following options to define which users or groups have access to the form.

    Important: If you don't select any users or groups to grant or deny access, your form is available to everyone.

    In our tutorial, we don't grant or deny access to any users or groups.
    - To make the form available to specific users or groups, select **Browse** in the **Available for** card, then search for the users or groups that you want to grant access to. Move the users or groups from the **Available options** list to the **User criteria granted access** list using the move icon ( ).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- To make the form unavailable for certain users or groups, select **Browse** in the **Unavailable for** card, then search for the users or groups that you want to deny access to. Move the users or groups from the **Available options** list to the **User criteria denied access** list using the move icon ( ).

18. Select **Continue to review and submit** to review and finalize your form.
19. Select **Submit**.

Result
Your record producer is configured and ready for testing.

What to do next
Continue building the employee travel request application by proceeding to the next step: Adding logic and automation.

Adding logic and automation

Add logic and automation to your application. Learn about logic and automation in App Engine Studio and how we automate the employee travel request application.

We continue building the employee travel request application by adding logic and automation.

This is the fifth step in the app creation tutorial. In this step, you learn about logic and automation in App Engine Studio. You build a decision table and create a flow for the employee travel request application.

Step 1	✓	Planning your application
Step 2	✓	Create an app
Step 3	✓	Building a data model
Step 4	✓	Creating user experiences
Step 5	○	Adding logic and automation
Step 6	○	Test your application

Overview of logic and automation

Logic and automation enable you to define the actions that your application performs and the conditions that trigger those actions. Logic helps you define when certain actions should occur. For example, your application logic can determine whether support tickets are marked as "critical." Automation then defines the actions that the application performs. For example, if support tickets are marked as "critical", your application automation can move those tickets to the top of the queue.

There are several ways to implement logic and automation in App Engine Studio. Our tutorial focuses on decision tables and flows. To learn about the full range of logic and automation capabilities in App Engine Studio, see Add logic and automation.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Tutorial logic and automation

In our tutorial, we create a decision table and a flow to store the logic and automation for the employee travel request application. Decision tables enable you to set and manage business logic in your application separate from your code. The decision table in our tutorial determines which regional heads receive travel requests based on employees' geographic location.

Flows are automated processes that use a trigger, actions, and logic to define the behavior of an application. The flow in our tutorial defines the sequence of actions that the employee travel request application completes once a travel request is created.

We add logic and automation to our application in two phases:

Phase 1	Build a decision table
Phase 2	Create a flow

Next steps

Start adding logic and automation to the employee travel request application: Build a decision table.

Build a decision table

Build a decision table for the employee travel request application that determines which regional head to route travel requests to.

Before you begin

Role required: admin, decision_table_admin, or delegated developer permissions

About this task

The first phase in adding logic and automation to the employee travel request application is to build a decision table. Our decision table evaluates where an employee is located and uses that information to determine which regional head to route their travel request to.

Phase 1		Build a decision table
Phase 2		Create a flow

For more information about using decision tables, see Create decision tables in Workflow Studio.

Follow along with the tutorial to build a decision table for the employee travel request application.

https://player.vimeo.com/video/953653842?badge=0&autopause=0&player_id=0&app_id=58479

Video sections

Timestamp	Section
0:05	Create a decision table.
0:12	Create a new decision table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Video sections (continued)

Timestamp	Section
0:13	Enter a unique name and choose the scope for the decision table.
0:28	Add an input.
0:36	Add a condition column.
0:52	Add conditions.
0:59	Add a result column.
1:09	Add results.
1:16	Add more conditions.
1:34	Save your changes.
1:36	Verify that the decision table has been added to your application.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. Select the travel request application.
3. On the application dashboard, select **+ Add** next to **Logic and automation**.
4. Select **Decision**, then select **Begin**.
5. Select **Create a new decision table**.
6. Define the properties of the decision table.
   a. Enter `Regional head approvals` in the **Name** field.
   b. In the **Accessible from** field, select **This application scope only** to have the logic in the decision table apply to only the selected application.
   If you want the logic in your decision table to apply to all applications, select **All application scopes**.
   c. Select **Continue**.
7. Select **Edit decision table** to continue setting up your decision table.
8. Add an input to the decision table.

   Note: Inputs define the data that is evaluated in the decision table.

   In our tutorial, we set the input as the travel request table so that the decision table can evaluate employee information associate with each travel request.

   a. Select **+ Add an input**.
   b. In the **Type** field, select **Reference**, then select the travel request table from the list.
   c. In the **Label** field, enter `Employee travel requests`.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

d. **Optional:** Repeat this procedure to create as many inputs as you need for your decision table.
In our tutorial, we don't create additional inputs.

9. Add a condition column to the decision table.

> **Note:** Condition columns act as filters that refine the input data by specifying which values should be evaluated by the decision table.

In our tutorial, the condition column refines the input data to evaluate only the employees' country codes. Employees' country codes determine which regional head their travel requests should route to.

a. Select **+ Add condition column**.

b. In the **Condition column label**, enter `Employee regions`.

c. In the **Input** field, select the travel request table input.

d. For **Data to evaluate**, select **Field**.

e. In the **Field** field, select **Opened by** > **Country code**.

f. In the **Default operator** field, select **is one of**.

g. Select **Done**.

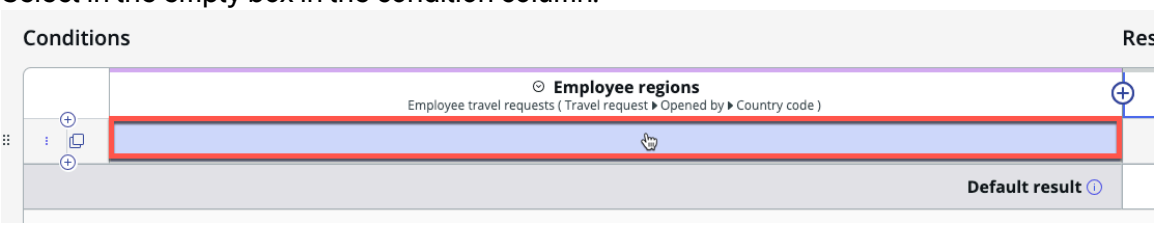
h. **Optional:** Repeat this procedure to create as many condition columns as you need for your decision table.
In our tutorial, we don't create additional condition columns.

10. Define individual conditions for the condition column.

> **Note:** Conditions represent the individual values that the decision table evaluates to return results.

In our tutorial, the conditions that we define represent each country that corresponds to a different regional head. We define individual conditions for the countries that make up the Americas (AMS) region, the Asia Pacific (APAC) region, and the Europe, Middle East, and Africa (EMEA) region.

a. Select in the empty box in the condition column.



b. Select **is**, then select **is one of** from the list.

c. In the empty space beneath **is one of**, select **Brazil** and **United States**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

d. Select **OK**.

e. Select **+ Add a new decision row** to add additional conditions to the condition column.

f. **Optional:** Repeat the process in steps a-e to define additional conditions.
In our tutorial, we define two additional conditions for the countries that make up the APAC and EMEA regions. The following table outlines the condition column values that we set:

Operator	Value
is one of	▪ **China** ▪ **Japan**
is one of	▪ **France** ▪ **Germany** ▪ **Italy** ▪ **Spain** ▪ **United Kingdom**

11. Add a result column to the decision table.

Note: Result columns determine the type of data to return after evaluating the input and condition values.

In our tutorial, the result column determines which regional head to route travel requests to. As the regional heads are individual users in the system, the result column references the User [sys_user] table.

a. Select **+ Add result column**.

b. In the **Result column label** field, enter `Regional heads`.

c. **Optional:** In the **Description** field, enter a description for the result column.

d. In the **Type** field, enter `Reference`, then select **Reference** from the list.

e. In the **Reference table** field, enter `User`, then select **User [sys_user]** from the list.

f. Select **Done**.

g. **Optional:** If you want to define additional results for your decision table, repeat this procedure to create as many result columns as needed.
In our tutorial, we don't create additional result columns.

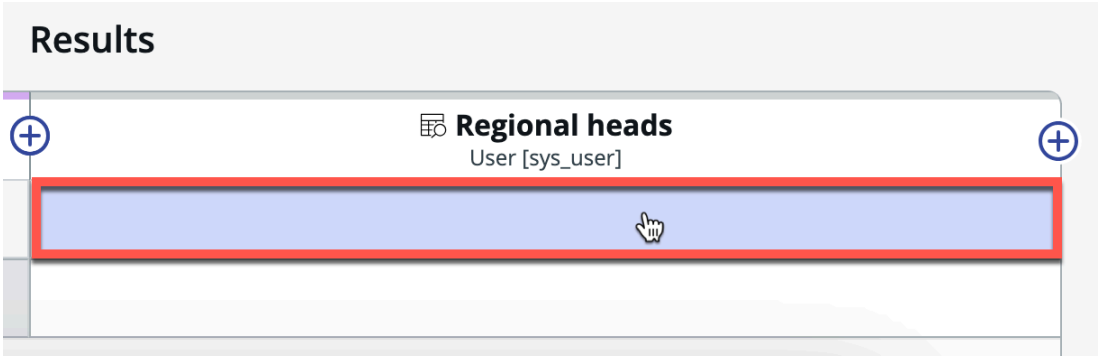
12. Define the results for the result column.

Note: Results are the individual values, or decisions, that are generated based on inputs, conditions, and results defined in the decision table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

In our tutorial, the results represent the regional heads for each region.

a. Select in the empty box in the result column.



b. In the empty field, enter the name of the regional head.
For the Americas (AMS) region, enter `Lucius Bagnoli`.

c. Select the regional head from the list, then select the Enter key.

d. Optional: Repeat the process in steps a-c to create as many results as needed for the decision table.
In our tutorial, we create two additional results for the APAC and EMEA regional heads. The following table outlines the regional head user information for each region:

Region	User
Asia Pacific (APAC)	`Rob Phil`
Europe, Middle East, and Africa (EMEA)	`Abel Tuter`

13. Select **Save**.
14. **Optional:** Verify that your decision table has been added to your application.
    a. Navigate to your application home page.
    b. Confirm that your decision table appears in the **Logic and automation** section.

Result
You have built a decision table for your application.

What to do next
Proceed to the next phase in adding logic and automation to the employee travel request application: Create a flow.

Create a flow

Automate the workflow for the employee travel request application by creating a flow.

Before you begin
Role required: admin or flow_designer

About this task
The second phase in adding logic and automation to the employee travel request application is to create a flow. The flow that we create automates the workflow for the application after an employee travel request is created.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Phase 1	✓	Build a decision table
Phase 2	○	Create a flow

For more information about flows, see Building flows.

Use the following tutorial to create a flow for the employee travel request application.

https://player.vimeo.com/video/953653961?badge=0&autopause=0&player_id=0&app_id=58479

Video sections

Timestamp	Section
0:09	Add a flow to your application.
0:14	Build the flow from scratch.
0:17	Enter a unique name and description for the flow.
0:22	Continue setting up your flow.
0:25	Create a trigger.
0:41	Add an Ask for Approval action.
1:09	Add If flow logic.
1:25	Integrate decision tables.
1:48	Duplicate repeating actions.
2:17	Add an Update Record action.
2:38	Add Else flow logic.
2:50	Add a Send Email action.
4:14	Add End Flow flow logic.
4:19	View your flow as a diagram.
4:22	Save your changes.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. Open the employee travel request application.
3. On the application dashboard, select **+ Add** next to **Logic and automation**.
4. Select **Flow**, then select **Begin**.
5. Select **Build your flow from scratch**.
   You can also choose to build your flow with Now Assist. For more information, see Create a flow with AI.
6. Define the properties of the flow.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. In the **Name** field, enter `Approvals workflow`.

b. In the **Description** field, enter `Approval workflow for managers and regional heads`.

c. Select **Continue**.

7. Select **Edit this flow** to continue setting up your flow.

8. Add triggers to the flow.

> **Note:** A trigger specifies when to run the flow. When the trigger conditions are met, the system runs the flow using the data provided by the trigger.

In our tutorial, we create a trigger that runs the flow whenever an employee submits a travel request.

a. In the **Trigger** section, select **+ Add a trigger**.

b. In the **Trigger** field, select **Record** > **Created**.
For more information about trigger types, see Workflow Studio flow trigger types.

c. In the **Table** field, enter the name of the travel request table and select it from the list.

d. Select **Done**.

9. Add actions, flow logic, and sublows to the flow.

> **Note:** You can add actions, flow logic, and subflows to enable your flow to perform desired actions. For example, you can create an action that enables your application to send emails automatically.

Instructions for adding the actions and flow logic demonstrated in the tutorial are outlined in the following topics.

10. **Optional:** See how your flow appears as a diagram by toggling the switch next to **View**.

11. Select **Save** to save your changes.

Result

You have created a flow that will automate the end-to-end workflow for the employee travel request application.

What to do next

Continue building the flow for the employee travel request application. Proceed to the next step: Add an Ask for Approval action.

Add an Ask for Approval action

Request approvals for any record by adding the Ask for Approval action to your flow.

Before you begin

Role required: admin or flow_designer

About this task

The Ask for Approval action is a ServiceNow Core action template that enables you to request approvals easily. You can specify which records require approval and assign users to approve or reject the record.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

In our tutorial, we use the Ask for Approval action to request approvals from both managers and regional head for employee travel requests. This procedure outlines the Ask for Approval action that routes to the manager. The procedure for creating the Ask for Approval action that routes to the regional head is outlined in Duplicate repeating actions.

For more information about the Ask for Approval action, see Ask for Approval action.

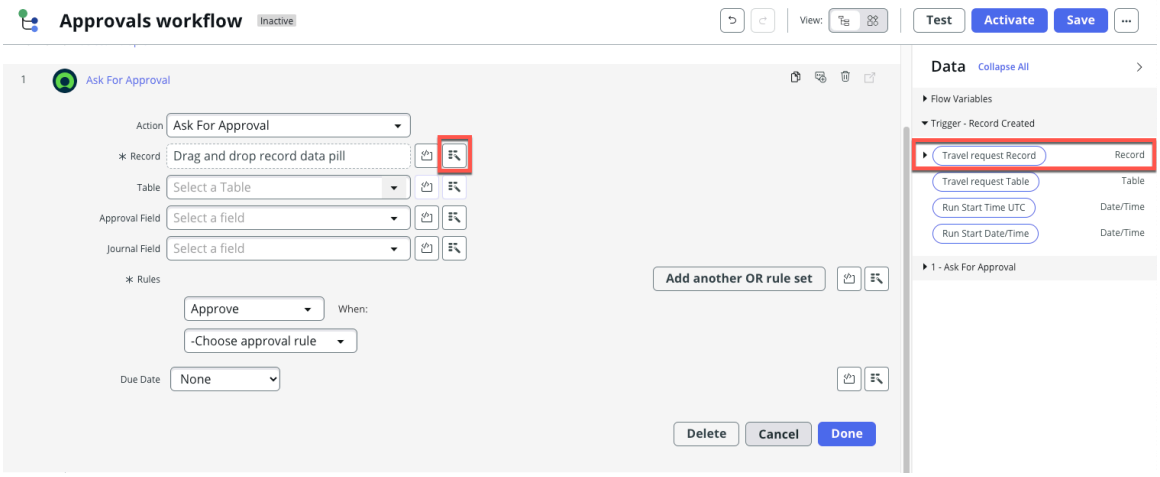
Procedure

1. At the correct location in your flow, select **+ Add an Action, Flow Logic, or Subflow**.
   In our tutorial, the Ask for Approval action that routes to the manager occurs just after the trigger. To see where the Ask for Approval action occurs in the tutorial flow, see the diagram in Create a flow.
2. Select **Action**.
3. From the **ServiceNow Core** options, select **Ask for Approval**.
4. Choose from one of the following options to connect the action to the appropriate trigger.

   Note: Some flows have multiple triggers, so you must specify which trigger corresponds to the action.

   In our tutorial, we add the travel request record trigger to the **Record** field, so that the action runs whenever an employee submits a travel request.

   - To add your trigger within the Ask for Approval dialog, select the **Data Pill Picker** (), then select the trigger from the list.
   - To add your trigger from the **Data** panel, drag the data pill that contains the trigger into the **Record** field.



5. Set the rules for the action.

   Note: Rules determine which users can approve or reject requests, and what happens after approval or rejection. For more information about the rules available for the Ask for Approval action, see Ask for Approval action.

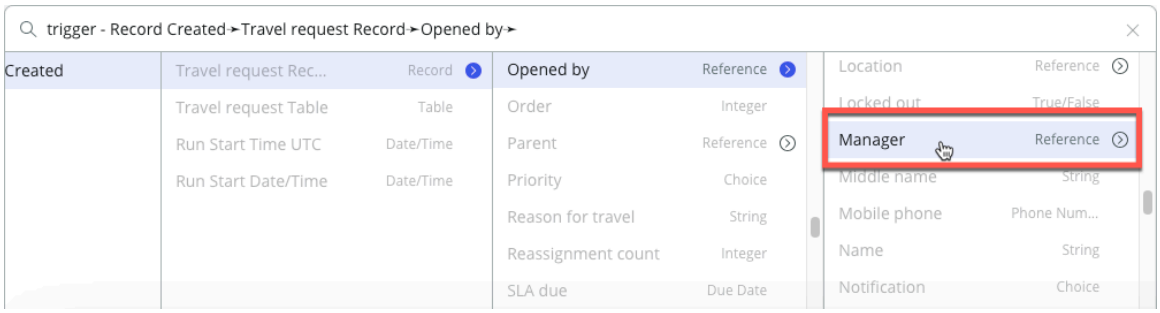
   In our tutorial, we configure the rules for the Ask for Approval action to send travel requests to the manager of the user who submitted the request. The manager can then approve or deny the request.

   a. Select the **Approve** field, then select **Approve or Reject** from the list.

   b. Select the **-Choose approval rule** field, then select **All users approve or reject**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

c. Choose which users or groups can complete the approval actions using the **Data Pill Picker** ( ), the add user icon ( ), add group icon ( ), the add manual approvers icon ( ), or by dragging data pills from the **Data** panel into the field.
In our tutorial, we want the Ask for Approval action to route to the manager of the user who submitted the travel request. So we select **Trigger - Record Created** > **Travel request Record** > **Opened by** > **Manager**.



The second Ask for Approval action, which you create in Duplicate repeating actions, routes the approval to the regional head.

6. Select **Done**.
7. Select **Save**.

What to do next
Proceed to the next step: Add If flow logic.

Add If flow logic

Use If flow logic to direct your flow to perform actions based on specific conditions.

Before you begin
Role required: admin or flow_designer

About this task
Add If flow logic to your flow to have your application perform actions only when certain criteria are met. If flow logic creates a specific path, or branch, within a flow that runs only when the conditions are met.

In our tutorial, we use If flow logic to define what the application does if a travel request has been approved. For example, if a travel request has been approved by the regional head, the **State** of the travel request record should be changed to **Closed complete**.

There are two instances of If flow logic in the flow in the app creation tutorial: One that represents manager approval and another that represents regional head approval. This procedure covers creating the branch that represents manager approval. You can use this procedure to create the branch that represents the regional head approval and adjust the **Column label** and **Condition** field values accordingly.

For more information about using If flow logic, see If flow logic .

Procedure

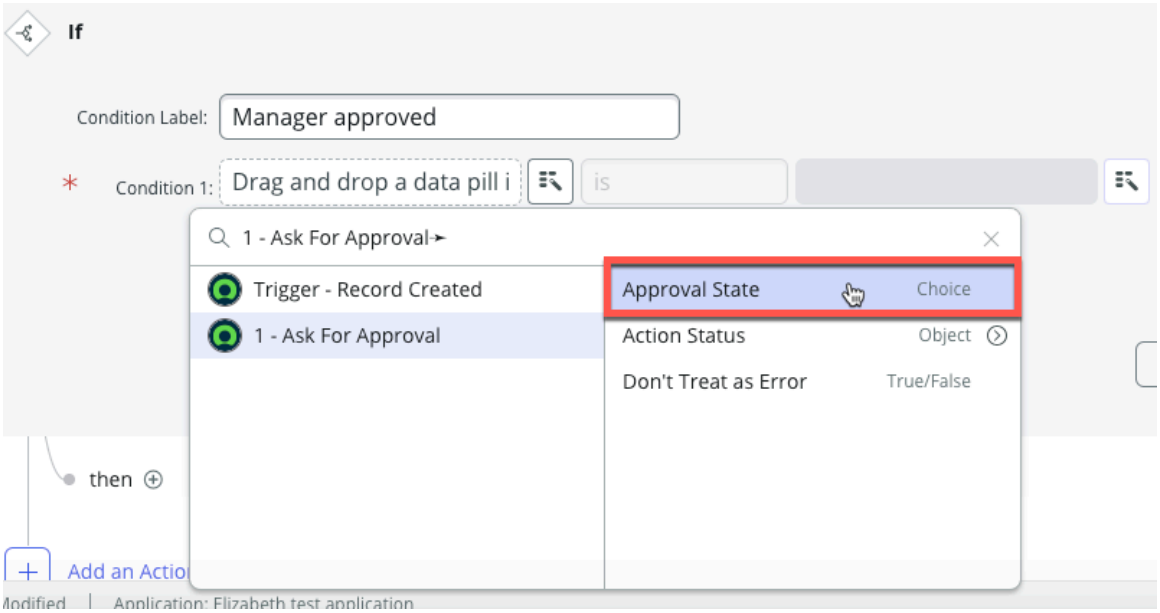
1. At the correct location in your flow, select **+ Add an Action, Flow Logic, or Subflow**.
The If flow logic that represents manager approval occurs just after the first Ask for Approval action. To see where If flow logic occurs in the tutorial flow, see the diagram in Create a flow.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

2. Select **Flow Logic**.
3. Select **If**.
4. **Optional:** In the **Condition label** field, enter a description to describe the conditions that the branch represents.
   In our tutorial, we enter `Manager Approved`, as this branch represents that a manager has approved the travel request.
5. Select the condition under which the branch runs using the **Data Pill Picker** ( ) or by dragging the data pill from the **Data** panel into the **Condition 1** field.

   Note: The condition is the data container that the flow evaluates when determining whether to run the branch. You can define one or multiple conditions in If flow logic.

   In our tutorial, we want this branch of the flow to represent what happens if a manager has approved a travel request, so we set the condition to **Approval State**.



6. Define the condition state that enables the branch to run by selecting the fields beside **Condition 1**, then choosing the appropriate values.

   Note: The condition state is the actual value of the condition that defines when the branch should run.

   In our case, this branch should run when a travel request has been approved by a manager. So we set the condition state to **is** and **Approved**.
7. Select **Done**.
8. Select **Save**.

Result

By default, If flow logic has no output. It simply evaluates whether a condition is met or unmet. To have your flow complete actions when the conditions in the If flow logic are met, you must add branching actions or flow logic.

What to do next

Proceed to the next step: Integrate decision tables.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Integrate decision tables

Integrate decision tables into your flow using Make a decision flow logic.

Before you begin

Role required: admin or flow_designer

About this task

Make a decision flow logic enables you to integrate decision tables within flows easily. Make a decision flow logic runs the decision table and returns results, which can be displayed as branches in the flow or data pills in the **Data** panel.

In our tutorial, employee travel requests are sent to the regional heads after receiving manager approval. We use Make a decision flow logic to connect the flow to the regional head decision table and retrieve the correct regional information.

For more information about Make a decision flow logic, see Make a decision flow logic.

Procedure

1. At the correct location in your flow, select **+ Add an Action, Flow Logic, or Subflow**.
   We add Make a decision flow logic beneath the branch of If flow logic that represents manager approval. To see where we add Make a decision flow logic in the tutorial flow, see the diagram in Create a flow.
2. Select **Flow logic**.
3. Select **Make a decision**.
4. Select the **Decision table** field, then select the correct decision table from the list.
   In our tutorial, we select the regional head approvals decision table.
5. Choose to select or deselect the check box for **Use Branches**.

   Note: If you select the check box for **Use Branches**, the flow displays each possible decision table result as a separate branch.

   In our tutorial, the approval process is the same for each of the regional heads, so we deselect **Use Branches**. If the employee travel request process was different for employees of a certain region, we could select **Use Branches** to represent the different processes based on region.
6. Add the correct trigger to the **Decision table inputs** field using either the **Data Pill Picker**( ) or by dragging the data pill from the **Data** panel into the field.

   Tip: When you integrate a decision table into a flow, you must specify the data values for the decision table to evaluate within the context of the flow.

   In our tutorial, we need the decision table to evaluate the user information that's associated with the travel request trigger. So we add our trigger to the **Decision table inputs** field.
7. Select **Done**.

Result

The Make a decision flow logic generates results from the decision table, which you can use in your flow.

What to do next

Proceed to the next step: Duplicate repeating actions.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Duplicate repeating actions

For actions that repeat within your flow, save time by duplicating the actions and adjusting their values as needed.

Before you begin

Role required: admin or flow_designer

About this task

Duplicating actions enables you to build flows more quickly and efficiently.

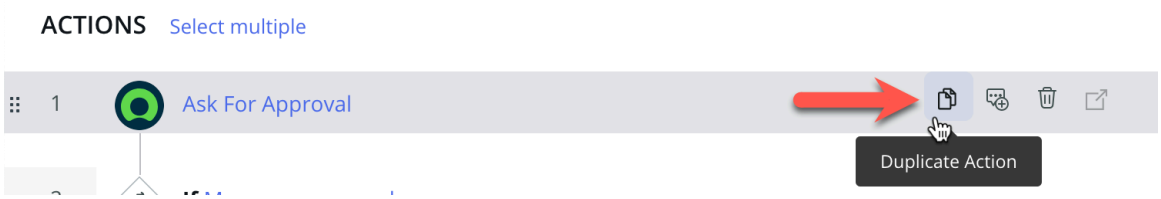
In our tutorial, there are several actions that repeat within the employee travel request flow.

- The Ask for Approval action repeats to request both the manager and regional head approvals for employee travel requests.
- The Update Record action updates the travel request record at several places in the flow, such as when the request is rejected by a manager.
- The Send Email action occurs in two places to notify employees that their travel requests have been rejected by either the manager or the regional head.

This procedure outlines duplicating the Ask for Approval action to request regional head approval. You can use the steps outlined here to duplicate the other actions that repeat and adjust the action location and field values accordingly.

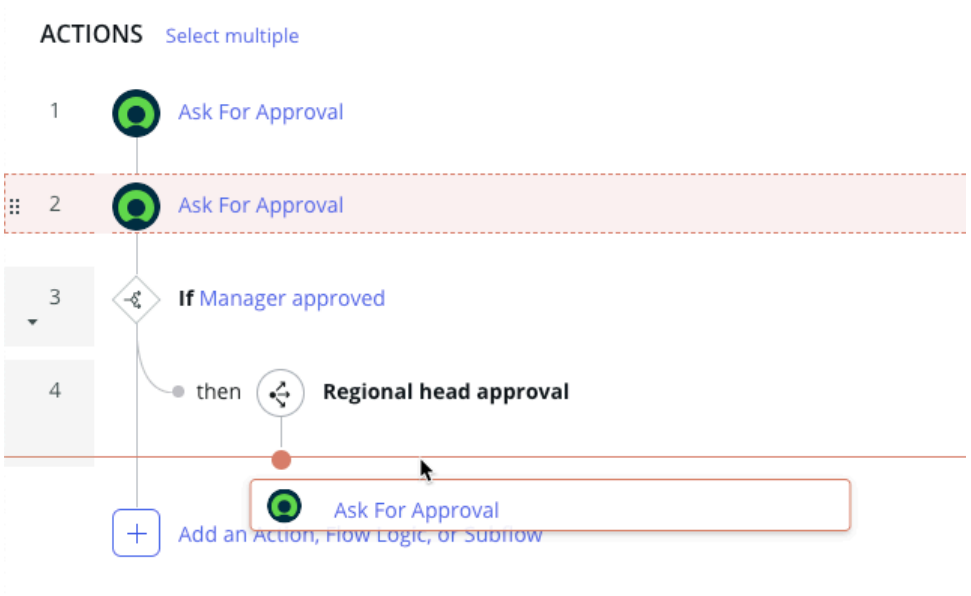
Procedure

1. Hover over the action that you want to duplicate and select the Duplicate action icon ( ).
   In our tutorial, we duplicate the Ask for Approval action.



2. Move the duplicated action into the correct place in the flow by hovering over the action, selecting the drag icon ( ), and dragging the action into the correct place.
   We move the duplicated Ask for Approval action beneath the Make a decision flow logic that returns the regional head information.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



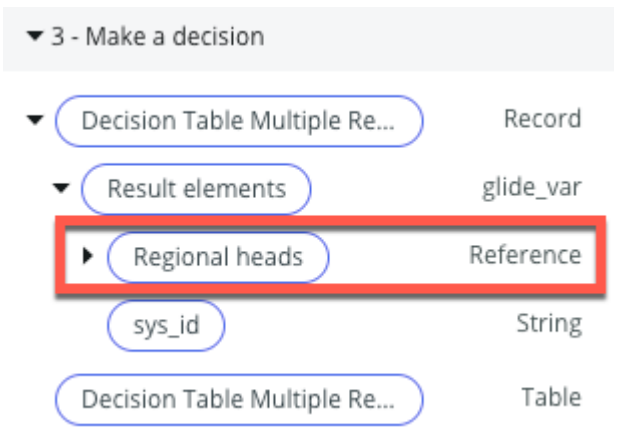
3. Adjust values in the action as needed.

a. Select the duplicated action to open the action dialog.

b. Select the remove icon ( ) to remove any field values that you must change.
In our tutorial, we must change the user that the Ask for Approval routes to. So we select the remove icon ( ) in the data pill that contains the manager user information.

c. Provide the correct field values by selecting the **Data Pill Picker** ( ) or by dragging data pills from the **Data** panel into the fields.
To route the Ask for Approval action to the correct regional head, we must add the data pill that contains the regional head information.

- Using the **Data Pill Picker** ( ), we select **Make a decision** > **Decision Table Multiple Result Record** > **Result elements** > **Regional heads**.
- Using the **Data** panel, we select and drag the regional heads data pill into the field.



4. Select **Save**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

5. **Optional:** Repeat this procedure to duplicate as many actions as needed in your flow.
   In our tutorial, we also duplicate the Update Record and Send Email actions.

What to do next
Proceed to the next step: Add an Update Record action.

Add an Update Record action

Automate record updates in a flow using the Update Record action.

Before you begin
Role required: admin or flow_designer

About this task
The Update Record action is a ServiceNow Core action template that you can add to your flow to update existing records.

In our tutorial, we use the Update Record action to update the travel request record after it has been approved or rejected. This procedure outlines creating the Update Record action if a travel request has been approved by the manager and regional head. You can use this procedure to create the other Update Record actions and change the fields and field values as needed.

For more information about the Update Record action, see Update Record action.

Procedure
1. At the correct location in your flow, select **+ Add an Action, Flow Logic, or Subflow**.
   We add the Update Record action that represents manager and regional head approvals beneath the If flow logic branch for regional head approval. To see where we add the Update Record action in the tutorial flow, see the diagram in Create a flow.
2. Select **Action**.
3. From the **ServiceNow Core** options, select **Update Record**.
4. Select the record that you want to update by selecting the **Data Pill Picker** ( ) or by dragging the data pill from the **Data** panel into the **Record** field.
   In our tutorial, we want to update the travel request record, so we add the travel request trigger to the **Record** field.
5. Choose the field or fields that you want to update in the record by selecting **Add a field value**, then selecting the field that you want to update from the list.
   In our tutorial, we must update several fields in our travel request record after the request has been approved by the manager and regional head. We select the following fields:
   - **Approval**
   - **Assignment group**
   - **State**
6. Set the desired value of each field that you added by selecting anywhere in the **Select** field, then selecting the correct value from the list.
   In our tutorial, we set the following values for the fields that we added:

Field	Value
Approval	Approved
Assignment group	Travel desk
State	Closed complete

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

7. Select **Done**.
8. Select **Save**.

What to do next
Proceed to the next step: Add Else flow logic.

Add Else flow logic

In flows with If flow logic, add Else flow logic to create alternate paths when conditions aren't met.

Before you begin
Role required: admin or flow_designer

About this task
Add Else flow logic to define the actions that your flow performs when If flow logic conditions aren't met. Else flow logic creates a path in the flow that runs only when the If flow logic conditions aren't met.

In our tutorial, we add Else flow logic to specify what happens when a travel request is rejected by either a manager or regional head.

Procedure
1. At the correct location in your flow, select **+ Add an Action, Flow Logic, or Subflow**.
   We add Else flow logic at two places in the flow: One beneath the branch of the flow that represents manager approval and another beneath the branch that represents regional head approval. To see where we add Else flow logic in the tutorial flow, see the diagram in Create a flow.
2. Select **Flow logic**.
3. In the **Search flow logics** list, select **Else**.
4. Select **Save**.

Result
By default, Else flow logic has no output. It just creates a branch for the flow to run when the If flow logic conditions aren't met. To have your flow complete actions in the Else flow logic block, you must add branching actions or flow logic.

What to do next
Proceed to the next step: Add a Send Email action.

Add a Send Email action

Automate sending email notifications using the Send Email action.

Before you begin
Role required: admin

About this task
The Send Email action is a ServiceNow Core action template that you can add to your flow to automate sending emails to specified users or groups.

In our tutorial, we add Send Email actions to notify employees that their travel requests have been rejected. This procedure covers the Send Email action that notifies employees that their travel request has been rejected by the regional head. You can use this procedure to create the other Send Email action for the manager rejection and adjust the field values accordingly.

For more information about the Send Email action, see Send Email action.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. At the correct location in your flow, select **+ Add an Action, Flow Logic, or Subflow**.
   We add the Send Email action beneath the Else flow logic that represents regional head rejection. To see where we add Send Email actions in the tutorial flow, see the diagram in Create a flow.
2. Select **Action**.
3. From the **ServiceNow Core** options, select **Send Email**.
4. Add the record that the email is associated with to the **Target Record** field using the Data pill picker ( ), or by dragging the data pill from the **Data** panel.
   In our tutorial, we want the email to be associated with the travel request record in our trigger.
5. Add recipients for the email in the **To** field using the Data pill picker ( ), or by dragging the data pill from the **Data** panel.
   In our tutorial, we want the user who created the travel request to receive the email. The user information is stored under **Trigger - Record Created** > **Travel request Record** > **Opened by**.
6. **Optional:** Add additional recipients to the email in the **CC** and **BCC** fields.
   We don't add additional recipients to the Send Email action in our tutorial.
7. Enter a subject for the email in the **Subject** field.
   In our tutorial, we enter the following subject for the email: `Travel request rejected`.
8. Enter the content for the email message in the **Body** field.

   Tip: You can add dynamically created content, or content that is generated within the flow, to the Send Email action. To add dynamic content, drag the data pills from the **Data** panel into the **Body** of the email.

   In our tutorial, the body of our email notifies the employee that their travel request has been rejected and to contact their manager for additional information.
9. Select **Done**.
10. Select **Save**.

What to do next

Proceed to the next step: Add End Flow flow logic.

Add End Flow flow logic

Add End Flow flow logic to stop running the current flow.

Before you begin

Role required: admin

About this task

If you want your flow to end when certain conditions are met, you can add End Flow flow logic. End Flow flow logic can be contained within a conditional flow logic block, such as an If, Else If, or Else flow logic. End Flow flow logic stops running the current flow.

In our tutorial, we add End Flow flow logic to end the flow if a travel request has been rejected.

For more information about End Flow flow logic, see End Flow flow logic .

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. At the correct location in your flow, select **+ Add an Action, Flow Logic, or Subflow**.
   We add End Flow flow logic in the branch that represents manager rejection of a travel request.
   To see where we add End Flow logic in the tutorial flow, see the diagram in Create a flow.
2. Select **Flow Logic**.
3. In the **Search flow logics** list, select **End Flow**.
4. Select **Save**.

What to do next

The logic and automation for the employee travel request application are complete. Proceed to the final step in the app creation tutorial: Test your application.

Test your application

Verify that your application works as intended before publishing by testing the application experience and flow.

Before you begin

Role required: admin or flow_designer

About this task

The final step in building an application is testing. There are several ways to test an application, including simulating the user experience and running flow executions to verify that everything works as expected.

This is the last step in the app creation tutorial. In this step, we create a test record in the record producer to verify that it triggers the flow to run. We also verify that approvals route to the correct users and that the flow performs the actions that we defined, such as updating records.

Step 1	✓	Planning your application
Step 2	✓	Create an app
Step 3	✓	Building a data model
Step 4	✓	Creating user experiences
Step 5	✓	Adding logic and automation
Step 6	○	Test your application

For more information about testing your application, see Testing and debugging applications.

Follow along with the tutorial to test the employee travel request application.

https://player.vimeo.com/video/953653889?badge=0&autopause=0&player_id=0&app_id=58479

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

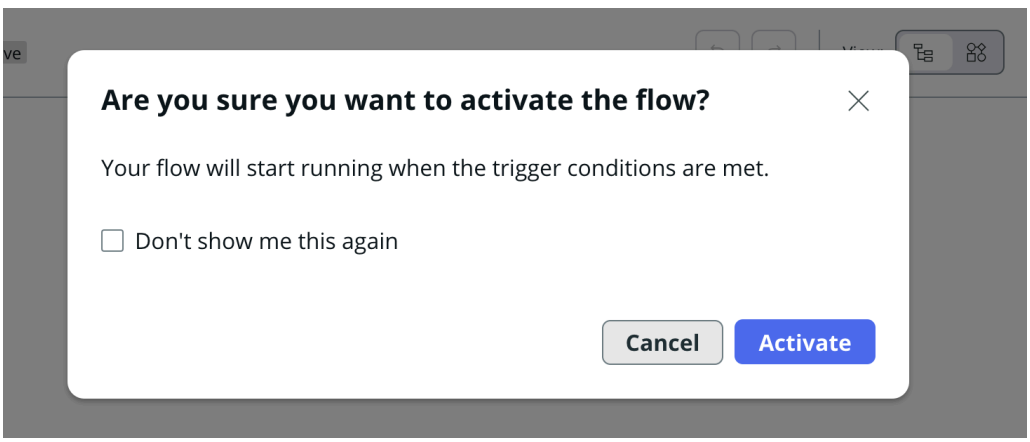
Video sections

Timestamp	Section
0:07	Activate your flow.
0:10	Navigate to your instance home page.
0:15	Impersonate a user to test your user experience.
0:20	Find and open your record producer.
0:26	Create a test record.
0:42	End the impersonation.
0:45	Navigate back to your flow.
0:48	Test your flow using the test record that you created.
0:56	Review the flow execution details.
1:00	Perform additional tests to verify that branching actions and flow logic run correctly.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. Open your application's flow.
   - If your application isn't already open, navigate to **All** > **App Engine** > *App Engine Studio*. Then open your application and select the flow under the **Logic and automation** heading.
   - If your flow is already open, verify that your changes are saved.
3. Select **Activate**, then select **Activate** again to make your flow available to run.

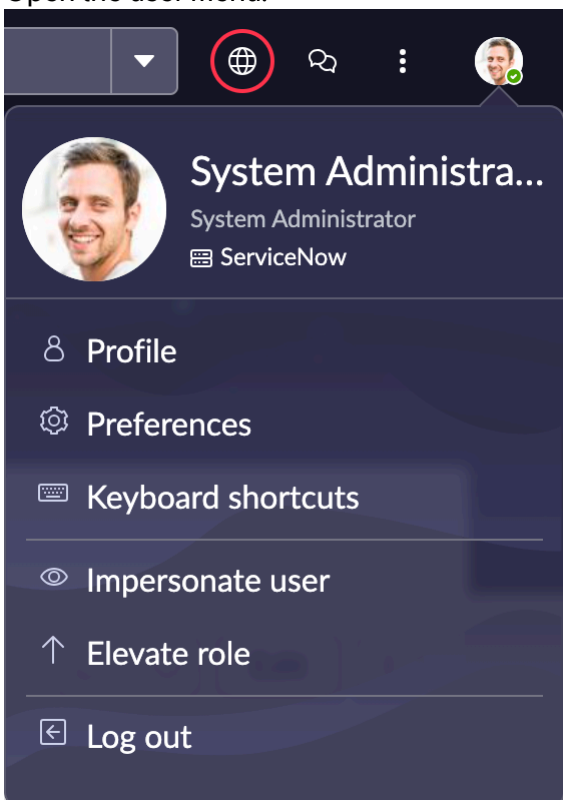
> **Note:** You can also test your flow without activating it by creating a test record within Workflow Studio. For more information about testing before activating your flow, see Test a flow.



4. Navigate to your ServiceNow AI Platform instance home page.
5. Impersonate a user to test that your record producer functions as intended.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. Open the user menu.



b. Select **Impersonate user**.

c. Select a user from the **Recent Impersonations** list or enter a different user's name in the user selection field.
When testing your application, you should impersonate users who have the permissions required to access the features that you're testing.
In our tutorial, we select a user with regular user permissions, as the record producer that we're testing is accessible to all users.

d. Select **Impersonate user**.
The system updates the user menu and dashboard to reflect the correct information for the user that you're impersonating.

6. On the home page, navigate to **All** > **Self Service** > **Service Catalog**.
7. Select **Services**.
8. Select your record producer from the list.
In our tutorial, we select our record producer called `Raise a travel request`.
9. Complete the record producer fields to generate a test record.
10. Select **Submit**.
11. End the impersonation.
    a. Open the user menu.
    b. Select **End impersonation**.
12. Reopen your application's flow.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. Navigate to **All** > **App Engine** > *App Engine Studio*.

b. Open your application.

c. Select your flow under the **Logic and automation** heading.

13. Select **Test**.

14. In the **Test Flow** dialog, select the record field, then select the record that you created during the impersonation.

15. Select **Run Test**.
The system runs the flow based on the test record that you selected.

16. Select **Your test has finished running. View the flow execution details**.

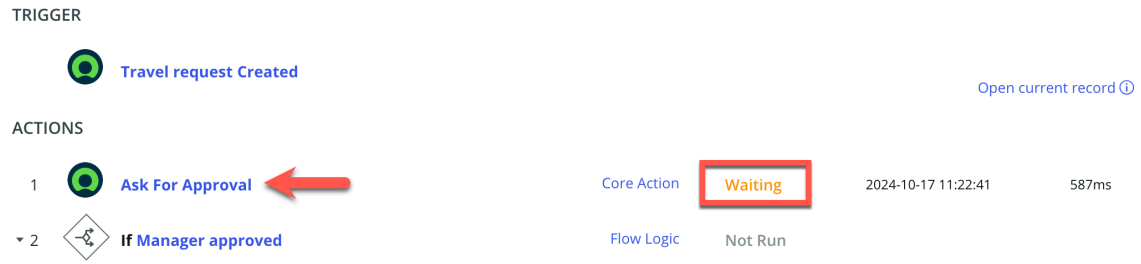
17. Review the flow execution details to verify that your flow runs as intended.
The flow execution details page enables you to view detailed information about an action or flow, such as the run-time, current state, items run, and values produced. For more information about flow execution details, see Flow execution details.

18. **Optional:** Perform additional tests to verify that the flow completes all desired branches of actions and flow logic.

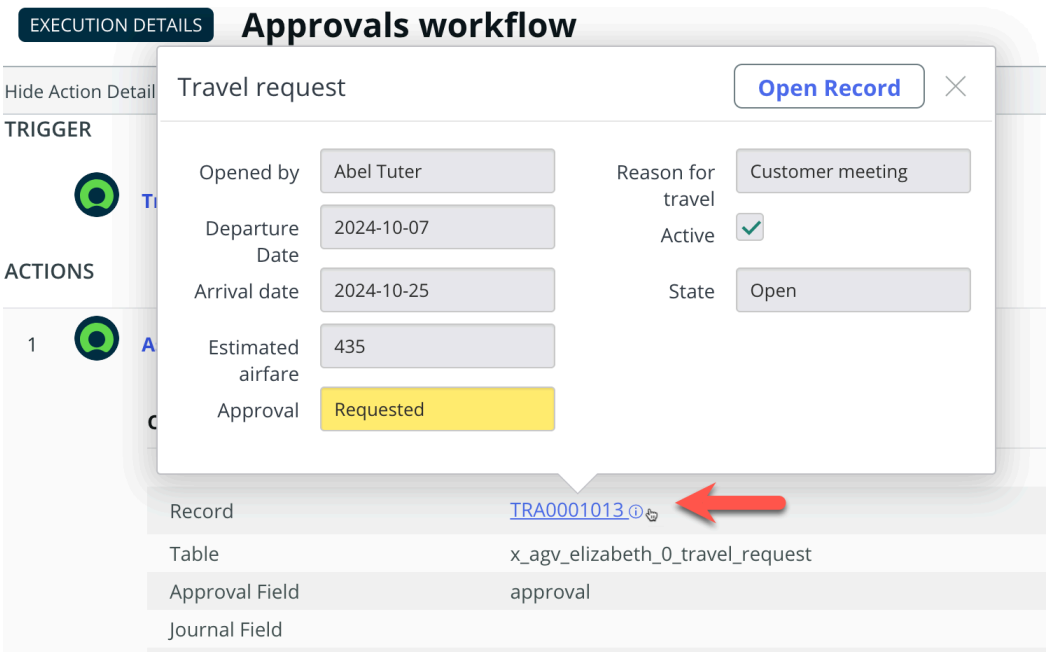
(Optional) For example, if your flow performs an action only after certain conditions are met, you can create those conditions to verify that the branching actions and logic run correctly.

In our tutorial, we simulate the manager's approval to test the branching actions and flow logic.

a. Select the action that is in the **Waiting** state to view the action details.



b. In the action details, select the test record.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

c. Select **Open Record**.

d. Open the approval record by selecting the list item in the **Approvers** list.

> **Tip:** If your test record page doesn't display the **Approvers** list, you can adjust the page configurations to display any lists that you want to appear. Complete the following steps to configure the lists that appear on your record page.
>
> i. On the record page, select the Additional actions icon ( ).
>
> ii. Select **Configure** > **Related lists**.
>
> iii. Select the **Approvers** list item or any other lists that you want to appear on the record page and move them from the **Available** list to the **Selected** list.
>
> iv. Select **Save**.

e. On the approval record page, select **Approve** to simulate the manager's approval.

f. Navigate back to your flow.

g. Select the refresh icon ( ) to update your flow.

h. Verify that your flow has run branching actions or flow logic based on the additional test that you performed.

19. **Optional:** Complete this procedure to perform as many additional tests as needed to verify that all branches of your flow run correctly.

Result

Your have tested the employee travel request application and verified that it functions as expected.

What to do next

Congratulations! You have completed the app creation tutorial by planning, building, and testing an application.You are ready to start building your own applications in *App Engine Studio*.

To see what tasks you could perform after building an application, such as publishing or managing app deployments, see Publish your app.

Plan your app development

The application development process in App Engine Studio (AES) begins with your idea for the next effective app. Careful planning is the key to creating a successful app that is beneficial to your business.

Fit for the ServiceNow AI Platform

Consider the characteristics of your application to help determine whether it is suited for the ServiceNow AI Platform.

Good fit	Poor fit
• Simple forms • Task management	• Unstructured data • Unrepeatable processes

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Good fit	Poor fit
• Request management • Spreadsheet-driven processes • Repeatable processes • Third-party integrations • Orchestration of multiple systems • Single experience from functions in multiple systems • Web and mobile access to the same apps and data simultaneously	• Content that needs graphics processing • Streaming audio or video • Highly customized user interface

Plan before you build

To build your application, you need access to a ServiceNow instance and an admin or delegated developer role in that instance. If you have delegated developer permissions, you can develop applications but have fewer privileges than an admin.

Essentially, an application is a digital program that supports user tasks. As you plan your app, consider the questions in the following table.

Questions	Answers and considerations
What are the goals, objectives, and outputs of your application? What business problem are you trying to solve?	Without a clear business objective, you might have difficulty measuring the success of your application or justifying its continued use within the organization.
Are you turning a spreadsheet into a ServiceNow application or does the application exist somewhere else?	The answer to this question helps determine the tools you would use within the ServiceNow AI Platform to support your efforts.
Who will use your application?	Identifying your target audience has a direct impact on the features you include in your application, the data you choose to capture, and the interface you provide for the application.
Do you want all users to have the same ability to see and edit parts of your application or should some people have more or less access than others?	Identifying who has access to what information during the planning stage is a critical step in application development.
What is the main purpose of the application? Is the application used to provide information, collect information, route information, look up information, request something, or collaborate on information?	Identifying the application's purpose helps to establish the features and functions you need to build into your application.
Will your application require entering information or does the data in your application exist in your ServiceNow instance (that is, user data)? Will data be imported from an external source?	Use available data sources as much as possible to avoid duplication of data and to ensure your application has the data it needs to meet business objectives.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Questions	Answers and considerations
How will users interact with your application? Will users access your application using their computer or mobile device?	Understanding how users access your application affects how you build your application's functionality.
How will you report on your application to stakeholders?	If your application is meeting a business purpose, you might need to provide reports showing usage, adoption, and key business objectives related to your application.
How will you maintain this application?	Planning for the maintenance of your application at the start can prevent roadblocks further into the development process.

The answers to these questions can help you decide whether you want to base your application on a predefined template or create your app from scratch.

Irreversible considerations

Some actions you take when building an application might be irreversible. Be aware of these actions and plan for them in advance.

Action	Considerations
Creating a scoped or global application	When you create an application, you can choose to create it in a private scope (scoped application) or in a global scope (global application). Scoped applications have extra functionality for managing development, application deployment, and data security. By default, all applications are created in a private scope. Citizen developers should generally work with scoped applications.
Using different instances	Proof of concept (PoC) applications can be built in a personal developer instance (PDI) obtained from the ServiceNow Developer site. • PDI names are similar to dev12345.service-now.com. • PoC applications can be rebuilt in your PDI. Don't import the PoC app into your PDI. Production applications that your organization uses should be created in a development instance so they can be subject to your organization's testing and deployment process. Ask your ServiceNow administrator which instance to use for developing applications that will eventually be deployed to your organization's production instance.
Selecting an application name	Based on your application's display name, auto-complete displays suggestions for an

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Action	Considerations
	internal name also known as the application scope. Application scopes are written in the form x_[company code]_[app_name], for example, x_acme_legal_request. Every property created in your application inherits the application scope name so be sure to choose it carefully before starting to develop your application. **Note:** You can always change the application's display name but the application scope name can't be changed.
Selecting table and field names	After your application is created, begin creating tables and fields. Tables and fields have internal database names that should be edited only at creation time.

Application Intake process

Any member of your organization can submit ideas for app development through a process called Application Intake. After the idea has been submitted and approved, developers can begin building the app with a guided, intuitive app development environment in App Engine Studio.

The Application Intake request process benefits your organization as follows:

- Simplifies the process to request, approve, track, and store app ideas.
- Improves planning by providing visibility into all app ideas.
- Provides an efficient process to filter out repetitive requests.

For more information, see Submit your idea for app development.

- The intake request form is available through the standard Service Catalog. For information about how you can modify this form, see Service catalog items.
- The intake request process uses Flows in Workflow Studio to determine the approval flow. To adjust the approval flow, see Flows.
- Admins review intake application requests in App Engine Management Center . For more information, see Managing app development using the App Engine Management Center.

Application templates

An application template provides predefined content to support a certain purpose. For example, the Travel Request template provides application content for submitting and approving employee travel requests. Choose the template that most closely fits your application goals.

If the available templates don't fit your application goals, you can create your app from scratch and control all aspects of the development process.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Additional resources for planning your app development

Learn more about planning app development	ServiceNow resources
ServiceNow provides additional resources on planning your app development	 Launching an effective Citizen Development program

Create your app

Use a template in App Engine Studio (AES) to build data, experience, logic and automation, and security into your application automatically. Or, create your app from scratch and customize it as needed.

Note: Before creating an app, Plan your app development.

To build an application in App Engine Studio, add the following content:

Data

Information that is stored in your application. For example, employee phone numbers or office locations. You configure application data using tables.

Experience

Graphical interface that your users interact with. For example, you can create a portal where users find information, submit requests, or complete business tasks.

Logic and automation

Automate all the work in your application by adding logic and automation. For example, you can build a flow that sends a notification to the admin when someone makes a request.

Security

Roles and access controls to limit who can use your application.

Note: To create apps in AES, you must be an admin or in the App Engine Studio Users group. If you are in the App Engine Studio User Limited group, you can only edit existing apps, not create new ones.

If you use an application template, the template automatically adds one or more of these items to your application. You can then edit them to tailor to your specific business needs. You can also add more data, experience, logic and automation, or security to an application that you created using a template.

To build an application from scratch, add data, experience, logic and automation, and security to your application and then edit each.

Create your app using an application template

Build an application in App Engine Studio (AES) that uses predefined data, experience, logic and automation, and security. Add to the template contents to customize the app for your organization.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

Note: To create apps in AES, you must be an admin or in the App Engine Studio Users group. If you are in the App Engine Studio User Limited group, you can only edit existing apps, not create new ones.

Role required: admin, sn_app_eng_studio.user

About this task

Using templates in App Engine Studio helps you save time and effort creating applications. You can download several templates from the ServiceNow Store, or you can create templates from your existing applications. To install a template on your PDI, see Activating a plugin from your PDI.

This video shows you how to perform the following procedure.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the **Templates** tab, browse the available templates.
3. Hover over the template you want to use and select **Use template**.
4. Enter a **Name** and **Description** for your application.
5. **Optional:** Add a logo by dragging an image into the AES page or browsing your computer for the file.
6. Select **Continue**.
7. On the summary screen, select **Go to app home**.
8. From the app home, review the data, experience, logic and automation, and security that was created with the template.
   To tailor the application to your business needs, you can edit these predefined items or add your own. For more information, see Enhance your app.

What to do next

After you've finished building your application, submit the application for approval to get it reviewed and deployed by an administrator. For more information, see Submit your app for approval and publishing.

Available templates

Use an application template to create an application in App Engine Studio (AES) with preconfigured data, experience, logic and automation, and security.

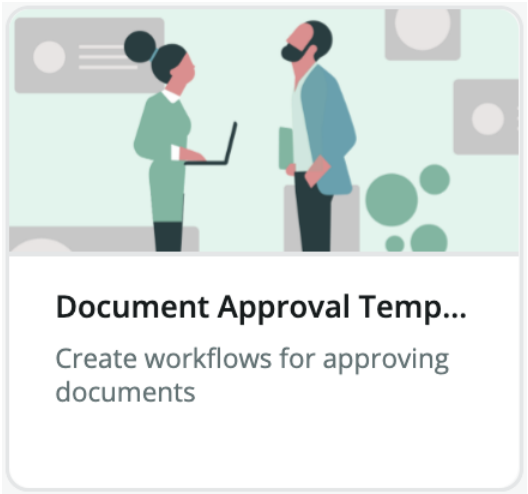
Templates are a good starting place for many applications you might want to build. You can also begin your application using a template and then customize the application. Use any of the templates below to build a fully-fledged application, or use them as a starting point to create an app customized to your needs.

Document Approval template

Build an application in App Engine Studio (AES) using the Document Approval template to manage the approval of documents in your organization.

To use the Document Approval template, install this application from the ServiceNow Store.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

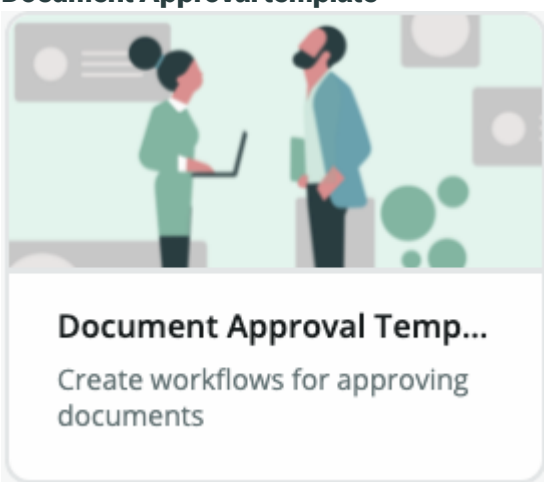


A document approval app enables your users to quickly create and use document approval workflows. Document owners and submitters can upload documents needing approvals and check the progress on those approvals. Designated approvers can approve, reject, or return documents with comments.

Admins can use the document approval application to create different workflows for different document types.

This app template is an ideal way to quickly and efficiently make your internal workflows for processing documents. Document owners and submitters can upload a document needing approval to a portal using simple upload functionality that automatically sends submissions through a unique workflow. Once a document has been received, submitters follow the progress of their documents while designated users approve, reject, or return their document for edits. Both submitters and approvers use comments to add notes and feedback to the submission, which is stored in the system for future reference. To give this app even more variety, admins can create unlimited workflows that are customized for each document type and user that allows for multiple stages and approvers depending on what the document category requires.

Document Approval template



For more information on creating an application using a template, see Create your app using an application template.

Document Approval template contents

The following tables, experiences, automated workflows, and roles are included in the Document Approval template for App Engine Studio (AES).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Tables in the Document Approval app template

Table label [name]	Description
Document Approver Table [x_ <company-code>_ <document_approver>]	Tracks document approvers, their comments, and state of a document.
Document Approval Category Table [x_ <company-code>_ <document_approval_category>]	Tracks the category of document approval requests.
Document Approval Table [x_ <company-code>_ <document_approval>]	Tracks the document approval requests.

Experiences in the Document Approval app template

Experience	Experience type	Description
Document Approval | Default view	Form	Form for creating a document approval request.
Document Approval | Portal	Form	Simplified form for creating a document approval request.
Document Approval Category | Default view	Form	Form to attach a flow to a document approval record.
Document Approval Category | Workspace	Form	Form to attach a flow to a document approval record.
App name Workspace	Workspace	Main document approval app workspace, where users can upload documents, check the status of an approval, edit documents, and view their submissions in the workspace. Approvers can view assigned approvals, view previous submissions, and approve or reject submissions.
Document Approver | Workspace	Form	Form for assigning document approvers to records.
Document Approval | Workspace	Form	Form for creating a document approval request.
Document Approver | Default view	Form	Approvers can view assigned approvals and approve or reject submissions.
App name Portal	Portal	Main portal for people to submit documents for approvals and manage previous submissions.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Experience	Experience type	Description
Document Approval Mobile Experience	Mobile experience	Mobile app that employees and managers can use to check the status of an approval, approve or reject documents, and view submissions using your organization's iOS or Android app.

Automated workflows in the Document Approval app template

Workflow name	Workflow type	Description
Document Approval - Master Flow	Flow - record	Executed when a new document approval record is submitted. It then retrieves the flow to be executed from the document category and executes it.
Document Approval Notification	Email	Notifies users of changes to their document approvals.
Single Stage Approval	Flow	Allows for the user's manager to be the approver, with basic email notifications being generated on approval, return, or rejection.

Roles in the Document Approval app template

Role title [name]	Description	Contains roles
Approvers [x_<company-name>_<document_approval_app>.approver]	Approvers can create, read, delete all documents, and read document approval category data.	Create, Read (all), Edit (all)
Admins [x_<company-name>_<document_approval_app>.admin]	Admins can create, read, write, and delete in the Document Approval [x_ <company-code>_<document_approval>] and Document Approval Category [x_ <company-code>_<document_approval_category>] tables.	Create, Read (all), Edit (all), Delete (all)
Submitters [x_<company-name>_<document_approval_app>.submitter]	Submitters can create documents, read their own documents, edit their own documents when status is not submitted and delete their own documents. They can	Create, Read (own), Edit (own when status is not submitted), Delete (own)

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Role title [name]	Description	Contains roles
	also read data from document approval category.	

Emergency Alert template

Build an application in App Engine Studio (AES) using the Emergency Alert template to manage communications and track essential resources during an emergency.

An emergency alert app allows your organization to keep employees informed and connected during natural disasters and infectious diseases, such as COVID-19.

You can use the Emergency Alert template to create command center dashboards and a mobile app to monitor and notify employees about the latest news. Employees can self-report their status, request work-from-home or time off, and receive instructions during an emergency. Managers can assess the impact on their respective teams, approve or reject requests, request for status, and automatically receive instruction on actions to take during any emergency.

For more information on creating an application using a template, see Create your app using an application template.

Emergency Alert template contents

The following tables, experiences, automated workflows, and roles are included in the Emergency Alert template for App Engine Studio (AES).

Tables in the Emergency Alert app template

Table label [name]	Description
Status Report Request Table [x_<company>_<my emergency_alert_app>_status_report_requests]	Track the severity of an emergency reported by employees. For example, the status, the color associated with danger levels, and conditions.
Self Report Table [x_<company>_<my emergency_alert_app>_status_report]	Track reported emergencies created by employees or managers. For example, the emergency details, status, duration, and affected individuals.
Emergency Alert Table [x_<company>_<my emergency_alert_app>_emergency_alert]	Track emergency alerts for your company. For example, the emergency type, who created the alert, and who has read the alerts.
Work Status Request Task Table [x_<company>_<my emergency_alert_app>_work_status_request_task]	Track the work status request of an emergency. For example, the requester, the assignee, and the state of the work status.

Experiences in the Emergency Alert app template

Experience	Experience type	Description
Emergency alert | SendEmergency AlertWS	Form	Form to create a new emergency alert record.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Experience	Experience type	Description
Self Report | Employee View	Form	Form for employees to report an emergency.
Self Report | Default View	Form	Form for employees to report an emergency.
Emergency Alert - Portal	Portal	Portal where employees can view reported emergencies, the status of their team, and submit a self report.
Emergency Alert | Default View	Form	Form to create a new emergency alert and connect knowledge resources to it.
Self Report | Update Report	Form	Form for employees to self report their health status.
Self Report | Activity View	Form	Form to add comments about a self report.
Self Report | RequestReportWS	Form	Form to self report an emergency within a specific time frame.
Status Report Request | Default View	Form	Form to send an update about an emergency with time and location details.
Self Report | Update status report	Form	Form to add an update about an emergency.
App name Workspace	Workspace	Workspace where admins can set up and send notifications, view dashboards, and configure lists and forms.
Work Status Request Task | Default view	Form	Form to update an employee about a work status request.

Automated workflows in the Emergency Alert app template

Workflow name	Workflow type	Description
Emergency report/update state for managers	Email	Notifies managers to update an emergency report if the source is empty.
Emergency Alert App - Please report or update status	Flow - record	Requests a status update on work.
Emergency Alert App - Status request for employees from manager	Flow - record	Notifies employees that a status request is made by their manager.
Self report submitted without manager	Email	Notifies that an employee submitted an expense request without listing their manager.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Workflow name	Workflow type	Description
Emergency Alert App - Create Work Status Task When Self Report is created	Flow - record	Starts the work status task when a self report emergency is submitted.
Emergency Alert App - Create Status Report Requests for Affected	Flow	Creates or updates status report requests.
Emergency alert app - Create App Self Reports	Flow - record	Sends a notification when a report request is created or updated.
Emergency Alert App - Request for WFH/Time off submitted	Flow - record	Starts the work from home or time off request approval flow.
Emergency instructions for employees	Email	Notifies users if a status update is requested.
Emergency Alert App - Employee emergency report changes to submitted	Flow - record	Sends a notification when an employee self report has been submitted.
Emergency Alert App - Request for WFH/Time off approved/denied	Flow - record	Sends a notification to an employee when a WFH or Time off request is approved or denied.
Emergency Alert App - Employee emergency report request response submitted	Flow - record	Sends a notification when an employee emergency report has been submitted.
Request for WFH approved	Email	Notifies an employee that their request to work from home was approved.
Emergency alert app - Emergency instructions for employees	Flow - record	Sends employees instructions to follow during an emergency.
Please report/update status of employee	Email	Requests information about the status of an employee.
Self-report submitted for manager	Email	Notifies a manager that an emergency alert has been created and needs review.
Emergency Instructions for managers	Email	Notifies managers that their request was created and that their team will receive information about it.
Please report/update status	Email	Sends a notification when a status change has been committed.
Request for WFH/Time off submitted	Email	Notifies an employee that their work from home or time off request was successfully submitted.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Workflow name	Workflow type	Description
Employee Emergency report submitted	Email	Notifies an employee that their emergency report was successfully submitted.
Emergency Alert App - Manager emergency request submitted	Flow - record	Sends a notification when a manager emergency report has been submitted.
Emergency Alert App - Employee emergency report submitted	Flow - record	Sends a notification when an employee submits changes to an emergency report.
Create KB article	Flow - record	Creates a knowledge base article when a new emergency alert is created.
Employee emergency report request submit	Email	Notifies an employee that their request for status was submitted and that their manager will be notified.
Emergency Alert App - Cancel work status task if self report closed	Flow - record	Cancels a work status request if the self report is closed.
Manager emergency request submitted	Email	Notifies managers that the emergency report was sent.
Please report/update status of employee	Email	Notifies employees of action needed on their self report.
Request for WFH denied	Email	Notifies the employee that their work from home request was denied.
Emergency Alert App - Emergency report/update state for managers	Flow - record	Sends a notification based on changes in an emergency alert report.

Roles in the Emergency Alert app template

Role title [name]	Description	Contains roles
Emergency alert admin [x_<company-code>_ <my emergency_alert_app>.emergency_alert_admin]	Admins manage the app and configure the dashboard	Create, Read (all), Edit (all), Delete
Emergency alert manager [x_<company-code>_ <my emergency_alert_app>.manager]	Managers can approve or reject requests, access dashboards, and send notifications	Create, Read (all), Edit (all)
Emergency alert employee [x_<company-name>_ <my emergency_alert_app>.emergency_alert_employee]	Self-report emergency alerts	Create, Read (own), Edit (own)

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Role title [name]	Description	Contains roles
Admin	The System Administrator role. This role has access to all system features, functions, and data regardless of security constraints.	Create, Read (all), Edit (all), Delete
Public	No login is required to access features or functions with the public role.	No access by default.

Event Registration template

Build an application in App Engine Studio (AES) using the Event Registration template to manage scheduling, booking, and registration for company events.

To use the Event Registration template, install this application from the ServiceNow Store.

An internal event registration app allows your event organizers to publish a list of events to employees and collect attendees in a centralized database. The app helps streamline the event discovery and registration process.

Event organizers can manage events and wait-lists, publish new events, and view registered users' details. Employees and attendees can view a list of events, view event info, and register to attend in a portal. Organizers can set the size of the event, and requesters will automatically be added to a wait-list.

For more information on creating an application using a template, see Use an application template.

Event Registration template contents

The following tables, experiences, automated workflows, and roles are included in the Event Registration template for App Engine Studio (AES).

Tables in the Event Registration app template

Table label [name]	Description
Attendee Table [x_<company-code>_<attendee>]	Track event attendee information such as name, status, and date of submission.
Event Table [x_<company-code>_<event>]	Track event information such as event name, date of the event, and capacity.
Event Location Table [x_<company-code>_<event_location>]	Track event location details such as capacity and location.

Experiences in the Event Registration app template

Experience	Experience type	Description
Event Location | Workspace	Form	Workspace where organizers and admins can configure

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Experience	Experience type	Description
		events, create event applications, and view lists.
App name Portal	Portal	Main portal for organizers to create and view details about the event.
Attendee | Default view	Form	Form that attendees can use to view the event name, change their status, and take notes about the event.
Event | Default view	Form	Form that organizers can use to create events.
App name Workspace	Workspace	Workspace where users can register for an event and where admins/organizers can manage registration information.
Event Location | Default view	Form	Form for organizers to add event location name, location address, status, and capacity.
Event | Workspace	Form	Form for organizers to add event name, location details, timelines, and comments.
Attendee | Workspace	Form	Form for attendees to register for events and add notes about the event.
Mobile app for organizers	Mobile experience	Organizers can view or edit events and view attendees using your organization's iOS or Android app.

Automated workflows in the Event Registration app template

Workflow name	Workflow type	Description
Event Details Have Changed	Email	Notifies users/attendees of changes to the event.
Event Reminder - Confirmed attendees	Email	Sends a reminder about the event to confirmed attendees.
Attendee registered confirmed	Email	Notifies attendees of confirmed registration.
Send Event Reminder Email	Flow	Sends a reminder email to event attendees.
Send Event Details Changed Email	Flow - record	Sends an email to all confirmed and waitlisted attendees when details of an event have changed or the event was cancelled.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Workflow name	Workflow type	Description
Send Waitlist Confirmation Email	Flow - record	Sends an email to attendees when they are added to the waitlist of an event.
Event Registration cancelled events	Flow - record	Updates attendee records when an event is cancelled.
Update Event Registration attendee records	Flow	Updates attendee records when an event is cancelled or is past the end date.
Attendee waitlisted confirmed	Email	Confirms attendees on event waitlist.
Event Registration expired events	Flow	Sets the **Event state** field to Expired and the **Attendees** field to Event ended after the event ends.
Send On Hold Confirmation	Flow - record	Allows for organizers to send an email to attendees when they are on-hold to attend an event.
Event Registration Confirmation Email	Flow - record	Sends an email to attendees when they are confirmed to attend an event.
Attendee registration on hold	Email	Informs attendees that their registration is on hold.
Event Has Been Cancelled	Email	Informs users of event cancellation.
Event Registration Process Flow	Flow	Processes all pending requests.

Roles in the Event Registration app template

Role title [name]	Description	Contains roles
Organizer [x_<company-name>_<my events app>.organizer]	Organizers create and manage events.	Create, Read (all), Update (all).
Attendee [x_<company-name>_<my events app>.attendee]	Attendees can search, sign up, and RSVP to events.	Create, Read (all), Update (own).
Admin [x_<company-name>_<my events app>.admin]	Admins configure event locations and applications.	Create, Read (all), Update (all), Delete.
Public	No login is required to access features or functions with the public role.	No access by default.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Role title [name]	Description	Contains roles
[x_<company-name>_<my events app>.public]		

Expense Pre-Approval template

Build an application in App Engine Studio (AES) using the Expense Pre-Approval template to create, track, and manage planned expenses in your organization.

To use the Expense Pre-Approval template, install this application from the ServiceNow Store.

Employees can easily submit their planned expenses for multiple levels of approval using the Expense Pre-Approval application. Managers can then review every submitted request. Approved requests are assigned to the budget review group, which can approve or reject the request. The employee receives a notification with the final status of the request.

To ensure compliance and improved financial planning, admins can customize the existing flow and levels of approval using decision tables. For more information, see Exploring Decision Tables.

For more information on creating an application using a template, see Create your app using an application template.

Expense Pre-Approval template contents

The following tables, experiences, automated workflows, and roles are included in the Expense Pre-Approval template for App Engine Studio (AES).

Tables in the Expense Pre-Approval app template

Table label [name]	Description
Budget definition [x_<company-code>_<expense_approval_app>_budget_definition]	The description of all budgets.
Expense request [x_<company-code>_<expense_approval_app>expense_request]	The expense requests submitted by employees.
Expense line item [x_<company-code>_<expense_approval_app>_expense_line_item]	Each expense line item in an expense request.

Experiences in the Expense Pre-Approval app template

Experience	Experience type	Description
Expense request | Default view	Form	Form to create a new expense request.
Budget definition | Default view	Form	Form to define budgets for a specified time frame.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Experience	Experience type	Description
Budget Reviewer	Mobile experience	Way to define what the mobile experience for budget reviewers looks like.
Expense request | Workspace	Form	Form to create a new expense request.
Expense request | Cost Approval	Form	Form to create a new expense request from an admin.
Manager	Mobile experience	Way to define what the mobile experience for managers looks like.
Budget definition | Workspace	Form	Form in the workspace view to define budgets for a specified time frame.
App name - Budget workspace	Workspace	Way to configure a high-level view of all budget requests.
Expense request	Record producer	Way for employees to submit an expense request.
Expense line item | Default view	Form	Form to add an expense line item to an expense request.
Expense Request SLA | Default view	Form	Form to create an expense request based on a breach of service level agreement (SLA).
Budget definition | Cost Approval	Form	Form in the cost approval view to define budgets for a specified time frame.
App name - Manager workspace	Workspace	Workspace where managers can approve or reject the expense requests assigned to them.
Cost Approval Mobile App	Mobile experience	iOS or Android app used for the following actions: • Approving or rejecting expense requests. • Viewing dashboards that display historical data to compare budget and expense requests. • Creating or modifying budgets by budget reviewers.
Cost Approval Workspace - Budget Reviewer	Workspace	Budget reviewers can use the workspace for the following actions:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Experience	Experience type	Description
		• Approving or rejecting expense requests. • Creating or modifying budgets. • Viewing the data on the budget funds allocated, spent, available, and allocated for the future. • Viewing dashboards that display historical data to compare the budget and expense requests.

Automated workflows in the Expense Pre-Approval app template

Workflow name	Workflow type	Description
Expense request approved - reviewer	Email	Notifies users that an expense request has been approved.
Expense Request Assigned	Email	Notifies users that an expense request has been assigned.
Expense request - SLA breached	Email	Notifies users that a service level agreement (SLA) has been breached.
Budget Creation	Flow - record	Executed when a budget reviewer creates a budget. It checks the start date of an expense request and associates the request to a budget with corresponding time frame.
Expense request - SLA- close to due	Email	Notifies users that a service level agreement (SLA) is close to due.
Expense Request Reassigned	Email	Notifies users that their request was reassigned.
Expense request comment added - Assigned	Email	Notifies users about a comment that has been added to their request.
New Expense Request or manager approval	Email	Notifies users about a new expense request.
Expense request reject	Email	Notifies users that their request was rejected.
Budget state update	Flow	Updates each budget record by priority.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Workflow name	Workflow type	Description
Expense Request Inserted - Requestor	Email	Notifies users that their request was successfully created.
Expense Request Approved	Email	Notifies users that their request was approved.
Expense request work note added	Email	Notifies users a work note was added to their request.
Notify Team - Work note added	Email	Notifies a team that a work note was added to a request.
Expense request comment added - Employee	Email	Notifies users a comment was added to a request.
Expense Request Rejected	Email	Notifies users that their request was rejected.
Budget Updation	Flow - record	Executed when a budget reviewer modifies the budget. After a budget is modified, this flow updates the **Budget** field in the corresponding expense requests.
Notify Requestor - Comments added	Email	Notifies users that a comment was added to a request.
SLA Notification - Expense Request - warning	Flow - record	Warns admins about a potential service level agreement (SLA) issue.
Budget Approval Process	Flow - record	Executed to obtain expense approvals in the following scenarios: • When an employee submits an expense request for the manager's approval. • Upon the manager's approval, the expense request is sent to the budget reviewer for the final level of approval.
Budget Calculations	Flow	Recalculates the budget spent and the available budget when a budget reviewer performs any of the following actions: • Creates a budget • Modifies an existing budget • Approves an expense request

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Roles in the Expense Pre-Approval app template

Role label [name]	Description
Budget Reviewer [x_<company-code>_<expense_approval_app>.Budget reviewer]	Budget reviewers can create or modify budgets and provide the final level of approval on expense requests.
Manager [x_<company-code>_<expense_approval_app>.Manager]	Managers can submit, view, and approve or reject expense requests at the first level.
Employee [x_<company-code>_<expense_approval_app>.Employee]	Employees can submit and track expense requests.
Admin	The System Administrator role. This role has access to all system features, functions, and data, regardless of security constraints.

Inventory Tracker template

Build an application in App Engine Studio (AES) using the Inventory Tracker template to track the status of inventory and assets.

To use the Inventory Tracker template, install this application from the ServiceNow Store.

An inventory tracker app enables your business and teams to easily track what they have, where it is, and who has it. Anytime, anywhere, and on any device. If your company needs to track the stock of items, this app can be used as a stock management tool or inventory tracker at home or in a warehouse.

This app covers three inventory use cases: consumables, loaners, and discontinued. Employees can request items via mobile app or web portal from any warehouse. Inventory managers can set reorder levels and will be automatically notified if an item is low on stock. Discontinued items with outstanding stock can still be requested but will not prompt inventory managers for reorder. Inventory managers can create items manually or by uploading a spreadsheet.

For more information on creating an application using a template, see Create your app using an application template.

Inventory Tracker template contents

The following tables, experiences, automated workflows, and roles are included in the Inventory Tracker template for App Engine Studio (AES).

Tables in the Inventory Tracker app template

Table label [name]	Description
Inventory Control Table [inventory_ x_<company-code>_<my_inventory_tracker_app>_inventory_control]	Track the stock of items used by your business. For example, item name, manufacturer, image, category, and location.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Table label [name]	Description
Inventory Request Table [x_<company-code>_<my_inventory_tracker_app>_inventory_request]	Track inventory requests submitted by employees. For example, the name and department of the requester, stock location, count, requested date, and return date.
Stock Location Table [x_<company-code>_<my_inventory_tracker_app>_stock_location]	Track the location of stocked items. For example, name, address, and inventory manager.

Experiences in the Inventory Tracker app template

Experience	Experience type	Description
Inventory Control | Portal	Form	Form for admins and managers to add inventory items, view stock quantity, and add stock locations.
Inventory request | Workspace	Form	Form for employees to submit and manage inventory requests.
Inventory Control | Default view	Form	Form for admins and managers to manage inventory items, view stock quantity, and manage stock locations.
App name | Portal	Portal	Portal where all users can manage inventory requests and tracking.
Inventory request | Default view	Form	Form for admins and managers to inventory items, view stock quantity, stock locations, and add notes.
Inventory request | Portal	Form	Form for admins and managers to request inventory items.
App name | Workspace	Workspace	Workspace to manage all Inventory tracker app configuration.
Stock Location | Workspace	Form	Form for inventory managers to add stock names, addresses, and descriptions.
Stock Location | Default view	Form	Form for inventory managers to add stock name, address, description, and department.
Inventory Control | Workspace	Form	Form for admins and managers to manage inventory items, view stock quantity, and stock locations.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Experience	Experience type	Description
Inventory Tracker App	Form	Form for employees to view inventory list, see stock quantities, request items, and check the status of their requests using their organization's iOS or Android app.

Automated workflows in the Inventory Tracker app template

Workflow name	Workflow type	Description
Inventory Tracker Request Completed - Admin/Manager	Flow - record	Sends a notification when an inventory request has been completed.
Request Update as Submitted - Manager	Email	Notifies a manager that they have a new update on a requested item.
Item Out of Stock- Admin	Email	Informs an admin of items that are out of stock.
Return Date Updated - Manager	Email	Notifies manager of an updated return date.
Inventory Tracker Request Submitted - Manager	Flow - record	Notifies managers when an inventory request is submitted.
Inventory Tracker Inventory track Created - Admin	Flow - record	Notifies admins when an inventory track is created.
Stock Location Updated - Admin	Email	Informs an admin of updated stock location.
Inventory Tracker Checked Out - Employee	Flow - record	Notifies an employee when an item is checked out.
Request Overdue - Manager	Email	Informs a manager of an overdue request.
Item Returned - Manager	Email	Informs a manager of returned items.
Request Completed - Manager	Email	Informs a manager of a completed request.
Inventory Tracker Request Approved - Employee/Admin/ Manager	Flow - record	Notifies employees, admins, and managers when an inventory request is approved.
Inventory Tracker Request Submitted - Employee	Flow - record	Notifies employees when their inventory request is submitted.
Checked Out - Employee	Email	Notifies employees of items that are checked out.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Workflow name	Workflow type	Description
Inventory Tracker Out of Stock - employee	Flow - record	Notifies employees if an inventory item is out of stock.
Return Date Updated - Admin	Email	Informs an admin of an updated return date.
Request Canceled - Manager	Email	Notifies a manager when a request is cancelled.
Inventory Tracker Stock Location Updated - Admin	Flow - record	Notifies admins when a stock location is updated.
Item Out of Stock - Manager	Email	Notifies a manager when an item is out of stock.
Inventory Tracker Item Returned - Employee/ Manager	Flow - record	Notifies an employee or manager when an item is returned.
Inventory Tracker Inventory track updated- Admin	Flow - record	Notifies an admin when an inventory track is updated.
Item returned - Employee	Email	Notifies employees of returned items.
Request Denied - Admin	Email	Informs admins of denied requests.
Inventory Tracker Stock Location Created - Admin	Flow - record	Notifies an admin when a stock location is created.
Inventory Item Discontinued - Admin	Email	Notifies admins of discontinued items.
Request Approved - Manager	Email	Notifies managers of an approved request.
Inventory Item Updated - Admin	Email	Notifies admin of updated inventory item.
Inventory Tracker Inventory track Created - Admin	Flow - record	Notifies an admin when an inventory track is created.
Out of stock - Employee	Email	Notifies employees of items that are out of stock.
Inventory Tracker Request Denied - Admin	Flow - record	Notifies admins when an inventory request is denied.
Request Approved - Employee	Email	Notifies employees of approved requests.
Inventory Tracker Item Out of stock - Admin/Manager	Flow - record	Notifies admins and managers when an inventory item is out of stock.
Request Approved - Admin	Email	Notifies admins of an approved request.
Request submitted - Manager	Email	Notifies admins of a submitted request.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Workflow name	Workflow type	Description
Request Overdue - Admin	Email	Notifies admins of an overdue request.
Request Denied - Manager	Email	Notifies managers of a denied request.
Stock Location Created - Admin	Email	Notifies admin when a stock location is created.
Overdue - Employee	Email	Notifies employee of overdue items.
Inventory Tracker Request Updated - Manager	Flow - record	Notifies managers when an inventory request is updated.
Request Denied - Employee	Email	Notifies employee of a denied request.
Inventory Tracker Inventory Discontinued - Admin	Flow - record	Sends a notification when an inventory item is discontinued.
Inventory tracker Return Date Changes - Admin/ Manager	Flow - record	Sends a notification to admin and manager when a return date changes.
Comments Added - Manager	Email	Notifies a manager about added comments.
Comments Added - Admin	Email	Notifies admins about added comments.
Inventory tracker Request Denied - Employee	Flow - record	Sends a notification to employees when their inventory request is denied.
Request Submitted - Employee	Email	Notifies employees when they submit a request.
Request created - Admin	Email	Notifies admins when a request is created.
Inventory Tracker Comments Added - Admin/Manager	Flow - record	Sends a notification to admins and managers when comments are added to inventory requests.
Request Canceled - Admin	Email	Notifies admins of a canceled request.
Inventory Tracker Request Overdue - Employee/ Manager/Admin	Flow	Sends a notification to employees, managers, and admins if a request is overdue.
Inventory Item created - admin	Email	Notifies an admin when an inventory item is created.
Request Completed - Admin	Email	Notifies an admin when a request is completed.
Inventory tracker Request Cancelled - Admin/Manager	Flow - record	Sends a notification to admins and managers when

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Workflow name	Workflow type	Description
		an employee cancels an inventory request.
Inventory tracker Request Created - Admin	Flow - record	Sends a notification to admins when an inventory request is created.

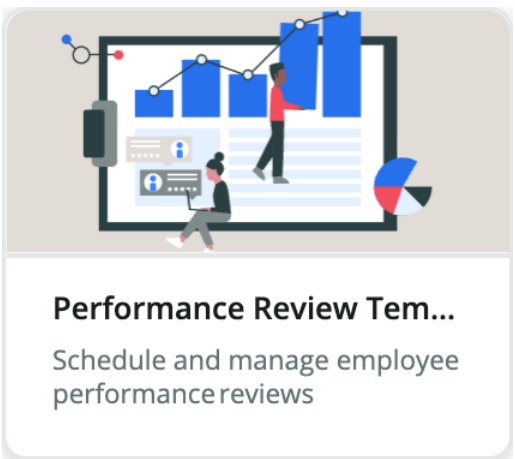
Roles in the Inventory Tracker app template

Role title [name]	Description	Contains roles
Employee [x_<company-name>_<my_inventory_tracker_app>.employee]	Employees can view inventory lists, check stock, request items, and check the status of their requests.	Create, Read (own), Update (own)
Admin [x_<company-name>_<my_inventory_tracker_app>.admin]	Admins can set up multiple inventory managers and stock locations.	Create, Read (all), Update (all), Delete
Inventory Manager [x_<company-name>_<my_inventory_tracker_app>.inventory_manager]	Inventory managers can create inventory items, define the type of item, monitor inventory levels, and update vendor information.	Create, Read (all), Update (all)
Public [x_<company-name>_<my_inventory_tracker_app>.public]	Public users require no login to access features or functions with the public role.	Contains no roles.

Performance Review template

Build an application in App Engine Studio (AES) using the Performance Review template to automate scheduling, notifications, and feedback for performance reviews.

To use the Performance Review template, install this application from the ServiceNow Store.



A performance review app automates the process of scheduling annual or quarterly reviews, reviewing employee performance, and providing manager feedback. The app displays report

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

information by default and lets managers view, update, and request employee performance reviews.

Employees can complete their self-evaluation, monitor the status, and confirm evaluations via a web portal. The app will automatically notify employees and managers to submit their evaluations.

Managers can request an update to the self-evaluations, submit manager-evaluations, confirm evaluations, and monitor the status of their team's submissions via a portal.

The HR/admin role can select an annual, bi-annual, or quarterly review process schedule and edit both the self-evaluation and manager-evaluation templates. HR/admins can monitor the status of all evaluations by department, organization, or manager.

You can improve this template by incorporating the following features:

- Add continuous option as a review process
- Use Playbook to manage review schedules
- Include HR review/escalation process for both self-reviews and manager reviews
- Add peer reviews process
- Add goal setting or objectives and key results (OKRs)
- Add integration with HR software
- Add skill set assessment metrics
- Add performance improvement plan (PIP) process/individual development plans
- Include compensation info/annual performance review (APR)

For more information on creating an application using a template, see Create your app using an application template.

Performance Review template contents

The following tables, experiences, automated workflows, and roles are included in the Performance Review template for App Engine Studio (AES).

Tables in the Performance Review app template

Table label [name]	Description
Performance Review Table [x_<company-code>_<my performance_review_app>_performance_review]	Track the performance evaluations created by employees and managers. For example, the employee name, state of the review, manager feedback, and state of the review.

Experiences in the Performance Review app template

Experience	Experience type	Description
Performance Review | Edit Settings	Form	Form where managers can configure performance review settings and add notifications.
App name | Workspace	Workspace	Workspace where admins and managers can configure review schedules, update

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Experience	Experience type	Description
		evaluation templates, and monitor the status of all evaluations.
Performance Review Test | Emp Mgr feedback	Form	Form for managers to create employee evaluation questions, monitor the status, and leave feedback.
Performance Review | Manager Review	Form	Form managers can use to include questions, review employees responses, and leave feedback.
Performance Review | Self Review WS	Form	Form for employees to complete self review questions.
Performance Review | Workspace	Form	Form for mangers to monitor the status of their team's submissions.
App name | Portal	Portal	Portal where employees can submit, update, review, and confirm evaluations.
Performance Review | Default View	Form	Form that managers and employees can use to view performance reviews.
Performance Review | Self Review	Form	Form employees and managers can use to submit a self-assessment of their performance.
Mobile app	Mobile experience	Employees can access their performance reviews using your organization's iOS or Android app.

Automated workflows in the Performance Review app template

Workflow name	Workflow type	Description
Perf Review - Send PR Employee Update Self-Eval email	Flow - record	Triggers the PR Employee Update Self-Eval email.
PR Employee Manager - Eval submitted	Email	Notifies managers that an evaluation has been submitted.
Perf Review - Send PR Employee Complete Evaluation email	Flow - record	Triggers the PR Employee Complete Evaluation email.
PR Employee Manager - Eval submitted DJ	Email	Notifies managers when self-evaluations are submitted.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Workflow name	Workflow type	Description
PR Admin Eval Deadline Passed	Email	Notifies admins when an evaluation deadline has passed.
PR Reminder: Manager need to submit eval	Email	Reminds managers to submit their evaluations.
PR Manager Self Evaluation Submitted	Email	Notifies admins that managers have submitted self-evaluations.
Perf Review - Send PR Employee Manager - Eval submitted	Flow - record	Triggers the PR Employee Manager - Eval submitted email.
PR Reminder - Employee Self Evaluation	Email	Notifies employees to submit self-evaluation.
Perf Review - Send deadline email	Flow	Triggers a deadline reminder email on the date indicated in the system properties.
PR Employee Complete Evaluation	Email	Notifies employees their evaluation is complete.
PR Employee Update Self Evaluation	Email	Notifies employees they need to update their self-evaluation.
Perf Review - Main Flow	Flow	Controls when reviews are triggered, email reminders, and end of review period is reached. This is the main flow that triggers daily.
Perf Review - Send reminder emails	Flow	Triggers reminder emails on the date indicated by sys_properties.
Perf Review - Create records	Flow	Triggers review records on the date indicated by sys_properties.
Perf Review - Set review inactive	Flow	Inactivates reviews on the end date of the review period, specified in sys_properites.
PR Admin Self - Eval Submitted w/o Manager	Email	Notifies admins if a self-evaluation is submitted without a manager or title.
PR Reminder Employee Update Self Eval	Email	Notifies employees to update self-evaluation.
Perf Review - Send Manager/ Admin Self-Evaluation Submitted email	Flow - record	Triggers the PR Admin Self-Evaluation Submitted without manager or the PR Manager Self Evaluation Submitted email.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

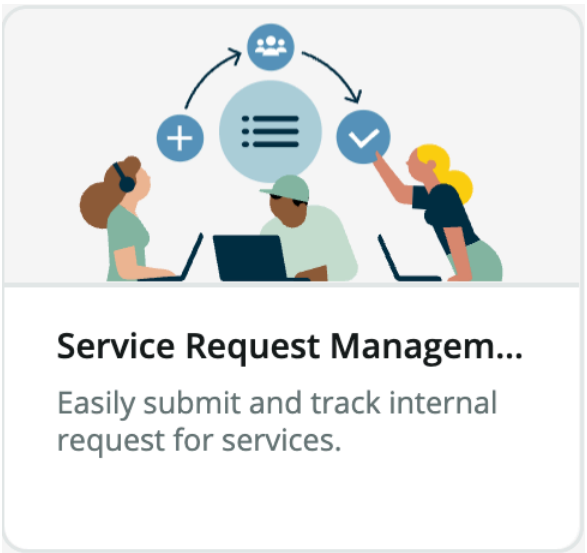
Roles in the Performance Review app template

Role title [name]	Description	Contains roles
Employee [x_<company-name>_ <my performance_review_app>.employee]	Employees can complete/ update self-evaluations and confirm evaluations.	Read (own) and Update (own)
HR/Admin [x_<company-name>_ <my performance_review_app>.admin]	HR/Admins can schedule reviews (annual, bi-annual, custom, or manual), edit self and manager evaluation templates, and monitor the status of all reviews.	Create, Read (own), Update (own), Delete (all)
Manager [x_<company-name>_ <my performance_review_app>.manager]	Managers can request updates to a self-evaluation, submit manager-evaluations, and monitor issues related to evaluation submissions.	Read (my team) and Update (my team)

Service Request Management template

Build an application in App Engine Studio (AES) using the Service Request Management template to create, track, and manage service requests in your organization.

To use the Service Request Management template, install the application from the ServiceNow Store.



Personas in the Service Request Management template

This template contains several important roles that correspond to individuals or groups who may use the app in some way. The following information describes how each persona may interact with the application you build. For more information on the permissions each role contains, see the Security section.

Requester

You are a person who has a request.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Problem: You need a service or a physical item to complete your responsibilities. You want to know that your request is in progress and reassurance that it is making progress. If you have questions or your request has changed, you want the ability to communicate that to those working on the request.

App objective: Empower employees by guiding them through the process, including what information is required. Make the approval process transparent, and keep them updated on the status and outcome of their request.

Fulfiller

You are a person or team who must complete a task in a timely manner.

Problem: You only want to be assigned a task when it is ready to be fulfilled, meaning after required approvals or dependent tasks (if relevant) are closed. You need to ensure that requests are closed in a timely fashion, especially in adherence with SLAs.

App objective: Direct the fulfiller's attention to the tasks assigned to them and their team. Empower them by providing the information they need such as request information and SLA.

Fulfiller's manager

You are the manager of the fulfillment team, and you can fulfill tasks.

Problem: You need to ensure tasks are being efficiently closed by your team. You want to spot bottlenecks such as popular service requests creating a backlog of tasks, a fulfiller who is under-performing, or late approvals blocking fulfillers from completing their tasks on time.

App objective: Direct the fullfiller's manager to tasks that need immediate attention so SLAs are met. Clearly show what their team is working on so they can re-prioritize work as needed.

Service owner

You are the owner of the services within the app.

Problem: You need to analyze how services are performing and identify which services need attention. You need information that will help you recommend improvements to the process, if necessary.

App objective: Provide aggregate information that indicates each service's performance, measured by resolution and response time.

Approver

You are the gatekeeper and can respond to or reject requests, saving time for fulfillers.

Problem: You must review requests before they are assigned to a fulfillment team to ensure the ticket is appropriate.

App objective: Direct the approver's attention to the requests that require their approval. Empower them by providing the information they need such as request and SLA information.

Related concepts

For information about how to set up a service for your Service Request Management application, see Service catalog setup.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Service Request Management template contents

The following tables, experiences, automated workflows, and roles are included in the Service Request Management template for App Engine Studio (AES).

Tables in the Service Request Management app template

Table label [name]	Description
Primary [x_ <company-code>_<primary>]	Table that extends the Task table and stores the details of requests raised for the Service Request Management: Primary catalog item. To create a new service and table, extend this Primary table.
Example Request [x_ <company-code>_<example_request>]	Example Request table created by extending the Primary table that stores the details submitted from the Travel Request catalog item.
Fulfiller Task [x_ <company-code>_<fulfiller_task>]	Stores fulfiller tasks and helps admins keep service requests organized.

Experiences in the Service Request Management app template

Experience	Experience type	Description
Travel Request | Fulfiller view	Form	Form for fulfillers to create a travel request record.
Primary | Fulfiller view	Form	Form for fulfillers to create a service request record.
Fulfiller task | Default view	Form	Form for users to create a fulfiller task.
Fulfiller Workspace	Workspace	Workspace for fulfillers to see requests that need their attention, create requests, view request assignments, and see data and analytics related to their requests.
Approver Workspace	Workspace	Workspace for approvers to see requests pending their approval, approve or deny requests, and see data and analytics related to their requests.
Fulfiller Task | Fulfiller task view	Form	Form for users to create a fulfiller task.
Service Request Management: Primary	Record producer	Contains questions to add information and attach documents relevant to a service request.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Experience	Experience type	Description
Travel Request	Record producer	Contains questions to add information and attach documents relevant to a travel request.
Primary | Default view	Form	Form for users to create a service request record.
Fulfiller's Manager	Workspace	Workspace for the fulfiller's manager to see requests pending their approval, create requests, approve or deny requests, and see data and analytics related to their and their team's requests.
Travel Request | Default view	Form	Form for users to create a travel request record.

Automated workflows in the Service Request Management app template

Workflow name	Workflow type	Description
Fulfiller task work notes added	Email	Notifies users in the Assigned to field, Work notes list, and Watch list when work notes are added to a fulfiller task.
Fulfillment Process Trigger	Email	Notifies a service owner when a fulfiller task has been created but has no default Fulfiller or Fulfillment Assignment Group.
On Insert empty Assigned to	Email	Notifies an assignment group when a request record has been created and has no assignee.
On Insert/Change Assigned to	Email	Notifies a user that a request record has been assigned to them.
Assignment Process	Flow - record	Creates and updates a new record as it moves through the assignment, approval, and fulfillment processes.
Fulfiller Decision	Decision table	Defines the default fulfiller user and fulfiller assignment group for each service.
Approval Process	Flow	Moves a record through the approval process.
Approver Decision	Decision table	Defines the default approver and approver assignment group for each service.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Workflow name	Workflow type	Description
Fulfillment Process	Flow	Moves a record through the fulfillment process.
Child worknotes added	Email	Notifies users named in the **Assigned to** field, Work notes list, and Watch list when work notes are added to a travel request task.
Primary worknotes added	Email	Notifies users in the **Assigned to** field, Work notes list, and Watch list when work notes are added to a request record on the Primary table.

Roles in the Service Request Management app template

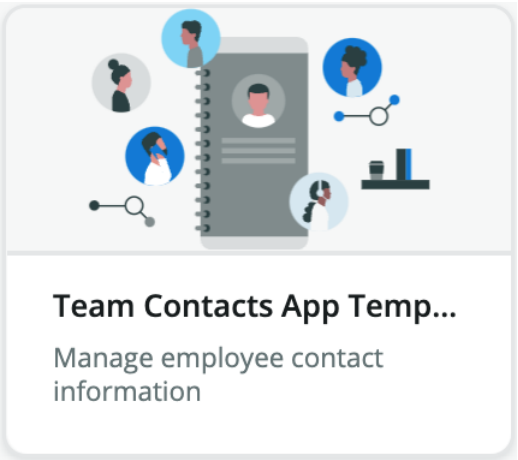
Role title [name]	Description
Approver [x_<company-name>_ <app_name>.approver]	Provides access to approver workspace and provides read and write access to the Primary [x_ <company-code>_<primary>] and Travel Request [x_ <company-code>_<travel_request>] tables.
App admin [x_<company-name>_ <app_name>.app_admin]	Provides complete access to the Service Request Management data and workspaces.
Fulfiller's manager [x_<company-name>_ <app_name>.fulfillers_manager]	Provides access to fulfiller's Manager workspace and provides read and write access to the Primary [x_ <company-code>_<primary>], Travel Request [x_ <company-code>_<travel_request>], and Fulfiller Task [x_ <company-code>_<fulfiller_task>] tables.
Requestor [x_<company-name>_ <app_name>.requestor]	Provides access to submit a request for default record producers from a service portal and provides read and write access to the Primary [x_ <company-code>_<primary>] and Travel Request [x_ <company-code>_<travel_request>] tables.
Fulfiller [x_<company-name>_ <app_name>.fulfiller]	Provides access to the fulfiller workspace and provides read and write access to the Primary [x_ <company-code>_<primary>], Travel Request [x_ <company-code>_<travel_request>], and Fulfiller Task [x_ <company-code>_<fulfiller_task>] tables.
admin [x_<company-name>_ <app_name>.admin]	The System Administrator role. This role has access to all system features, functions, and data, regardless of security constraints.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Team Contacts template

Build an application in App Engine Studio (AES) using the Team Contacts template to manage contact information for the people in your organization.

To use the Team Contacts template, install this application from the ServiceNow Store.



A team contacts app uses a mobile experience to view employee contact information, tag favorites, and add privacy notes. The app displays report information by default and lets an employee search for employees, view employee profiles, and create a list of favorite employee contacts.

For more information on creating an application using a template, see Create your app using an application template.

Team Contacts template contents

The following tables, experiences, automated workflows, and roles are included in the Team Contacts template for App Engine Studio (AES).

Tables in the Team Contacts app template

Table label [name]	Description
Relationship Contact Table [x_<company-code>_<team_contact>_relationship_contact]	Tracks employee contact information. For example, contact names, date created, favorites, and updates.
Reminder Table [x_<company-code>_<team_contact>_relationship_contact_reminders]	Tracks reminders created by your team. For example, when to send a reminder and the subject of a reminder.
Relationship Notes Table [x_<company-code>_<team_contact>_relationship_notes]	Tracks the notes that employees create for each other. For example, the employee name, reason for contact, HTML notes, and option to mark the contact as a favorite.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Experiences in the Team Contacts app template

Experience	Experience type	Description
Employees Mobile experience	Mobile experience	Provides employees access to an employee directory and relationship notes using your organization's iOS or Android app.
Team contact	Portal	Portal where employees and managers can view employee profiles, mark favorite contacts, and add relationship notes using a website. The team contacts portal includes a sign-in page, a home page, and a page to add relationship notes.
Note | Default view	Form	Form to add notes to contact information.
Relationship contact	Form	Form to view contact information.
Reminder | Portal	Form	Form for employees and managers to set reminders.
Reminder | Default view	Form	Form for employees and managers to set reminders.
Note | Portal	Form	Form for managers and employees to add notes to contacts.

Automated workflows in the Team Contacts app template

Workflow name	Workflow type	Description
Team Contact Reminder Push	Email	Sends a reminder push to employees.
Notify Contact Reminder	Flow - record	Sends a notification about a reminder that an employee sets up in a relationship note. The notification contains details from the relationship note.
Team Contact Reminder Notification	Email	Sends a reminder to employees.
Team Contact Reminder Ackn Notification	Email	Sends employees a reminder to acknowledge a notification.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Roles in the Team Contacts app template

Role title [name]	Description	Contains roles
Manager [x_<company-name>_ <app-name>.manager]	Managers can view profile info, add relationship notes, and see their direct reports.	Create, Read (own), Update (own), Delete (own)
Admin [x_<company-name>_ <app-name>.admin]	Managers can configure the team contacts app and grant access to the app.	Read (all), Update (all), Delete
Employee [x_<company-name>_ <app-name>.employee]	Employees can view information on their team, search for employees, view employee profiles, and mark favorite contacts.	Create, Read (own), Update (own), Delete (own)
Canvas user	User role for Canvas Core application	

Time Off template

Build an application in App Engine Studio (AES) using the Time Off Requests template to schedule employee vacations and time-off.

To use the Time Off template, install this application from the ServiceNow Store.

Create a time off application to streamline the process for employees to request time off through a single mobile or web-based form. Managers are notified via email when a request is submitted. They can use a mobile app or a web-based portal to view historical and pending requests by month or year for an employee or their entire team. This allows the manager to make informed decisions, improve planning for upcoming work, or identify where shifts must be covered. Employees are automatically notified when their requests are approved or rejected and why. If plans change and updates to an approved request are required, the employee can edit and resubmit their requests for approval through the portal.

For more information on creating an application using a template, see Create your app using an application template.

Time Off template contents

The following tables, experiences, automated workflows, and roles are included in the Time Off template for App Engine Studio (AES).

Tables in the Time Off app template

Table label [name]	Description
Time Off Request Table [x_<company-code>_<my_time_off_request_app>_request]	Track time off requests that employees create using the Time Off Request app. For example, the employee name, creation date, manager, requested days off, and state of the time off request.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Experiences in the Time Off app template

Experience	Experience type	Description
Time Off Request | Default view	Form	Form for managers and admins to use as a template for time off requests.
Time Off Request | Workspace	Form	Workspace where managers can view, submit, and approve or reject time off requests. The workspace also contains historical records which can be viewed by team, employee, and time.
App name | Portal	Portal	Portal for managers and admins to view, submit, and approve or reject time off requests and for employees to view, edit, review, or cancel their time off requests.
App name | Workspace	Workspace	Workspace where managers and admins can use view, submit, and approve or reject time off requests.
Time Off Request | Record	Form	Form that managers can use to view, submit, and approve or reject time off requests.
Time Off Request | New	Form	Users can create new (save, submit, or cancel), update (save, submit, delete, or cancel), and review (approve or reject) time off requests.
Mobile app	Mobile experience	Employees can request time off and access their requests. Managers can submit, view, update, approve or reject time off requests using your organization's iOS or Android app.
Dashboard	Mobile experience	Managers and admins can view time off requests using your organization's iOS or Android app.

Automated workflows in the Time Off app template

Workflow name	Workflow type	Description
Time off request - Send comments	Flow - record	Sends a notification to an employee or manager when a

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Workflow name	Workflow type	Description
		comment has been added to a time off request.
Daily Time Off inactivating records	Flow	Updates the remaining time off on Time off request records until the remaining days is set to 0.
Time Off Req Approved	Email	Notifies an employee of an approved time off request.
Time Off Req Updated/ resubmitted	Email	Notifies managers of updated/ resubmitted time off requests.
Time Off Comments	Email	Notifies an employee or manager of a comment added to a time off request.
Time Off requests - Send notifications	Flow - record	Sends a notification to an employee or manager when a time off request state changes. The notification contains details from the time off request.
Time Off Request Cancelled	Email	Notifies an employee or manager of a cancelled time off request.
Time Off Request Rejected	Email	Notifies an employee of a rejected time off request.
Time Off Request - updated/ resubmitted	Flow - record	Sends a notification to an employee or manager that a time off request is updated/ resubmitted.
Time Off Req Submitted	Email	Notifies a manager when a time off request is submitted.
Time Off Request	Email	Employees can draft, request, approve, reject, and cancel time off requests.

Roles in the Time Off app template

Role title [name]	Description	Contains roles
Manager [x_<company-name>_ <my time_off_request_app>.manager]	Managers can submit, view, approve, and reject time off requests	Create, Read (all), Edit (all)
Employee [x_<company-name>_ <my time_off_request_app>.employee]	Employees can submit, view, edit, and cancel time off requests	Create, Read (own), Edit (own), Delete (own)

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Role title [name]	Description	Contains roles
Admin [x_<company-name>_ <my time_off_request_app>.admin]	Admins can manage the time off app and view all records	Create, Read (all), Edit (all), Delete (all)
Public	Public users require no login to access features or functions within the application.	Role contains no access by default.

Build a custom template

App Engine Studio (AES) provides various predefined application templates. However, creating custom templates can speed up time to production if you have types of applications that you want to create more than once.

You can create a custom template from scratch or use existing content as a starting point. If one of the predefined AES templates is close to what you need, consider extending that template and then customizing it for your needs. Complete your template by adding and customizing data, experiences, logic and automation, and security roles for access.

Build a custom template from an existing application

Use an existing application as a starting point to build a custom template in App Engine Studio (AES). Existing data, experience, logic and automation, and security roles from the application are automatically added to the custom template.

Before you begin

Role required: admin, app_template_author

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the Template page, select **Create new template**.
3. Select **Existing app**, and then select **Continue**.
4. On the app selection screen, select an existing application for the custom template creation, and then select **Continue**.
   The following outcomes are possible.
   - All items are supported: The custom template is created. All the objects are available in the custom template and in apps created from the custom template.
   - Some items aren't supported: You can continue to create the template but a few items are not available in the custom template. There's no impact on the custom template or the apps created from the template.
   - Some items are denied: You encounter an error message and cannot continue further until the errors are fixed. The author must change the app, if it was created from an existing app or the template, if it was created from scratch.

   Note: For more information, see Supported features and metadata in custom templates.
5. Enter a name and description for the custom template.
6. Accept or change the application's logo.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- If the existing application has a logo in SVG format, the same logo is applied to the custom template.
- If there's no logo or the logo is in a format other than SVG, upload an SVG logo for the custom template by either dragging the image to the **Browse or drag an SVG image** field or selecting the field, selecting the image from your file directory, and selecting **Open**.

7. Select **Continue**.
8. Specify which users or groups should have access to this template.

Option	Description
Share with specific users and groups	**a.** From the **Shared with** list, select **Specific users and groups**. **b.** Add users individually or in groups. **c.** Select **Give access**. **d.** Select **Continue**.
Share with all users and groups	**a.** From the **Shared with** list, select **All users and groups**. **b.** Select **Continue**.

For more information about template sharing and permissions, see App template sharing.

9. Access the custom template in one of the following ways.

Option	Action
Go to all templates	Access custom templates from the template library.
View template	Access the custom template you just created.

Result
Your custom template is created and ready for your use. The custom template is activated and is available to App Engine Studio users who have access to it.

Build a custom template from scratch

Build your custom template from scratch if none of the predefined App Engine Studio (AES) templates fit your business needs or you don't have an existing application as a starting point.

Before you begin
Role required: admin, app_template_author

Procedure
1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the Template page, select **Create new template**.
3. Select **Start from scratch**, and then select **Continue**.
4. Enter a name and description for the custom template.
5. **Optional:** Add a logo to the custom template by either dragging the image to the **Browse or drag an SVG image** field or selecting the field, selecting the image from your file directory, and selecting **Open**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

6. Select **Continue**.
   A progress message indicates when the empty custom template has been created.
7. Specify which users or groups should have access to this template.

Option	Description
Share with specific users and groups	**a.** From the **Shared with** list, select **Specific users and groups**. **b.** Add users individually or in groups. **c.** Select **Give access**. **d.** Select **Continue**.
Share with all users and groups	**a.** From the **Shared with** list, select **All users and groups**. **b.** Select **Continue**.

   For more information about template sharing and permissions, see App template sharing.
8. On the template details screen, select **Go to template dashboard**.
9. Add data, experience, logic and automation, and security measures to the custom template. For more information, see the following topics:
   - Create a data model for your application
   - Add an application experience
   - Add logic and automation
   - Add application security
10. Select **Save**.

What to do next

Provide an overview of the custom template and activate the template to make it available to App Engine Studio users through the template library. For more information, see Update custom template contents and properties.

Identifying and fixing app template creation errors

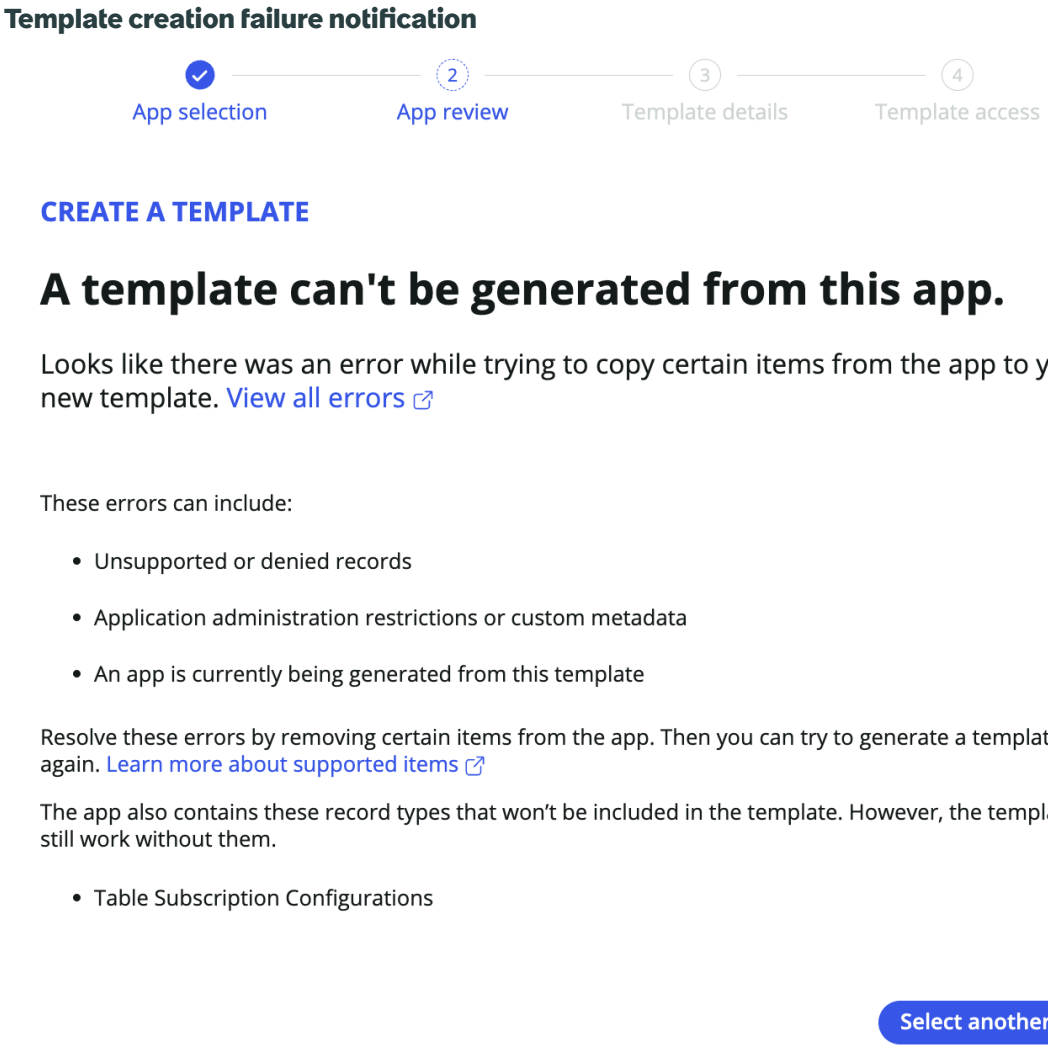
Learn how to identify and fix your app when unsupported features or metadata stop the template creation process in App Engine Studio (AES).

Unsupported features

Sometimes items in a source application from which you created a custom template aren't supported in the template. You must remove unsupported items before you can create a template from the application.

When an error happens during the template creation process, an error message is displayed.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



You can identify which elements of your application are causing it to fail the template creation process by selecting **View all errors**, which displays the Scan Failure [sys_app_scan_failure] table. For more information about the errors in the table and their cause, see Template creation errors in the Scan Failure table.

Template creation errors in the Scan Failure table

Errors that occurred during the creation of a custom template that are displayed in the Scan Failure [sys_app_scan_failure] table and their cause.

Error messages

For more information about what types of records are allowed and skipped, see Supported features and metadata in custom templates.

Error message	Definition	More details
Unsupported_table	Record class for which there is no allow rule, and there's an unknown level of support.	One or more unsupported records are unrecognized by App Engine Studio and must be removed for the template creation process to continue.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Error message	Definition	More details
Denied_table	Record class for which there is a deny rule, and template creation isn't supported.	The record class is explicitly denied and must be removed for the template creation process to continue.
Scripted_record	Scripted rule that runs on each record in the app.	Current scripted deny rules include, but aren't limited to: • Custom metadata: The app has tables that extend the sys_metadata table. • Scoped administration: The app has scoped administration turned on. • Template execution in progress: The template associated with the source app is currently being executed.
Scripted_app	Scripted rule that runs once for the app.	

Assign permissions to publish custom templates

Assign permissions for developers to publish a custom template to the application repository or the ServiceNow Store. You can grant permissions to publish either a specific custom template or publish all existing custom templates in App Engine Studio (AES).

Assign permissions to publish an individual custom template

Assign permissions to developers so they can publish an individual App Engine Studio (AES) custom template to the application repository or ServiceNow Store.

Before you begin

Role required: admin

Procedure

1. Navigate to **All** > **App Engine** > **Templates** > **App Templates**.
2. Select the personalize List icon (⚙).
3. Add the Package column by moving the **Package** value from the **Available** list to the **Selected** list and select **OK**.
4. From the Package column, select the package link for your custom template.
5. Under the **Related links** heading, select **Manage Collaborators**.
6. Assign permissions to new or existing collaborators.

Choice	Action
Assign permissions to a new user or group	**a.** In the **Invite people by name or group** field, enter a user name or group name to add as a collaborator. **b.** From the collaboration descriptor, select **Editor**. **c.** Select **Send**. After the collaboration request is approved, the user or group is added as a collaborator for the custom template.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Choice	Action
	d. From the collaboration descriptor, select **Customize**. **e.** On the Manage Collaborators window, se lect the **Publish To App Repo** and **Publish To App Store** check boxes. **f.** Select **Save**.
Assign permissions to an existing user or group	**a.** From the list of collaborators, select a user or group. **b.** Select **Customize**. **c.** On the Manage Collaborators window, se lect the **Publish To App Repo** and **Publish To App Store** check boxes. **d.** Select **Save**.

Result
The specified user or group now has permission to publish an individual custom template.

Assign permissions to publish custom templates and apps

Assign permissions for developers to publish custom App Engine Studio (AES) templates and applications to the app repository or ServiceNow Store.

Before you begin
Role required: admin

Procedure
1. Navigate to **All** > **App Engine** > **Collaboration** > **Descriptors**.
2. In the **Name** field of the All Collaboration Descriptors page, search for and select **Owner**.
3. From the **Development Permission Sets** tab, select **Edit**.
4. Move the **Publish To App Repo** and **Publish To App Store** values from the **Collection** list to the **Development Permission sets** list.
5. Select **Save**.

Result
Developers who own applications now have permissions at a platform level to publish all custom templates to their application repository or ServiceNow Store.

Publish a custom template

You can publish a custom template that is built from scratch or from an existing application using App Engine Studio (AES).

Before you begin
You must have permissions to publish a custom template. For details, see Assign permissions to publish custom templates.

Role required: admin, app_template_author

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio** > **Templates**.
2. Point to the custom template, select the More actions icon (•••), and select **Properties**.
3. On the **General** tab, select the **Activate this template** check box.
4. Select **Publish template**.
5. **Optional:** For a custom template built from scratch, select **Publish anyway**.

   (Optional) App Engine Studio displays the publish template fields.
6. Fill in the fields.

Provide a version number and release notes

Field	Description
Version	Provide a new version number for the custom template.
Release notes	Provide a brief description of the changes made to the custom template.
Publish options	Select where you want the custom template published. ○ To publish the template to your application repository, select **My application repository (Make the app available to my instances)**. ○ To publish the template to the ServiceNow Store, select **ServiceNow Store (Submit this app to the ServiceNow Store for review)**.

7. Select **Continue**.
8. On the confirmation screen, select **Close**.

Result
The custom template is published and available to users the app template admin shares it with.

Update custom template contents and properties

Improve the effectiveness of your custom template by updating its contents or properties in App Engine Studio (AES).

Before you begin
Role required: admin, app_template_author

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. Select the **Template** tab.
3. Modify different aspects of the template to make them more useful to your needs.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Choice	Procedure
Update the template contents	**a.** Point to the custom template, select the More actions icon (•••), and select **Edit Template**. **b.** Modify the custom template by editing da ta, experience, logic and automation, or se curity. For more information, see the following top ics: ▪ Modify application data tables ▪ Add an application experience ▪ Add logic and automation ▪ Add application security **c.** Select **Save template**.
Update the template properties	**a.** Point to the custom template, select the More actions icon (•••), and select **Edit Properties**. **b.** Select **General**. **c.** Modify aspects of the template. ▪ To rename the custom template, update the **Name** field. ▪ To rewrite the custom template descrip tion, update the **Description** field. ▪ To replace the custom template logo, se lect **Remove image** and then add a new logo. **d.** Select **Save properties**.
Update template sharing settings	**a.** Point to the custom template, select the More actions icon (•••), and select **Edit Properties**. **b.** Select **Sharing settings**, and share the tem plate in one of the following ways: **i.** Share with some users or groups. **ii.** Share with all users and groups. **c.** On the General tab, select **Save properties**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Choice	Procedure
Update content on the Template details page	**Note:** Only templates created from scratch have a Template details page that you can edit. Templates created from existing templates don't have a Template details page to edit, so you aren't able to change items such as the header, tag line, or other template information. **a.** Point to the custom template, select the More actions icon (•••), and select **Edit Properties**. **b.** Select **Details page**. **c.** Update details about the template that help users understand its features and why they should choose this template for their application. **d.** Select **Save properties**.
Delete the custom template	To delete the custom template, ensure you have administrator privileges or have permissions to delete. Select **Delete template**, type `Delete` to confirm your choice, and select **Delete**. **Note:** To request delete permissions, contact your administrator.

Manage template activation

Manage template activation and deactivation to ensure that users creating apps in App Engine Studio (AES) have access to the best application templates for their needs.

Before you begin

Role required: app_template_admin

About this task

App template admins manage the templates that are active in non-production instances and determine which users can see and use those templates. Admins may want to deactivate custom templates that deviate from company standards or don't suit the needs of their users.

Procedure

1. Navigate to **All** > **App Engine** > **Templates** > **App Templates**.
2. Select the template that you want to activate or deactivate.
3. To edit the template, select **here** in the message that appears.
4. Activate or deactivate the template.
   - To activate the template, select the **Active** check box.
   - To deactivate the template, clear the **Active** check box.
5. Select **Update**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

App template sharing

Specify which users and groups have access to predefined and custom templates in App Engine Studio (AES).

Implicit sharing

Templates are implicitly shared with some users in your organization, meaning that no sharing record exists that determines sharing. The three rules governing implicit sharing are:

- Instance administrators can see and use all templates.
- App template admins (users with the app_template_admin role) can also see and use all templates.
- App template authors (users with the app_template_author role) can see templates that are not shared with them if they have permission to edit the scope in which the template resides.

These sharing rules apply regardless of whether a template is active.

Explicit sharing

You can share app templates with anyone in your organization by creating a sharing record for that user or group. However, regardless of sharing permissions, the user or group must have the sn_app_eng_studio.user role to see and use the template. The three ways to share templates are:

- Share globally with every user in your organization
- Share with a specified user group
- Share with a specified single user

Note: If you stop sharing a template with an individual user but that user is a part of any group the template is shared with, that user is still able to see and use the template.

Manage template sharing permissions

Share App Engine Studio (AES) templates globally, with groups, or with individual users to ensure users have access to the right templates.

Before you begin

Role required: app_template_admin

Procedure

1. Navigate to **All** > **App Engine** > **Templates** > **App Templates**.
2. Select the template you want to modify sharing permissions for.
3. In the Template Sharing Permissions related list, select **New**.
4. Select a sharing option.

Choice	Permissions
Global	Shared with everyone in your organization.
User Group	Shared with the groups you select from the list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Choice	Permissions
User	Shared with individual users you select from the list.

5. Select **Submit**.

Create your application from scratch

If the available application templates in App Engine Studio (AES) don't fit your application goal, create an empty application to which you can add data, experience, logic and automation, and security.

Before you begin

Note: To create apps in AES, you must be an admin or in the App Engine Studio Users group. If you are in the App Engine Studio User Limited group, you can only edit existing apps, not create new ones.

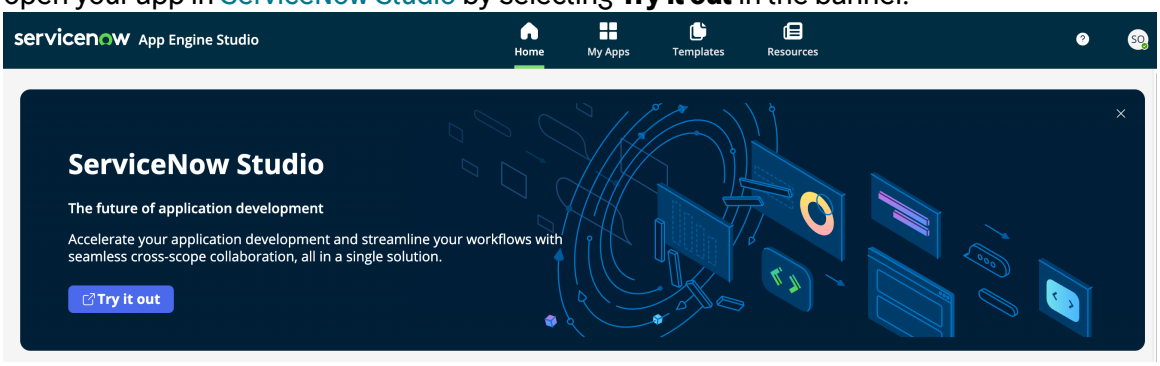
Role required: admin, sn_app_eng_studio.user

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. Select **Create app**.
3. Enter a **Name** and **Description** for your application.
4. **Optional:** Add a logo by either dragging the image to the **Drag app logo or browse to upload** field or selecting the field, selecting the image from your file directory, and selecting **Open**.
5. Select **Continue**.
6. Control who has access to create, read, write, and delete content in your application by adding or customizing roles.
   To add security roles to your app, select **Add a role**, and enter a name and description for the role.

   Note: At least one role must have **Read** access to be able to update application security.

7. Select **Continue**.
8. On the summary screen, select **Go to app dashboard**.
9. **Optional:** If you want a more advanced app development environment to work in, you can open your app in ServiceNow Studio by selecting **Try it out** in the banner.



10. **Optional:** Review the guiding information for each section of your application by enabling the **Guidance** toggle ( ) in the bottom left corner of the window.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

11. Continue building your application by adding data, experiences, logic and automation, and security to your app.
    For more information, see the following topics:
    - Create a data model for your application
    - Add an application experience
    - Add logic and automation
    - Add application security

What to do next

After you've finished building your application, submit the application for approval to get it reviewed and deployed by an administrator. For more information, see Submit your app for approval and publishing.

Prepare your app for approval

Prepare your app for the approval process by making any needed changes, such as renaming your app, editing the app description, or replacing the app image. You can also delete the app from App Engine Studio (AES) if it has taken a wrong turn.

Before you begin

Role required: admin, sn_app_eng_studio.user

About this task

You can change certain features of your application, based on your permissions. To request delete permissions, contact your administrator.

Procedure

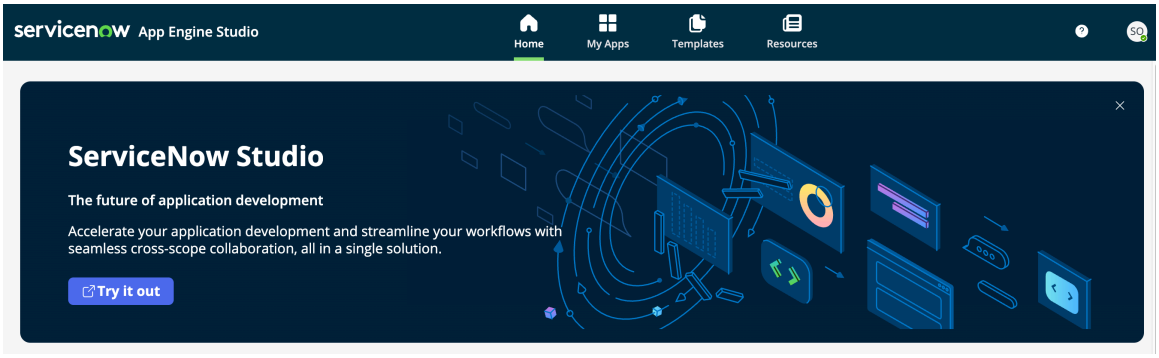
1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. From the app home, select the Edit application properties icon ( ).
4. Update the application properties.
    - To rename the application, update the **Name** field.
    - To rewrite the application description, update the **Description** field.
    - To replace an image, select **Remove image** and then add a new image.
    - To delete your application, select **Delete application**, type `Delete` to confirm your choice, and select **Delete**.
5. Select **Save**.

Enhance your app

Whether you use a template or create your app from scratch in App Engine Studio (AES), you must enhance it by creating a data model, user experiences, logic and automation, and security.

If you're creating apps in AES, you may be able to open your app in ServiceNow Studio depending on your permissions. If the ServiceNow Studio banner is displayed when you open your app in AES, select **Try it out**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Use the following topics as guides as you build out your application.

Resources for enhancing your app

As you add data, experience, logic and automation, and security to your application, you might interact with several different parts of the ServiceNow AI Platform. The links in the following table point you to resources for adding content in App Engine Studio and also Table Builder, UI Builder, Decision Builder, App Engine Management Center, and the ServiceNow AI Platform.

Resources for adding content to your application

Add data	Add an experience	Add logic and automation	Add security
Add data to your application in App Engine Studio. • Create a data model for your application • Create a blank table • Use a spreadsheet to add data • Create a data integration • Modify application data tables	Work in App Engine Studio to add experiences to your application. • Add an application experience • Add a record producer • Add a standard catalog item • • Add a workspaceAdd a portal • Add a mobile experience • Editing an experience in App Engine Studio	Build logic and automation into your application in App Engine Studio. • Add logic and automation • Add a pre-built flow using a template • Add a flow from scratch • Add a decision • Add an email notification • Edit a process	Configure roles to manage your application's security in App Engine Studio. • Add application security • Build a new role for your application • Use an existing role for your application • Collaborate with other developers • Change access settings for a role • Delete a role
Seamlessly transition to using Table Builder to enhance your application's data.	Use Workspace Builder and UI Builder to further enhance experiences in your application.	Use Decision Builder to easily manage business logic for your application.	Configure and manage roles for your organization and application using the ServiceNow AI Platform and App

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Resources for adding content to your application (continued)

Add data	Add an experience	Add logic and automation	Add security
• Exploring Table Builder • Data in Table Builder • Forms in Table Builder • Policies and rules in Table Builder • Flows in Table Builder	• Add a workspace • Edit a workspace in Workspace Builder • UI Builder and configurable workspaces • Configure how users interact with your applications in UI Builder • Customize UI Builder pages using components	• Create decision tables in Workflow Studio • Manage decision tables in Excel • Modify decision table structure in Workflow Studio • Modify decision table rules in Workflow Studio	Engine Management Center. • Configure AES personas and roles • Configure users and groups • Add users to the App Engine Admin group • Delegate developers using AES • Managing developers using AEMC • Manage collaboration requests

Create a data model for your application

Create a data model in App Engine Studio (AES) for your application by creating the tables that will house your application data. Create data tables that will be used in your application by creating the table from scratch, uploading a spreadsheet, or by using an existing table as a template. You can also create and schedule data imports by mapping a spreadsheet to an existing table in your application.

Note: Starting with the Brazil release, the PDF extractor tool has been deprecated. The feature is hidden and no longer available for installation or use. The capability to create tables from PDFs is available with Build Agent. For more information, see Migrating to Build Agent.

Creating a data model and populating data into user interfaces are crucial starting points for any new application. Users may provide data, such as their name and phone number, when they fill in a form, and other users may refer to data as they fulfill a request.

In App Engine Studio, application data is stored in table format. When users update application data, they create a row or change an existing row in the table (each row is also known as a data record). You can create tables from existing ServiceNow AI Platform tables, from a spreadsheet or PDF, or customize a completely new table for your application to store records in.

The data records stored in your application may come from several sources. One table might store data records entered by users who are using your application, another might be populated via the import and mapping of data from a spreadsheet or through a script that updates data records with data from another ServiceNow AI Platform table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Application templates and data

Application templates automatically add data to your application. If you use a template to create your application, you can edit the tables that were added or add different tables.

Forms

When you add data to your app, any associated form views display in the **Experience** section of your app. To edit the form, select it in the **Experience** section. Form views can be edited in the **Forms** tab of Table Builder. For more information, see Forms in Table Builder.

Table extension

There are several tables that your organization may already be using in other applications. For example, your organization may be using the Configuration item [cmdb_ci] table for an application that houses configuration data. You can create a table for your application by extending any of these existing tables.

Table extension allows a table to share fields and records with a parent table. You might extend a table if you expect users to fill in similar fields across different form views in an application. For example, to create a ticketing type of table, you might extend the Task [task] table. The Task table includes fields that are standard for most work tickets, such as **Number** and **Assigned to**.

Many organizations create extensible standardized tables for their citizen developers to use when designing their apps. This enables the developers to standardize the way data is utilized across multiple applications within the organization.

To make a table in your application extensible, select **Make extensible** during table creation. After extending an existing table, you can further customize the new table by adding more columns.

The following table provides a list of commonly extended tables in the ServiceNow AI Platform. For more information on commonly extended tables and models for extension, see Table extension and classes.

Commonly extended tables

Table	Short Description	Description
task	Task	Stores fields for the core applications such as Incident, Problem, and Change Management. It provides a series of standard fields used on each of the tables that extend it. Any table which extends the task table can take advantage of task-specific functionality for driving tasks.
cmdb_ci	Configuration Item	Stores configuration items. This table can be extended for configuration items such as hardware, services, etc.
sn_customerservice	Customer Service Case	Stores customer service case records.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Commonly extended tables (continued)

Table	Short Description	Description
_case		
sm_order	Service Order	Stores data that defines and manages work that needs to be performed.
sm_task	Service Task	Stores units of work performed by one person in one session (one location, one time).
planned_task	Planned Task	Stores additional fields for tasks pertaining to time and effort as part of a planned, multi-stage process.
cmn_location	Location	Stores location information.
cmdb_serviceorder _product_model	Service Order Model	Stores service order templates.
cmdb_servicetask _product_mode	Service Task Model	Stores service task templates.
cmdb_qb_result _base	Query Results	Stores query results created by the CMDB Query Builder.
alm_asset	Asset	Stores general, financial, and contractual information about assets.
cmdb_ci_service	Business Service	Stores IT Services data that directly supports a business process.
sys_user	User	Stores user data. The User table provides a list of all system users and their related department.
sys_user_group	User Group	Stores and groups a set of users who share a common purpose. Groups may perform tasks such as approving change requests, resolving incidents, receiving email notifications, or performing work order tasks.
core_company	Company	Stores company information.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Commonly extended tables (continued)

Table	Short Description	Description
cmn_schedule	Schedule	Stores records that specify a time zone and a type of schedule and use one or more schedule entries.
cmn_department	Department	Stores a list of all departments and their related business unit.
cmn_cost_center	Cost Center	Stores cost center records. This is a reference between financial systems and IT. Cost center records represent business entities, and have a related list of CI Cost Center Relationships that measure the cost center's consumption of business services.
cmdb_model	Product Model	Specific versions or various configurations of an asset. Models are used for managing and tracking assets through various ServiceNow platform asset applications, including Product Catalog, Asset Management, and Procurement.
life_cycle_stage	Life Cycle Stage	Stores standard fields and values for tracking life cycle stages for CIs. Using these standard values consistently across applications helps to effectively track assets through their life cycle transitions.
life_cycle_stage _status	Life Cycle Status	Stores standard fields and values for tracking life cycle stage status for CIs. Using these standard values consistently across applications helps to effectively track assets through their life cycle transitions.
incident	Incident	Stores deviations from an expected standard of operation.
sysapproval	Approval	Stores data that allows you to require authorization on tasks before the work is done. You can define approvals for all

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Commonly extended tables (continued)

Table	Short Description	Description
		tasks and associate users or groups to a task to approve or reject them.

Use the methods listed below to create a data model and add data to your application.

Create a blank table

Add a data table to your application from an existing table that you or someone else previously created or from scratch.

Before you begin

Role required: admin, sn_app_eng_studio.user, or delegated_developer

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. Next to the Data heading, select the Add icon ( ).
4. Select **Create a blank table**, and then select **Continue**.
5. Choose from the following choices.

Option	Description
Create new table	Select this option to create a blank table from scratch. **a.** Choose **Create new table**. **b.** Click **Continue**.
Create from an extensible table	Select this option to create a new blank table from an existing extensible table in the ServiceNow AI Platform. **a.** Choose **Create from an extensible table**. **b.** Search for and select an existing table. **Note:** Tables already in the app display first in the list and recommended ServiceNow AI Platform tables display next, along with a short description of the type of information the table stores. **c.** Select **Continue**.

6. On the form, fill in the properties of your new table.

Field	Description
Table label	Unique label to identify the table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Table name	Database name for the table. A table name is created automatically after you enter a table label. You can edit the name if needed.
Make extensible	Option to allow other tables to share data from this table. For more information on table extension, see Table extension.
Auto number	Option to track table records with a unique number. If you select this option, define the **Prefix**, **Starting number**, and **Number of digits**.
Prefix	Abbreviated name of the table to append to the beginning of the record number. For example, if you are creating a "Laptop" table, then your prefix may be "LPTP" or "LT."
Starting number	Number to identify the first record created for your table.
Number of digits	Maximum number of digits to allow in the record number. This value determines the highest possible record number. For example, if you enter 7, then the highest possible number is 9999999.

7. Select **Continue**.
8. Control who has access to create, read, write, and delete content from this table by adding new roles and/or defining the desired Create, Read, Write, Delete permissions for existing roles.

   Roles that you create in the table persist throughout the application and can be further defined in other tables you add. See Add application security for more information on how security permissions work.

   Note: At least one role must have read access for you to be able to preview the table.

9. Select **Continue** to add the new table.
10. Return to your application home page or continue editing your table.
    - Select **Done** to return to your application home page.
    - Select **Edit table** to review the columns that were added to your table and continue editing it in Table Builder.
11. If you continue editing, in the new tab that opens, review the table columns.

    Each table column appears as a row in Table Builder.

    A lock icon ( ) appears next to columns from the extended table. You can't modify these columns.

12. Customize your table by adding table columns.

    a. In Table Builder, select **+ Add new field**.

    b. On the empty row, fill in the fields.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Table column properties

Field	Description
Column label	Unique label for the column.
Column name	Database name for the column.
Type	Type of information that the column contains. For example, to contain plain text in the column, select **String**. Depending on the type that you select, fill in the additional fields to further define the table column. For example, if you select **String**, define the character limit of the string input. Or, if you select **Choice**, define the choices that users can choose from. See Field types.
Reference	Table that is associated with the column. This field applies only if the column type is **Reference**.
Max length	Maximum number of characters that users can enter in the field.
Default value	Value that populates the field automatically after a new record is created.
Display	Option to set the column as the display value for the table. A reference field shows the display value of the table to which it is referring. For example, the **Opened by** column of the task table refers to the user table. Because the display value of the user table is the user name, the **Opened by** field shows something like **Beth Anglin** or **Joe Employee**. When you select a display value, choose the table column that would act as an appropriate title for individual records. Only one column can act as the display value for a table.

13. Select **Save**.

What to do next

For more information on Table Builder, see Table Builder.

Use a spreadsheet to add data

Add data tables and records to your application by uploading a Microsoft Excel spreadsheet.

Use the following procedures to upload a spreadsheet and use it to create data tables and import data records. Use the Import a spreadsheet procedure first to upload your spreadsheet, and then create data tables from your spreadsheet import as needed.

Import a spreadsheet

Upload a Microsoft Excel spreadsheet and use it to create your application data model.

Before you begin

Check that the spreadsheet meets the following requirements:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Formatted with horizontal columns and a header label for each
- Saved as an XLSX file type

Note: The Excel spreadsheet you upload can have multiple worksheet tabs, each of which you might use to create a data table. If multiple worksheets are present, you will be prompted to select which worksheet tab to create the table from after you upload the file.

Role required: admin, sn_app_eng_studio.user or delegated developer permissions. For more information, see Delegate developers using AES.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. Next to the Data heading, select the Add icon ( ).
4. Select **Import a spreadsheet**, and then select **Continue**.
5. Upload your spreadsheet and then select **Continue**.

   You can drag the spreadsheet file onto App Engine Studio or browse your computer.
6. You will be presented with one of the following paths:
   - If your spreadsheet only has a single worksheet tab, proceed to the next step.
   - If the spreadsheet has multiple worksheet tabs, you will have a list of sheets to choose from. From this list of sheets, you can select which worksheet tab to add as a table to your application first.

   a. In the **Enter a row number for the table header** field, enter the row number in which the sheet column headers are located.
   For example, if the headers are in the second row, enter 2.

   b. **Optional:** To upload the information below the column headers in the sheet (e.g., the data in your spreadsheet), select the **Import spreadsheet data** option.
   To upload only the column headers, leave the check box cleared. You may want to upload only the headers if you intend to use this table as a template and don't want any of the other data added to your AES table.
7. Make one of the following selections.
   - If you have uploaded an Excel file with multiple worksheet tabs, select **Convert to table** to proceed with converting the selected worksheet tab to a table, and then proceed to next step.
   - If you have uploaded an Excel file with a single worksheet tab, select **Continue**.

   Options display for where to import your file into.
8. Based on what you would like to do with your spreadsheet data, navigate to one of the following procedures to continue.

Option	Description
Create a new table directly from the spread sheet import	Refer to Create new table from spreadsheet import for next steps.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Description
Create a new table by extending an existing table, and then using the spreadsheet import to update it	Refer to Create new table from extensible table and spreadsheet import for next steps.
Use spreadsheet import to update an existing application table	Refer to Modify existing table using spreadsheet import for next steps.

Create new table from spreadsheet import

Create new tables directly from a Microsoft Excel spreadsheet import.

Before you begin

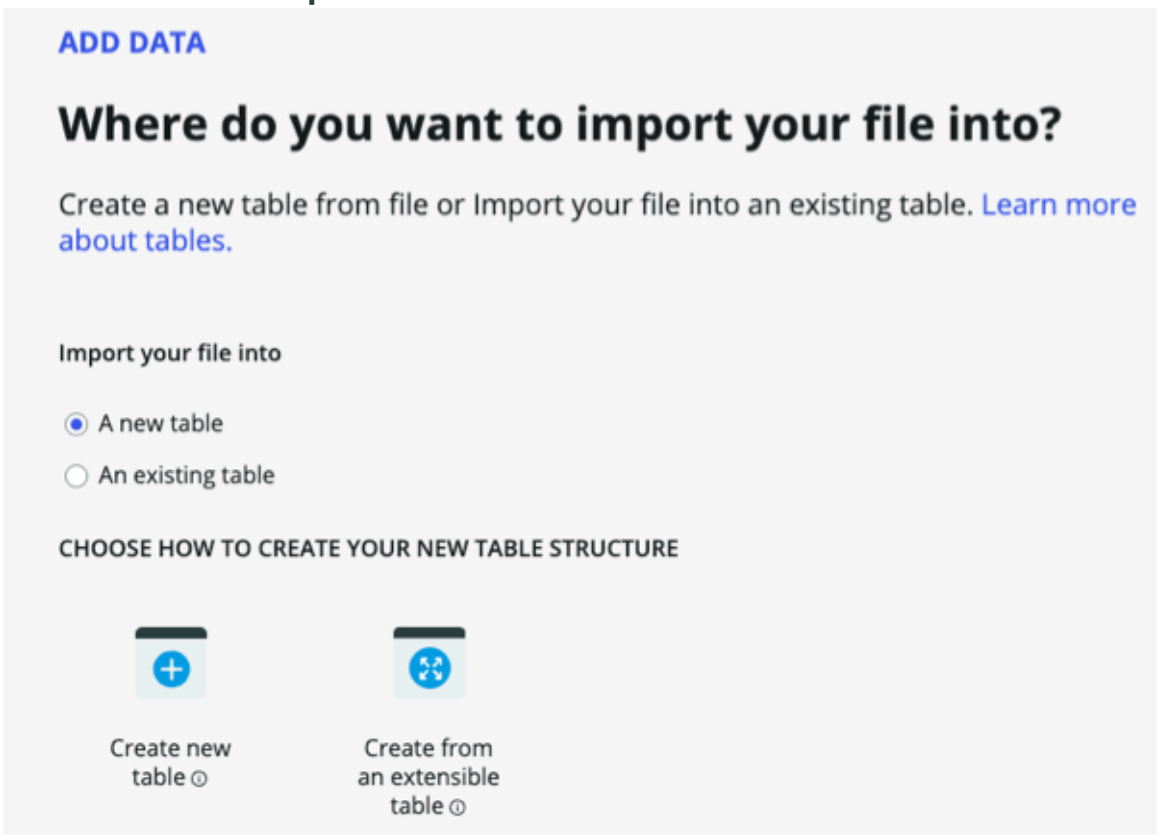
- Import a spreadsheet into App Engine Studio. See Import a spreadsheet.

Role required: admin, sn_app_eng_studio.user or delegated developer permissions. For more information, see Delegate developers using AES.

Procedure

1. After you have uploaded your spreadsheet as described in Import a spreadsheet, choose what you want to do with the imported data in the displayed wizard.

Create new table from spreadsheet



   a. Select **A new table**.
   b. Choose **Create new table**.
2. Select **Continue**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

3. Review the properties for each column header, using the **Expand options** drop-down to display more properties.

Column header property	Description
Field Label	Unique label for the column.
Field name	Database name for the column.
Type	Type of information that the column contains. For example, to contain plain text in the column, select **String**. Depending on the type that you select, fill in the additional fields to further define the table column. For example, if you select **String**, define the character limit of the string input. Or, if you select **Choice**, define the choices that users can choose from. See Field types reference.
Character limit	Max length of the strings that can be stored.
Display	Option to set the column as the display value for the table. A reference field shows the display value of the table to which it is referring. For example, the **Opened by** column of the task table refers to the user table. Because the display value of the user table is the user name, the **Opened by** field shows something like **Beth Anglin** or **Joe Employee**. When you select a display value, choose the table column that would act as an appropriate title for individual records. Only one column can act as the display value for a table.
Mandatory	Option to require that the column must contain a value before a new record can be saved.

To add another column to your table, select **Add new field**.

To delete a column, select the trash icon ().

4. Select **Continue** to define table properties.
5. On the form, fill in the fields.

Field	Description
Table label	Unique label to identify the table.
Table name	Database name for the table. A table name is created automatically after you enter a table label. You can edit the name if needed.
Make extensible	Option to allow other tables to share data from this table. For more information on table extension, see Table extension.
Auto number	Option to track table records with a unique number. If you select this option, define the **Prefix**, **Starting number**, and **Number of digits**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Prefix	Abbreviated name of the table to append to the beginning of the record number. For example, if you are creating a "Laptop" table, then your prefix may be "LPTP" or "LT."
Starting number	Number to identify the first record created for your table.
Number of digits	Maximum number of digits to allow in the record number. This value determines the highest possible record number. For example, if you enter 7, then the highest possible number is 9999999.

6. Select **Continue**.
7. Control who has access to create, read, write, and delete content from this table by adding new roles and/or defining the desired Create, Read, Write, Delete permissions for existing roles.

   Roles that you create in the table persist throughout the application and can be further defined in other tables you add. See Add application security for more information on how security permissions work.

   Note: At least one role must have read access for you to be able to preview the table.

8. Select **Continue** to add the new table created from your spreadsheet to your app.
9. **Optional:** Convert any remaining worksheets if your file contained multiple worksheet tabs.
   - **Note:** For each sheet that you want to convert, you will need to select **Convert to table**, and then repeat steps in this procedure or choose a different pathway for the additional worksheet (see Create new table from extensible table and spreadsheet import or Modify existing table using spreadsheet import for steps).
   - If you do not have multiple worksheets, proceed to the next step.
10. Select **Done** on the summary screen.
    Alternatively, to view your table in Table Builder, you can select **Edit table(s)** on the summary screen.

What to do next
For more information on editing your data tables in Table Builder, see Table Builder.

Create new table from extensible table and spreadsheet import

Create a new table by extending an existing table, and then using a Microsoft Excel spreadsheet import to update or customize it.

Before you begin
- Import a spreadsheet into App Engine Studio. See Import a spreadsheet.

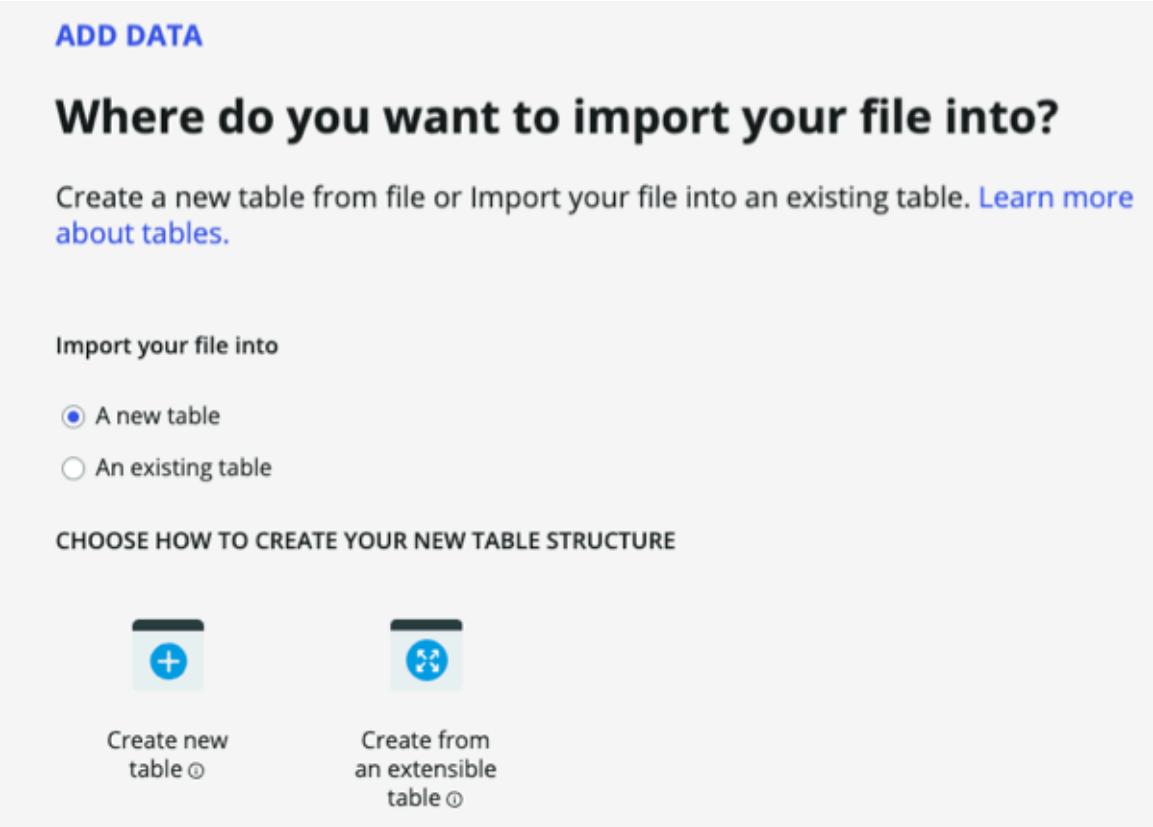
Role required: admin, sn_app_eng_studio.user, delegated developer permissions for "Integration" must be present to view data import and mapping options. For more information, see Delegate developers using AES.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. After you have uploaded your spreadsheet as described in Import a spreadsheet, choose where to import your file in the displayed wizard.

Import into a new data table



   a. Select **A new table**.

   b. Choose **Create from an extensible table**.

2. Select **Continue**.
3. Search for an select an existing **Table** to add that table's data to your app and create an extension with branching logic.

   Note: Tables already in the app display first in the list and recommended ServiceNow AI Platform tables display next, along with a short description of the type of information the table stores.

4. Select **Continue**.
5. On the form, fill in the fields.

Field	Description
Table label	Unique label to identify the table.
Table name	Database name for the table. A table name is created automatically after you enter a table label. You can edit the name if needed.
Make extensible	Option to allow other tables to share data from this table. For more information on table extension, see Table extension.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Auto number	Option to track table records with a unique number. If you select this option, define the **Prefix**, **Starting number**, and **Number of digits**.
Prefix	Abbreviated name of the table to append to the beginning of the record number. For example, if you are creating a "Laptop" table, then your prefix may be "LPTP" or "LT."
Starting number	Number to identify the first record created for your table.
Number of digits	Maximum number of digits to allow in the record number. This value determines the highest possible record number. For example, if you enter 7, then the highest possible number is 9999999.

6. Select **Continue**.
7. Control who has access to create, read, write, and delete content from this table by adding new roles and/or defining the desired Create, Read, Write, Delete permissions for existing roles.

   Roles that you create in the table persist throughout the application and can be further defined in other tables you add. See Add application security for more information on how security permissions work.

   Note: At least one role must have read access for you to be able to preview the table.

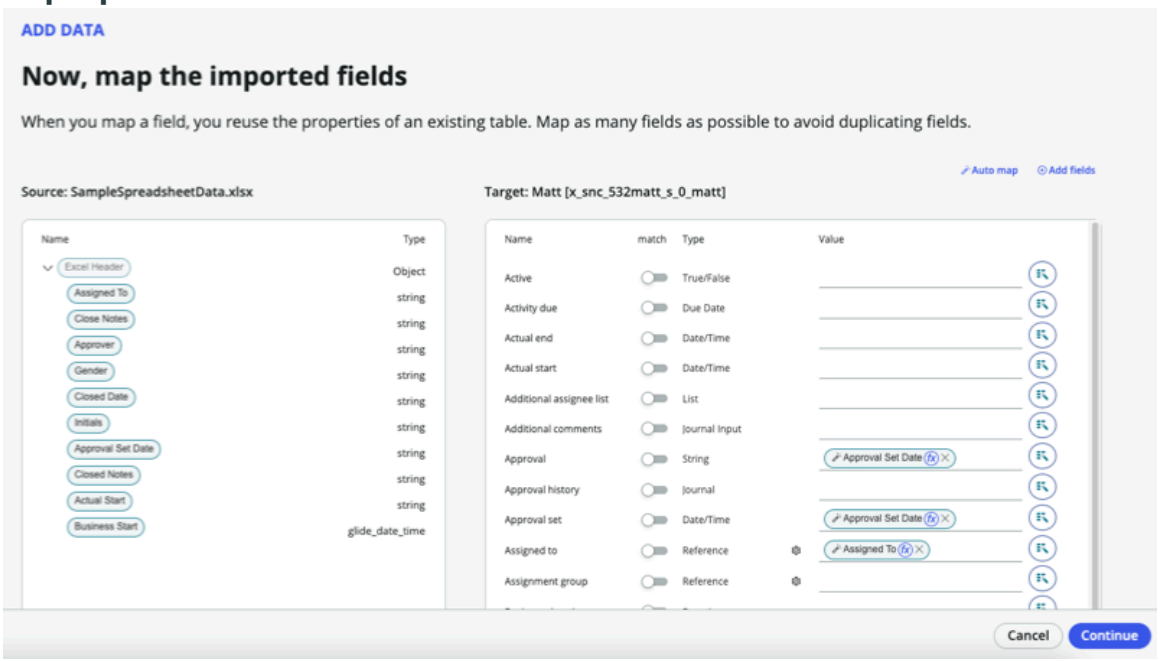
8. Select **Continue**.

   Note: Now your table is ready for data. In the next step, fields from your import are matched with fields on your newly extended table. You will have an opportunity to adjust mapping or create new fields from your import in it.

9. Select **Continue** to proceed to mapping fields.
   A screen displays where you can map field columns from your source spreadsheet on the left to field columns in your newly extended target table that you just created on the right.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Map imported fields



10. You have the following choices available to you during this step.

Option	Description
Batch map fields using Automap	Select **Automap** to automatically map your spreadsheet fields to displayed fields in the selected target table. Automap maps source entities to similarly named target entities. For a target entity that has already been mapped, selecting **Automap** doesn't change its value. An automapped field has a wand icon on its data pill.
Map fields by dragging and dropping	Map fields from your import to your target table by dragging data pills representing columns in the imported spreadsheet to the **Value** column in the target table. You can also map fields by using the pill picker to the right of each target table field.
Map fields using the data picker	Map data by using the pill picker ( ) on the right side of the target table. You can enter more than one value in the **Value** column, you can also enter text.
Add transform options	Add transform options for each of the individual mapped spreadsheet fields by clicking the FX icon ( ) on the right side pane. For example, you can use the Uppercase transform function to change an input string to all uppercase characters. See Transform func for information on how to use these

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Description
	to trim data for your imported spreadsheet fields. **Note:** Not all the transform functions available in Flow Designer are available.
Activate data record matching	Specify fields where the system should check for matching data and, if found, update existing records instead of creating new ones by enabling the **Match** toggle in your target table. For example, if a target table has an **Order number** field and a record with an order number of 743, there are two options for handling potential matches. ○ If the **Match** toggle switch for the **Order number** field is on, the system checks for matching data. If the source data has a record with the same order number, the system updates the existing record in the target table. ○ If the **Match** toggle switch for the **Order number** field is off, the system doesn't check for matching data. If the source data has a record with the same order number, the system creates a new record in the target table. In this case, the target table will have two records, both with an order number of 743.
Configure reference and choice field behavior	Configure behavior for **Choice** and **Reference** field types. **a.** Click the gear icon ( ). **b.** In **Field**, select the field column to match the incoming record value against in the target table. **c.** In **If no matching record exists then do the following**, specify what to do if that record isn't found in the target table. ▪ **Create a new choice/record:** Creates the choice or record in the target table's matching field column from the data imported for the field. ▪ **Ignore this field:** Ignores the field in the target table and leaves it null. ▪ **Skip this record:** Skips adding the entire record (row) in the target table. **d.** Click **Done**.
Add new field columns to target table	Add unmapped field columns from your import to your target table or create new fields in your target table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Description
	a. Select **Add fields** above the target table on the right. **b.** Choose which fields to add to your target table by selecting the checkbox on the left or select **All** to choose all of them. **c.** You can also perform the following functions: ▪ Edit the **Field Label** for the imported field. ▪ Edit field properties for the imported field by selecting the **Edit icon** ( ). ▪ Create a new field by clicking **Create a new field** in the top right corner of the window. **d.** Select **Add fields** to add the selected fields to your target table.

11. Select **Continue** to create the data import mapping and update your target table.
12. **Optional:** Convert any remaining worksheets if your file contained multiple worksheet tabs.
    - **Note:** For each sheet that you want to convert, you will need to select **Convert to table**, and then repeat steps in this procedure or choose a different pathway for the additional worksheet (see Create new table from spreadsheet import or Modify existing table using spreadsheet import for steps).
    - If you do not have multiple worksheets, proceed to the next step.
13. Select **Done** on the summary screen.
    Alternatively, to view your table in Table Builder, you can select **Edit table(s)** on the summary screen.

What to do next

For more information on editing your data tables in Table Builder, see Table Builder.

Modify existing table using spreadsheet import

Use a Microsoft Excel spreadsheet import to update an existing application table.

Before you begin

- Import a spreadsheet into App Engine Studio. See Import a spreadsheet.

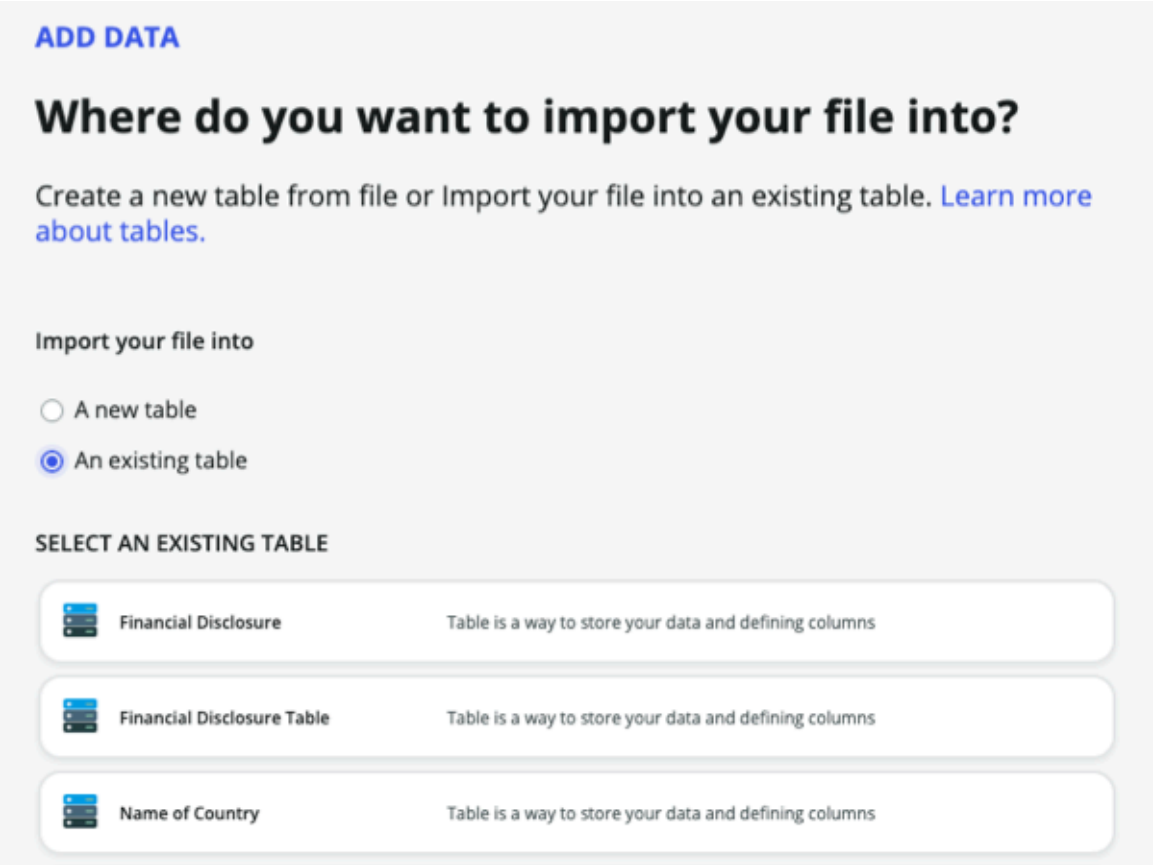
Role required: admin, sn_app_eng_studio.user or delegated developer permissions. For more information, see Delegate developers using AES.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. After you have uploaded your spreadsheet as described in Import a spreadsheet, choose where to import your file in the displayed wizard.

Import into an existing application table



   a. Select **An existing table**.

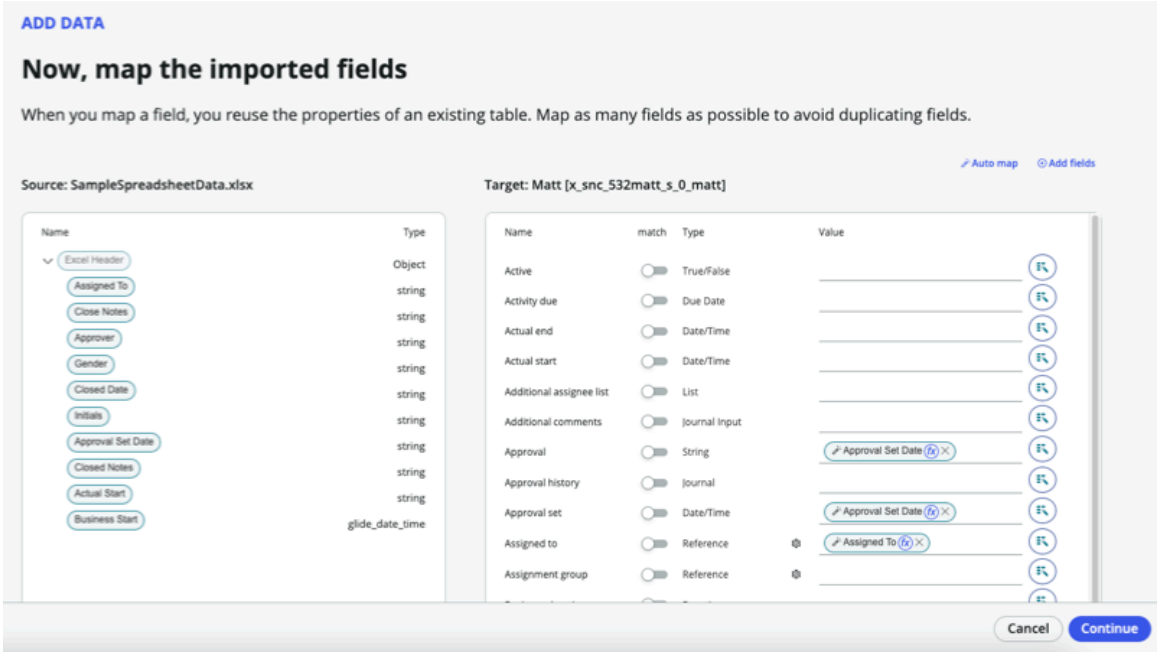
   b. Select an existing application table from the list that displays.

2. Select **Continue**.
   A screen displays where you can map field columns from your source spreadsheet on the left to field columns in your existing target application table on the right.

   Note: You will have the opportunity to adjust mapping or create new fields from your import in it.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Map imported fields



3. You have the following actions available to you during this step.

Option	Description
Batch map fields using Automap	Select **Automap** to automatically map your spreadsheet fields to displayed fields in the selected target table. Automap maps source entities to similarly named target entities. For a target entity that has already been mapped, selecting **Automap** doesn't change its value. An automapped field has a wand icon on its data pill.
Map fields by dragging and dropping	Map fields from your import to your target table by dragging data pills representing columns in the imported spreadsheet to the **Value** column in the target table. You can also map fields by using the pill picker to the right of each target table field.
Map fields using the data picker	Map data by using the pill picker ( ) on the right side of the target table. You can enter more than one value in the **Value** column, you can also enter text.
Add transform options	Add transform options for each of the individ ual mapped spreadsheet fields by clicking the FX icon ( ) on the right side pane. For example, you can use the Uppercase trans form function to change an input string to all uppercase characters. See Transform func or information on how to use these

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Description
	to trim data for your imported spreadsheet fields. **Note:** Not all the transform functions available in Flow Designer are available.
Activate data record matching	Specify fields where the system should check for matching data and, if found, update existing records instead of creating new ones by enabling the **Match** toggle in your target table. For example, if a target table has an **Order number** field and a record with an order number of 743, there are two options for handling potential matches. ○ If the **Match** toggle switch for the **Order number** field is on, the system checks for matching data. If the source data has a record with the same order number, the system updates the existing record in the target table. ○ If the **Match** toggle switch for the **Order number** field is off, the system doesn't check for matching data. If the source data has a record with the same order number, the system creates a new record in the target table. In this case, the target table will have two records, both with an order number of 743.
Configure reference and choice field behavior	Configure behavior for **Choice** and **Reference** field types. **a.** Click the gear icon (⚙). **b.** In **Field**, select the field column to match the incoming record value against in the target table. **c.** In **If no matching record exists then do the following**, specify what to do if that record isn't found in the target table. ▪ **Create a new choice/record:** Creates the choice or record in the target table's matching field column from the data imported for the field. ▪ **Ignore this field:** Ignores the field in the target table and leaves it null. ▪ **Skip this record:** Skips adding the entire record (row) in the target table. **d.** Click **Done**.
Add new field columns to target table	Add unmapped field columns from your import to your target table or create new fields in your target table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Description
	a. Select **Add fields** above the target table on the right. **b.** Choose which fields to add to your target ta ble by selecting the checkbox on the left or select **All** to choose all of them. **c.** You can also perform the following func tions: ▪ Edit the **Field Label** for the imported field. ▪ Edit field properties for the imported field by selecting the **Edit icon** ( ). ▪ Create a new field by clicking **Create a new field** in the top right corner of the window. **d.** Select **Add fields** to add the selected fields to your target table.

4. Select **Continue** to create the data import mapping and update your target table.

   Note: You will be prompted if there are any unmapped fields remaining in your spreadsheet. You can map those fields or continue without mapping them.

5. **Optional:** Convert any remaining worksheets if your file contained multiple worksheet tabs.

   - **Note:** For each sheet that you want to convert, you will need to select **Convert to table**, and then repeat steps in this procedure or choose a different pathway for the additional worksheet (see Create new table from spreadsheet import or Create new table from extensible table and spreadsheet import for steps).
   - If you do not have multiple worksheets, proceed to the next step.

6. Select **Done** on the summary screen.
   Alternatively, to view your table in Table Builder, you can select **Edit table(s)** on the summary screen.

What to do next

For more information on editing your data tables in Table Builder, see Table Builder.

Use a PDF to create data tables

Create tables from a PDF form using the PDF extractor tool in App Engine Studio.

Note: Starting with the Brazil release, the PDF extractor tool has been deprecated. The feature is hidden and no longer available for installation or use. The capability to create tables from PDFs is available with Build Agent. For more information, see Migrating to Build Agent.

Create a data integration

Add new or updated information to your current data sets by importing it into existing tables in your application through the power of Integration Hub. Schedule future imports to add data at specific times.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

This video shows you how to perform the following procedure.

https://player.vimeo.com/video/982217306?badge=0&autopause=0&player_id=0&app_id=58479

Before you begin

- You must have an existing table in your application. For more information, see Create a blank table.

  Note: Currently, Excel files are the only accepted data format users can upload.

- Role required: admin, sn_app_eng_studio.user, or delegated_developer. For more information, see Delegate developers using AES.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. Next to the Data heading, select the Add icon ( ).
4. Select **Create a data integration**.

Configure your data import source

5. Configure your data import source by performing the following steps.

   a. Fill in the fields for the data integration.

Field	Description
Name	Name of your integration.
Short description	Description of your integration.
Application	Name of the application scope for your integration.

   b. Select **Save & Continue**.

   c. Use the **Source type** menu to select a data source type.

   - To choose an Excel or CSV file, select **AD-HOC** > > **File**, then either **Excel** or **CSV**. In the File upload section, upload an XLSX, XLS, CSV, or ZIP file.
   - To choose a Data Stream action, select **Spokes**. Active Data Streams are organized by spoke and listed alphabetically. In the Connection section, select the connection alias. For Data Stream actions that use a connection alias, you can override the default connection alias with any of its children's aliases. For more information, see Create a Connection and Credential alias .

   d. Configure the data source.

   - To configure an Excel spreadsheet, specify which sheet to use and which row number to use for the header. Only one sheet and one header can be specified.
   - To configure a CSV file, select the delimiter.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- To configure a data stream action, enter any required inputs. This section only appears if the Data Stream action has inputs to enter. Data Stream actions with inputs have unique inputs, so the fields in this section depend on which Data Stream action is selected and its configuration. Any default values already configured by the spoke inputs are pre-filled. You can view the details of the Data Stream action by clicking the information icon () next to it. Clicking the icon opens the Data Stream action's configuration.

6. Select the **Save** button.

Choose your target table and configure import options

7. Select the **Map to target** tab.
8. Select the target table and options for the data integration you're creating.
   Mapping the source data to the target table allows you to control how the system treats the data you uploaded.

   a. On the Source to target table data mapping page, select **Add a table**.

   b. In the Mapping Properties window, fill in the fields.

Field	Description
Target table	Name of the table to transform your data to.
Run table's business rules when importing	Option to run the target table's business rules when you import the data.
Run the import synchronously	Option to run the import synchronously. When data is transformed in parallel, and there are multiple source records with a specific coalesce value, checking this option ensures that only one record with that coalesce value is inserted at a time. For more information, see the Synchronized inserts section on the Concurrent imports page.

   c. Select **Save**.

   You can add multiple target tables to your integration. Repeat steps 4a through 4c for each target table you want to add.

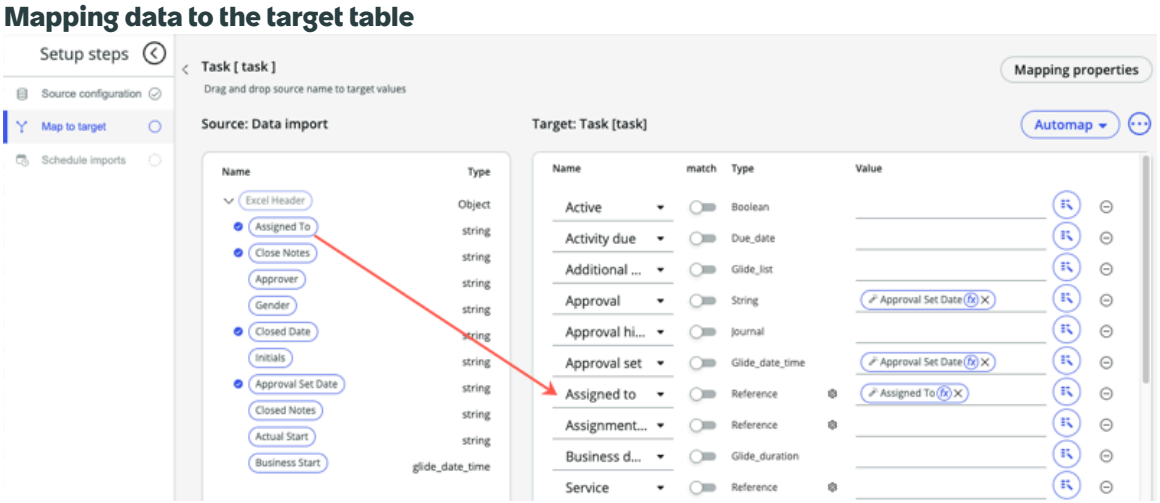
   Tables are listed in the order they're added. You can change the order by using the drag icon to the left of each table card to drag the card to a different position. The order of the table cards determines the order in which data is transformed into the target tables at runtime.

   Note: You can't select the same table more than once, and you can't select an extended table or parent table of a table already in the integration.

Map import fields to your target table

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

9. Map the fields in your import to the target table by performing the following actions.



a. On the Source to target table data mapping page, select the button with the target table you want to map to.
This opens the data-mapping section for the table. The left side of the page shows the source data. The right side is for the target table.

b. Use the **Add a field**, **Add all fields**, or **Automap** button to configure the target side and begin mapping data.

- To add all the fields in the target table, select **Add all fields**. Fields are added alphabetically.
- To add fields individually, select **Add a field** and choose the field name from the list. A field can only be added once.
- To add fields and map them automatically, select **Automap**. Automap maps source entities to similarly named target entities. The **Automap** button has two options.
  - **All matching fields in the target table** adds all the fields and maps the ones that match.
  - **All unmapped fields in the target selection** maps only the fields that you've already added but haven't mapped yet. If you haven't added any fields yet, this option is not available.
- For a target entity that has already been mapped, selecting **Automap** doesn't change its value. An automapped field has a wand icon on its data pill.

  By default, Automap maps a maximum of one source entity to a target entity. You can change this by setting the *glide.ih.automap.extractone* property to false.

  Automap matches source to target entities by looking for column names that are at least an 80% match. You can modify this with the *glide.ih.automap.minimum.score* property. For example, to make matches based on a 60% similarity, you can set *glide.ih.automap.minimum.score* to 60.

c. Map data by dragging data pills from the source to the **Value** column on the target table.

You can also map data by using the pill picker ( ) on the right side of the target table. You can enter more than one value in the Value column, you can also enter text.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

After you map a source field to a target field, a small check mark icon ( ) appears next to the source field name. If you move your cursor to the check mark, the tooltip lists the target fields that source field has been mapped to, up to five fields.

d. **Optional:** Use the **Match** toggle switch to specify fields where the system should check for matching data and, if found, update existing records instead of creating new ones.

(Optional) For example, if a target table has an **Order number** field and a record with an order number of 743, there are two options for handling potential matches.

- If the **Match** toggle switch for the **Order number** field is on, the system checks for matching data. If the source data has a record with the same order number, the system updates the existing record in the target table.
- If the **Match** toggle switch for the **Order number** field is off, the system doesn't check for matching data. If the source data has a record with the same order number, the system creates a new record in the target table. In this case, the target table will have two records, both with an order number of 743.

e. **Optional:** For choice and reference fields, specify the column to map data to.

i. Select the gear icon ( ).

ii. In **Field**, select the column to map to.

iii. In **If no matching record exists then do the following**, specify what to do if that column isn't available.

- **Create a new choice/record**: Creates the choice or record in the target table.
- **Ignore this field**: Ignores only the field in the target table.
- **Skip this record**: Skips the entire record in the target table.

iv. Select **Done**.

f. **Optional:** Change or modify your data by applying transform functions.
For example, you can use the Uppercase transform function to change an input string to all uppercase characters. For more information, see Transform functions .

Note: Not all the transform functions available while creating flows and subflows are available in Integration Hub - Import.

i. Select the transform function icon ( ).

ii. Select a transform function from the menu, then select **OK**.

iii. To apply additional transform functions, select **Add a new transform**. Transform functions are applied in the order they're selected.

iv. After you've added all your transform functions, select **Done**.

10. When you're done with your mapping, click **Save**.

Import the file or schedule the data import

11. Choose what to do with your data import you just created and mapped.

- To import the data set you uploaded now, select **Run Import**.
- To set up a future import of the data, select **Schedule an import**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

See Run or schedule a data import.

12. **Optional:** If you ran the import, preview the table and ensure it appears with the additional content by returning to the target table, refreshing your browser screen, and clicking **Preview**.
13. **Optional:** If you added new columns during the import and want them to display on the target table, select **Personalize list** to edit the columns that display.

Modify application data tables

Use Table Builder to modify data tables used in your application.

Role required: admin, sn_app_eng_studio.user, or delegated_developer

To modify data tables associated with your application:

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the **My Apps** page, open your application.
3. Under the Data heading, select the Additional actions icon (•••) for an existing table in your application, and then select **Edit**.

When you click on a data table or select **Edit** in the Additional actions menu (•••) to edit data for your application, Table Builder is launched. Table Builder is a tool for editing data tables that you've added to your application.

You can perform the following actions in Table Builder:

- Manage your application data tables on the **Data** tab.
  - Manage table fields.
  - Edit table and field column properties.
  - View schema relationships between your data tables.

    Note: This feature is only available if your licensing entitles you to "exclusive low code capability" and you have Table Builder for App Engine installed. Contact your Solutions consultant for more information.

- Create and manage form views and display logic that is associated with the table on the **Forms** tab.
- Create and manage flows triggered when the table is updated or created on the **Flows** tab.

  Note: This feature is only available if your licensing entitles you to "exclusive low code capability" and you have Table Builder for App Engine installed. Contact your Solutions consultant for more information.

See Data in Table Builder for more information on editing tables in Table Builder.

Note: To delete a table from your application, select the Additional actions icon (•••) for an existing table in your application, and then select **Delete** and follow the instructions.

Add an application experience

Use App Engine Studio (AES) to add an experience to your app. Experiences are the interface, or wrapper, for how users interact with your app.

Application templates automatically add experiences to your application.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Types of experiences

Experience	Definition	Example	Reference
Standard catalog item	A standard catalog item displays a form that users can fill in to create requests, such as a form to request an item or service. Standard catalog items make use of the Request table and produce a request record. In addition, generated request items are linked with a workflow, a catalog task, and any related approvals. Using standard catalog items, users can add multiple items to a cart and generate a request for each requested item. Use a catalog item when you want the service catalog form to generate a request, complete with a workflow, approvals, tasks, etc.	Create a catalog item form, such as a form to request time-off. After submitting the form, a request is created for a manager to approve or reject.	• Add a standard catalog item • Sample standard catalog item
Record producer	A record producer is a simplified type of catalog item that displays a form for users to fill out. The completed and submitted form inserts a new data record in a selected custom target table that you set up. This type of catalog item enables users to create task-based records, such as an incident record (without any of the more complex functionality that comes with using a standard catalog item, such as creating requests). Use a record producer when	Create a service catalog form that generates a task record, such as an incident, instead of a request.	• Add a record producer • Sample record producer

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Types of experiences (continued)

Experience	Definition	Example	Reference
	you want the service catalog form to create a simplified type of record that isn't a request (for example, a task record such as an incident, change, or enhancement, etc.).		
Workspace	A workspace is a suite of tools that provides agents, case managers, help desk professionals, and managers with tools to help answer customer questions and resolve customer problems.	Create an equipment fulfillment dashboard where employees can request laptops and other items. Workspaces are similar to portals but smaller in scope.	• Add a workspace • Using Workspace Builder • Edit a workspace in Workspace Builder • Configure workspace settings in Workspace Builder • Customize a workspace home page in Workspace Builder • Add a record page for a workspace in Workspace Builder • Create lists for a workspace in Workspace Builder • Enable analytics for a workspace in Workspace Builder • Sample workspaces you can build
Portal	A portal is a site where users inside of your organization can find information, submit requests, and complete business tasks.	Create a self-service portal, or entry point for employees to make requests and browse a knowledgebase, which you can then populate with additional workspaces and catalogs.	• Add a portal • Sample portals you can build

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Types of experiences (continued)

Experience	Definition	Example	Reference
Mobile	A mobile experience enables users to access your application from a ServiceNow native mobile app. You must then populate the app with data.	Create a time-off request application for mobile for managers to approve or reject requests in the Now Mobile app on their mobile device.	• Add a mobile experience • Sample mobile experience

Choosing between record producers and catalog items

Both the record producer and standard catalog item in App Engine Studio (AES) create experiences where users can submit requests through a service portal. However, record producers are more extensible, and enable users to create a record in any table.

Consider the level of customization

Record producers allow for more customization, making the most of the App Engine tools to build experiences from scratch.

Catalog items represent services or items a user can request. The catalog of all items is called the Service Catalog, and is part of Service Catalog Management.

Comparison of record producers and standard catalog items

Functional area	Record producer	Catalog item
Purpose	Create custom fulfillment experiences using App Engine automation tools.	Support simple request management applications for which a predefined approval and qualification process exists, such as requesting IT hardware and software.
Extensibility	Customizable.	Pre-defined.
Example	Password reset form that creates a password reset request.	Form to request a parking permit.
Development model	Process is as follows: **1.** Set up a business-focused data model. **2.** Define the kind of experience your users will leverage. **3.** Define the automation or business logic. **4.** Define the user roles that will use the app.	Catalog items heavily rely on the request fulfillment process for fulfilling items defined in Service Catalog, a central, accurate, and consistent source of data.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Functional area	Record producer	Catalog item
Reporting	Easier to create dashboards with a purpose-built data model.	Requires creating database views and scripting to gather variables into other tables to configure specific reports.
Scalability	Business-focused app makes it easier to find resources and make app changes in a low-code environment.	Typically part of a Service Catalog Management operation with a dedicated team managing each catalog.
Security	Scoped apps allow flexibility to manage permissions without being an administrator.	Catalog items are typically built in a specific scope, and less flexible.

Help topics on record producers and standard catalog items

For more information, see the following topics:

- Record producers:
  - Add a record producer
  - Sample record producer
- Standard catalog items:
  - Add a standard catalog item
  - Sample standard catalog item

Add a record producer

A record producer enables users to create task-based records, such as an incident record, in apps you create in App Engine Studio (AES) without creating a request. It enables a better end-user experience by standardizing requests using a simplified form.

Before you begin

Create a data table that the record producer will add a new record to when a user fills out the form. For more information, Create a data model for your application.

Role required: sn_app_eng_studio.user or delegated_developer. For more information, see Delegate developers using AES.

About this task

A record producer is a simplified type of catalog item that displays a form for users to fill out. The completed and submitted form inserts a new data record in a selected custom target table that you set up. This type of catalog item enables users to create task-based records, such as an incident record (without any of the more complex functionality that comes with using a standard catalog item, such as creating requests). Use a record producer when you want the service catalog form to create a simplified type of record that isn't a request (for example, a task record such as an incident, change, or enhancement, etc.). When you add a record producer, you create a basic version of the record producer. You must then continue editing it in Catalog Builder before it's ready to deploy.

Note: If you created an application using a template, a record producer may already be added to your application.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. In your application, next to Experience, select **Add**.
4. Select **Record Producer** and select **Begin**.
5. On the form, fill in the fields.

Record Producer form

Field	Description
Name	Name for the record producer.
Short description	Brief overview of the record producer.

6. Select **Edit record producer**.
7. In Catalog Builder, specify the required information.
   After completing each step, select **Save**.

Catalog Builder steps

Step	Description
Details	○ Basic info: Record producer name and short description. ○ Item details: Description of the record producer. **Note:** You can attach files such as graphics, images, videos, or a link to an external source of information.
Destination	Table in which the record producer creates records.
Location	○ Catalogs: Name of the catalog the record producer belongs to. ○ Categories: Available categories that can be linked with the record producer. ○ Topics: Available topics that can be connected to the standard catalog item. Find out more about topics in Associate a catalog item with a taxonomy topic in Employee Center.
Questions	Questions for the record producer. From the **Insert new question** list, you can include additional question sets, questions, deactivated questions, single-column container, two-column container, and line break. For information on creating a question and supported question types, see Create

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Step	Description
	a question for a catalog item in Catalog Builder. You can also add dynamic form behavior for a question. For information on adding dynamic form behavior, see Edit a question in Catalog Builder. **Note:** ○ The questions within a question set can't be edited. The question sets can be reordered by dragging and dropping them. ○ Question sets specified in the template can't be removed, while the ones added by the user can be removed. ○ Removal of question sets simply removes the association with the item and doesn't delete the question set. ○ A single-column question set can be added to a single-column or a two-column container. A two-column question set can't be added to a container.
Settings	Options to customize the record producer behavior. ○ Hide 'Add to wishlist' button: Option to hide the **Add to wishlist** button for the record producer. ○ Hide attachment button: Option to hide the **Attachment** button for the record producer ○ Make attachment mandatory: Option to make the **Attachment** field required for the record producer.
Access	Group or individual users for whom the record producer is available or not available.
Review and Submit	Review and submit the record producer.

8. To preview a catalog item in Portal or Now Mobile, select **Preview**.

   Note: When you preview an item, you can interact with it but not submit it.

   a. To preview the item in the portal, select **Portal** for the **View within** field.

      Note: Portal preview is based on the portal URL configured for the Catalog Builder.

   b. To view an item representation in Now Mobile, select **Now Mobile** for the **View within** field.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

c. To open the preview in a new tab, select **Open preview in a new tab**.

Note: When you change the item and save it, the preview is dynamically refreshed in the opened tab.

9. Select **Submit**.

What to do next

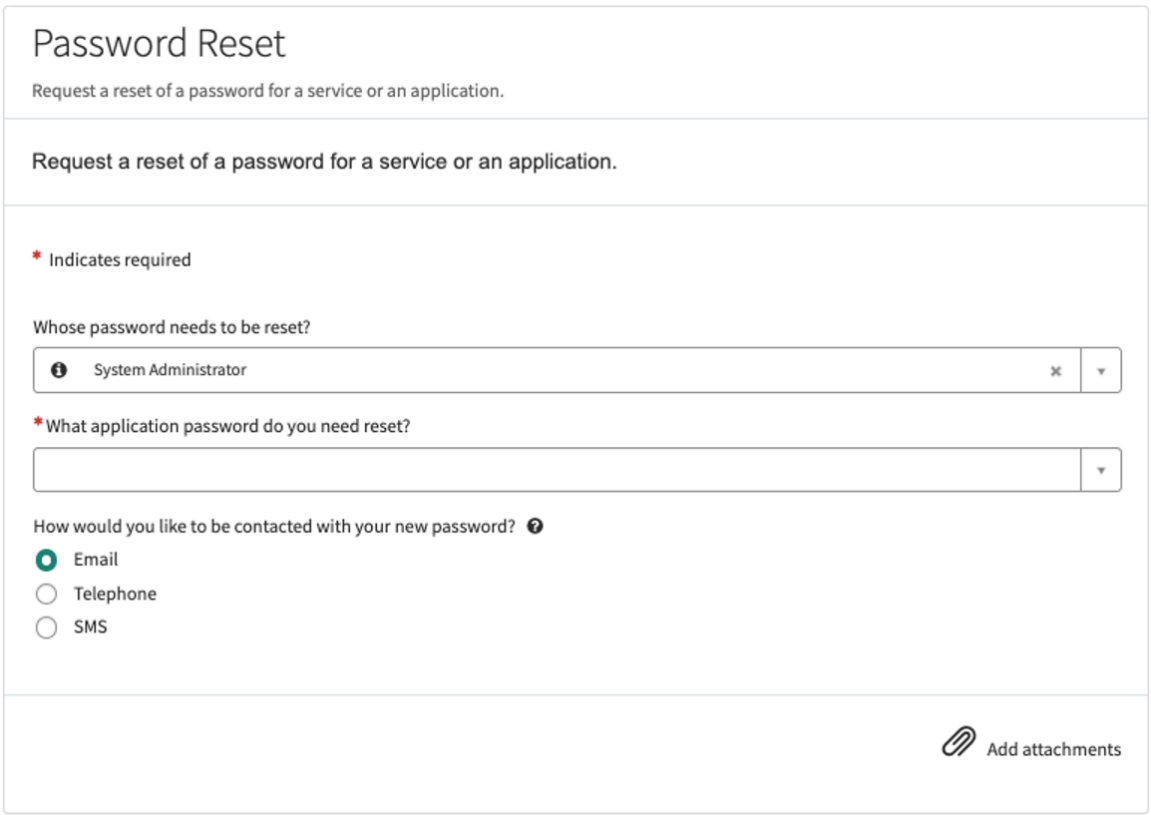
For the record producer you want to edit, select the menu icon (•••) and select **Edit**.

Sample record producer

A record producer experience that you create in App Engine Studio (AES) is a service catalog form that generates a task record, such as an incident, instead of a request.

Example record producer

The following provides an example of a record producer that creates a password reset request.



Add a standard catalog item

Add a standard catalog item to your application in App Engine Studio (AES) so users can submit requests for services and offerings.

Before you begin

Role required: sn_app_eng_studio.user or delegated_developer. For more information, see Delegate developers using AES.

About this task

A standard catalog item displays a form that users can fill in to create requests, such as a form to request an item or service. Standard catalog items make use of the Request table and produce

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a request record. In addition, generated request items are linked with a workflow, a catalog task, and any related approvals. Using standard catalog items, users can add multiple items to a cart and generate a request for each requested item. Use a catalog item when you want the service catalog form to generate a request, complete with a workflow, approvals, tasks, etc. When you add a standard catalog item, you're creating a standard version of the catalog item, which you must then continue editing in Catalog Builder before it's ready to deploy.

Note: If you created an application using a template, a standard catalog item may already be added to your application.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. In your application, next to Experience, select **Add**.
4. Select **Standard catalog item** and select **Begin**.
5. On the form, fill in the fields.

ADD EXPERIENCE form

Field	Description
Name	Name for the standard catalog item.
Short description	Brief overview of the standard catalog item.

6. Select **Edit catalog item**.
7. Specify the required information.

   After completing each step, select **Save**.

Catalog Builder steps

Step	Description
Details	○ Basic info: Standard catalog item name and short description. ○ Item details: Description of the standard catalog item. **Note:** You can attach files such as graphics, images, videos, or a link to an external source of information.
Location	○ Catalogs: Name of the catalog the standard catalog item belongs in. ○ Categories: Available categories that can be linked with the standard catalog item.
Questions	Questions for the standard catalog item. From the **Insert new question** list, you can include additional question sets, questions, deactivated questions, single-column container, two-column container, and line

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Step	Description
	break. For information on creating a question and supported question types, see Create a question for a catalog item in Catalog Builder. You can also add dynamic form behavior for a question. For information on adding dynamic form behavior, see Edit a question in Catalog Builder. **Note:** ○ The questions within a question set can't be edited. The question sets can be reordered by dragging and dropping them. ○ Question sets specified in the template can't be removed, while the ones added by the user can be removed. ○ Removal of question sets simply removes the association with the item and doesn't delete the question set. ○ A single-column question set can be added to a single-column or a two-column container. A two-column question set can't be added to a container.
Settings	Configure the standard catalog item based on the request method. Request method: ○ Order: Use this method in scenarios, such as a corporate laptop request. ○ Request: Use this method in scenarios, such as a code access request. ○ Submit: Use this method in scenarios, such as a reset password request. Other options: ○ Hide 'Add to cart' button: Option to hide the **Add to Cart** button for the standard catalog item. This option is automatically selected if the **Request method** is **Request** or **Submit**. ○ Hide 'Add to wishlist' button: Option to hide the **Add to wishlist** button on standard catalog items. ○ Hide quantity selector: Option to hide the **Quantity** list on standard catalog items. ○ Hide delivery time: Option to hide the **Delivery time** field on the standard catalog

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Step	Description
	items. This option is automatically selected if you set the **Request method** to **Submit**. ○ Hide attachment button: Option to hide the **Attachments** button on standard catalog items. ○ Make attachment mandatory: Option to make the **Attachment** field mandatory on standard catalog items.
Access	Specify the group or individual users for whom the standard catalog item is available or not available.
Fulfillment	Flow with which you can associate the standard catalog item. The flow defines the fulfillment process for the standard catalog item. For more information on flows, see Flows.
Review and Submit	Review and submit the standard catalog item.

8. To preview a catalog item in Portal or Now Mobile, select **Preview**.

   Note: When you preview an item, you can interact with it but not submit it.

   a. To preview the item in the portal, select **Portal** for the **View within** field.

      Note: Portal preview is based on the portal URL configured for the Catalog Builder.

   b. To view an item representation in Now Mobile, select **Now Mobile** for the **View within** field.

   c. To open the preview in a new tab, select **Open preview in a new tab**.

      Note: When you change the item and save it, the preview is dynamically refreshed in the opened tab.

9. Select **Submit**.

What to do next

For the standard catalog item you want to edit, select the menu icon (•••) and select **Edit**.

Sample standard catalog item

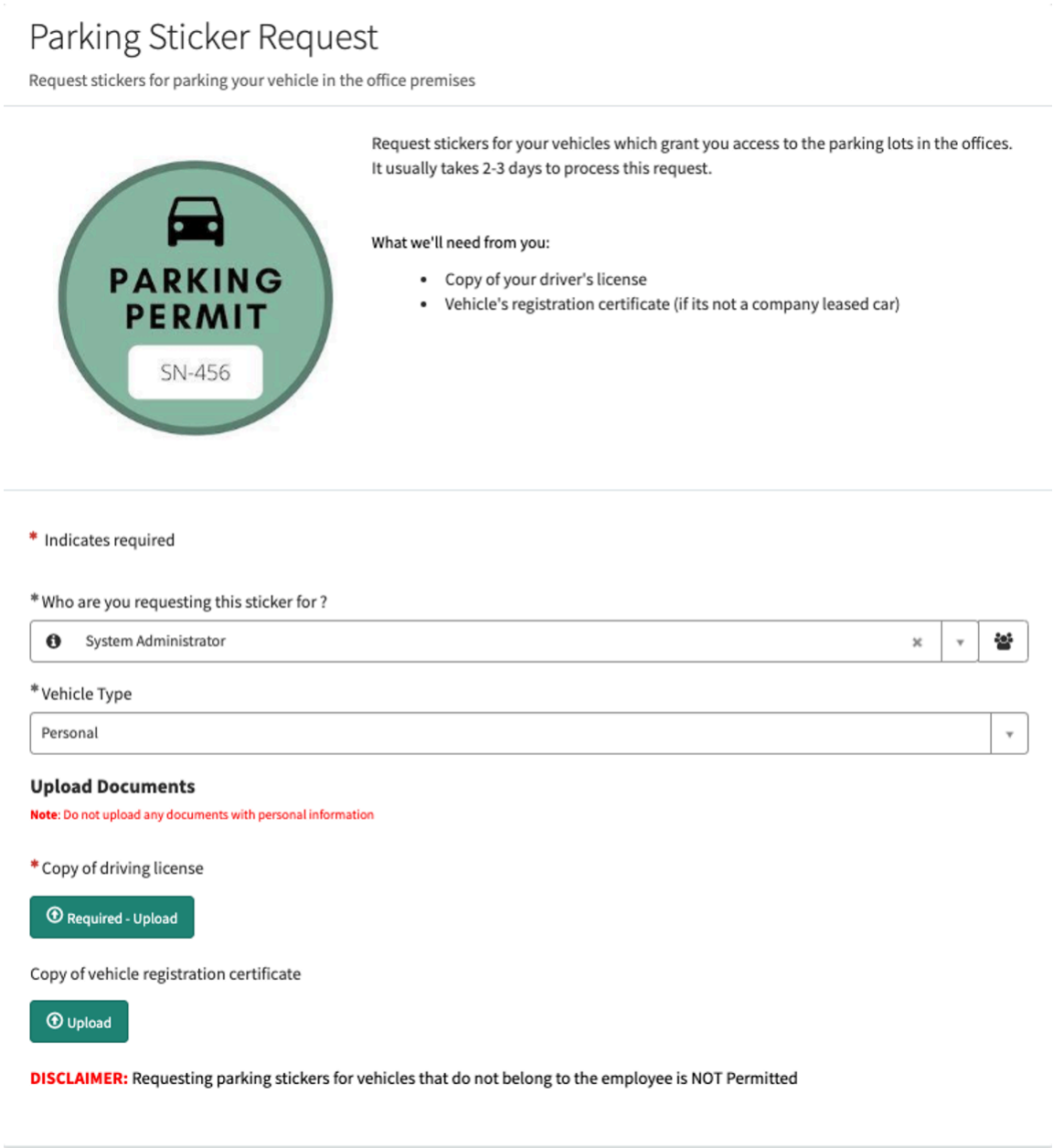
A standard catalog item displays a form that users can fill in to create requests, such as a form to request an item or service. Standard catalog items make use of the Request table and produce a request record. In addition, generated request items are linked with a workflow, a catalog task, and any related approvals. Using standard catalog items, users can add multiple items to a cart and generate a request for each requested item. Use a catalog item when you want the service catalog form to generate a request, complete with a workflow, approvals, tasks, etc.

For example, an employee can fill in a catalog item form to request time-off for a holiday. After submitting the form, a request is created for a manager to approve or reject.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example standard catalog item

The following provides an example of a standard catalog item to request a parking permit.



Add a workspace

Create a workspace using Workspace Builder in App Engine Studio (AES). Workspaces provide agents, case managers, help desk professionals, and managers with tools to help answer customer questions and resolve customer problems.

Before you begin

Role required: sn_app_eng_studio.user or delegated_developer. For more information, see Delegate developers using AES.

About this task

Workspaces are primarily used for request and fulfillment processes, such as a service desk to manage tickets.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

When you add a workspace you're creating a basic version of it, which you must then continue editing in Workspace Builder before it's ready to deploy. For more complex customizations, edit the workspace in UI Builder.

If you create a workspace from an App Engine Studio workspace experience template, you can edit the home page, lists, and record pages in Workspace Builder. However, if you create the workspace using an application template, you must edit some of its pages and objects in UI Builder.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. In your application, next to Experience, select **Add**.
4. Select **Workspace**, and then select **Begin**.
5. On the form, fill in the fields.

Field	Description
Name	Name of the workspace that appears to users. By default, the workspace shares the same name as your application. **Tip:** Use a name that uniquely identifies the workspace from other experiences. For example, `Office art requests workspace`.
Description	Description of the workspace.
URL	Unique web address of the workspace. By default, the workspace URL is based on the application name, but you can edit it.
Roles	User roles to limit who can access the workspace. ○ You must specify at least one role to grant access to the workspace. ○ To use a custom role for your workspace, you must create one in **Security** first. For more information, see Add application security.

6. Select **Continue** to define data for your workspace.
7. On the form, fill in the fields.

Field	Description
Primary Table	Data table that workspace users view or update.
Secondary Tables	Additional data tables that workspace users view and update.

Note: You can select only tables with default views. You can select from two sections: Tables that are already available in your app, and all tables outside the app scope.

Workspace Builder automatically generates a record page for the table you select.

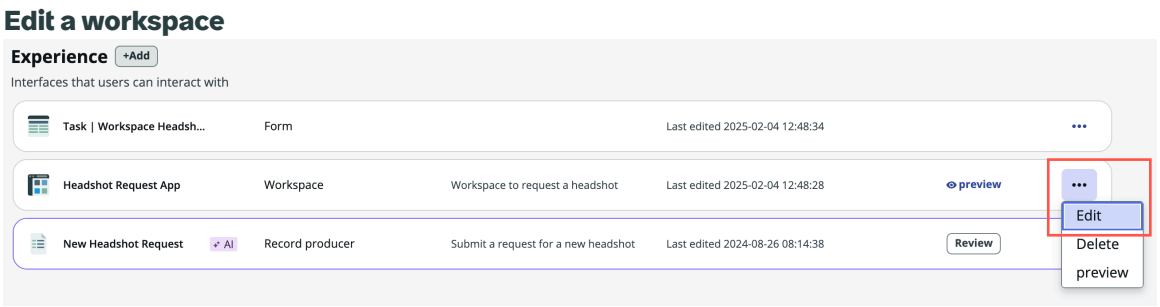
8. Select **Continue**, and then select *Done*.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

After you create the workspace in AES, you must edit it in Workspace Builder by adding or configuring components on each page.

9. Next to the workspace you created, select the additional actions icon (•••) and then select **Edit**.

   You can also select the bar that lists the workspace in the Experience section of your app.



10. **Optional:** Edit your workspace home page in the In-line editor by selecting **Preview**.

    a. On your workspace home page, select **Edit**.
    If you don't see **Edit**, you may not have the permissions needed to edit the workspace home page.

    b. If the workspace is missing an Access Control List (ACL), enter the roles that should have access in the Add user roles to continue modal and select the **Add roles** button.
    For more information, see Configure workspace settings in Workspace Builder and Access Control Lists (ACLs).

    c. Adjust the home page elements and widgets by resizing, reorganizing, or reconfiguring them.

    Editing your workspace home page works similarly to editing a dashboard on your workspace. For more information, see Edit Platform Analytics dashboards.

    Note: If the workspace was created before Tokyo, you must edit it in UI Builder. See UI Builder for more information.

    If you created the workspace that contains a technical dashboard, Workspace Builder prompts you to **Open in UI Builder** when you try to edit the workspace.

    d. Select **Add new element** to add items to your workspace home page.

    e. Select **Exit Editing Mode** to stop editing in the In-line editor.

11. Edit the workspace in Workspace Builder. For more information, see Edit a workspace in Workspace Builder.

    You can also add more record pages to the workspace, if needed.

    Note: The default workspace record pages are read-only and can't be edited. To change the content of each record page, create a page variant and edit the variant.

Add a portal

Create a portal in App Engine Studio (AES) to give your users a site where they can find information, create requests, and complete business tasks.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

Role required: sn_app_eng_studio.user or delegated_developer. For more information, see Delegate developers using AES.

About this task

When you add a portal you're creating a basic version of it, which you must then continue editing in UI Builder before it's ready to deploy.

Note: If you created an application using a template, a portal may already be added to your application.

Most companies implement a maximum of three portals, which they then populate with catalogs.

Procedure

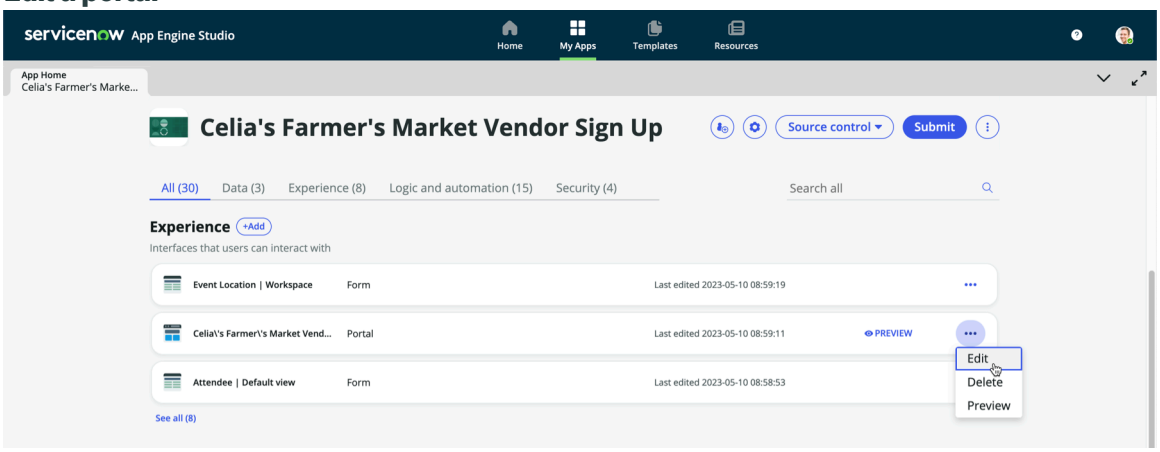
1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. In your application, next to Experience, select **Add**.
4. Select **Portal**, and then select **Begin**.
5. On the form, fill in the fields.

Field	Description
Name	Name of the portal. By default, the portal shares the same name as your application. **Tip:** Use a name that uniquely identifies the portal from other experiences. For example, `Office art portal`.
Description	Description of the portal.
URL	Web address of the portal. By default, the URL is based on the application name.
Roles	User roles to limit who can access the portal. To use a custom role for your portal, you must create one in **Security** first. For more information, see Add application security.

6. Select **Continue**, and then select **Done**.
   After you create the portal in AES, you must edit it in UI Builder by adding or configuring components on each page.
7. Next to the portal you created, select the menu icon (•••) and then select **Edit**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Edit a portal



8. In UI Builder, edit the portal by adding or configuring components on each page. You can also add more pages to the portal, if needed.

> **Note:** The default portal pages are read-only and can't be edited. To change the content of each page, create a page variant and edit the variant.

What to do next

For more information on using UI Builder to edit portals, see UI Builder.

Sample portals you can build

A portal is a site where users inside of your organization can find information, submit requests, and complete business tasks.

The portal that you create in App Engine Studio includes several default pages. You can use these pages as-is or edit them to suit your business needs.

- Article page
- Catalog item page
- Landing page
- Login
- Order success
- Record page(request)
- Record page(task)
- Search page
- Settings page
- Task approved
- Task rejected

> **Note:** Unlike Service Portal, the portal that you can create in App Engine Studio is built on the Next Experience UI Framework. You edit the portal by configuring pages and components in UI Builder.

Add a mobile experience

Add a mobile experience, or interface, to enable users to access your application from a ServiceNow native mobile app.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

Role required: sn_app_eng_studio.user or delegated_developer. For more information, see Delegate developers using AES.

About this task

When you add a mobile experience, you're creating a basic version of the mobile app, which you must then continue editing in Mobile App Builder before it's ready to deploy.

Use Mobile App Builder to build and manage screens and records that make up workflows within the mobile apps that you build with AES. For example, you can add a webpage or a list that appears as a tab on a phone app.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. In your application, next to Experience, select **Add**.
4. Select **Mobile experience**, and then select **Get started**.
5. Select the kind of mobile experience that you want to add and select **Continue.**
   - Mobile Agent
   - Now Mobile

   For more information about the mobile experience options and important considerations when choosing a mobile experience, see Design considerations for mobile apps.
6. On the form, fill in the fields.

Add experience form

Field	Description
Add an icon	Icon that best reflects the content of the screen.
Label for experience	Description for the screen.
Roles	Roles that have access to the screen. You can specify multiple roles, but you must select them one at a time from the list that appears when you start typing in the **Roles** field. **Note:** If you do not specify a role, all users will have access to the mobile experience.
Available offline	Option to make the experience available offline. **Note:** Offline experiences are available only in Mobile Agent apps.

7. Select **Continue**.
8. Select the type of screen to add and select **Continue**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Type of screen options

Option	Description
List	Screen that lists information retrieved from an existing data table, which you choose in the next step. **Note:** Creating a List screen adds all of the data table's values to the mobile experience.
Webpage	Screen that displays an existing webpage, which you specify the URL for in the next step.

9. To add a list page to the mobile experience, do the following:

   a. Select the table that will appear as a list on the screen.
   After you select a data table, some of its contents appear on the page.

   b. To make the table conditional, select **Set conditions for this table** and define the conditions that must be met for the data to appear using the condition builder.

   For more information, see Condition builder.

   c. Select **Continue**.

   d. Specify attributes for the data extracted from the table and select **Continue**.

Identifiers for table data

Field	Description
Add an icon	Icon that reflects the data in the table.
Screen label	Brief label to describe the list screen.

10. To add a webpage to the mobile experience, do the following:

    a. Specify the attributes for the table data and select **Continue**.

Identifiers for table data

Field	Description
Add an icon	Icon that reflects webpage.
Screen label	Brief label to describe the webpage.
URL	Absolute or relative URL for the webpage.

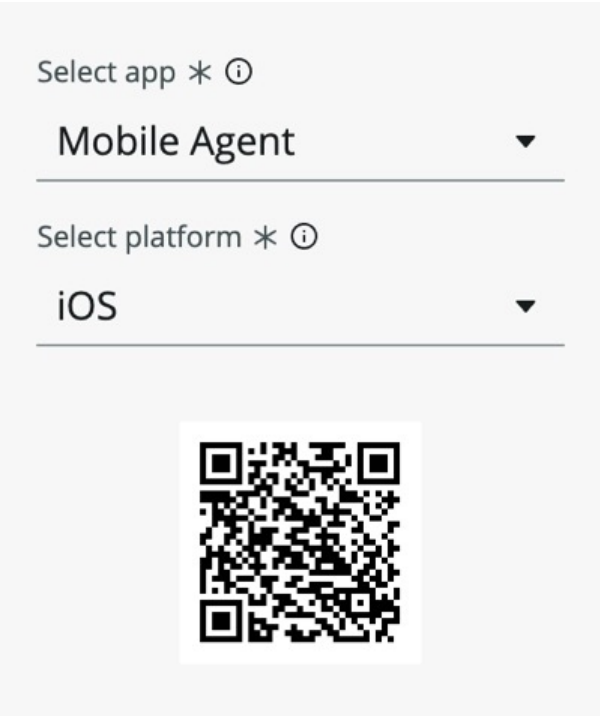
11. Depending on if you are done or not, do one of the following:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- If you are done adding screens to the mobile experience, select **Done adding screens**.
- To add more screens to the mobile experience, select **Add more screens**.

12. To finish creating the mobile experience, select **Done**.
    After you create the mobile experience in AES, you must edit it in Mobile App Builder.
13. Next to the mobile experience you created, select the menu icon (•••) and then select **Edit**.
14. Edit the mobile experience in Mobile App Builder.
    For more information, see Mobile App Builder.

What to do next
Scan or copy the QR code for your mobile experience using the **Select app** and **Select platform** fields.



For more information on modifying mobile experiences, see Mobile App Builder.

Sample mobile experience

A mobile experience enables users to access your application from a ServiceNow native mobile app.

Example mobile experience

The following example illustrates a sample mobile experience that you can build using the Mobile App Builder integration in App Engine Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



For more examples of mobile experiences that you can build, see Native Mobile Examples.

Editing an experience in App Engine Studio

Tailor your app's experience to suit your business needs by building interfaces in App Engine Studio (AES), such as for mobile apps or catalog items.

Editing workspaces in Workspace Builder

Workspace Builder is a streamlined no-code tool that enables you to configure custom workspaces built in AES quickly and easily. For more information on using Workspace Builder, see Using Workspace Builder. For more complex customizations, edit the workspace in UI Builder.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Editing forms created in Table Builder

When you add data to your app, any associated form views display in the **Experience** section of your app. To edit the form, select it in the **Experience** section. Form views can be edited in the **Forms** tab of Table Builder. For more information, see Forms in Table Builder.

Preview an experience

See what an application experience built in App Engine Studio (AES) looks like to your users by previewing it in a browser.

Before you begin

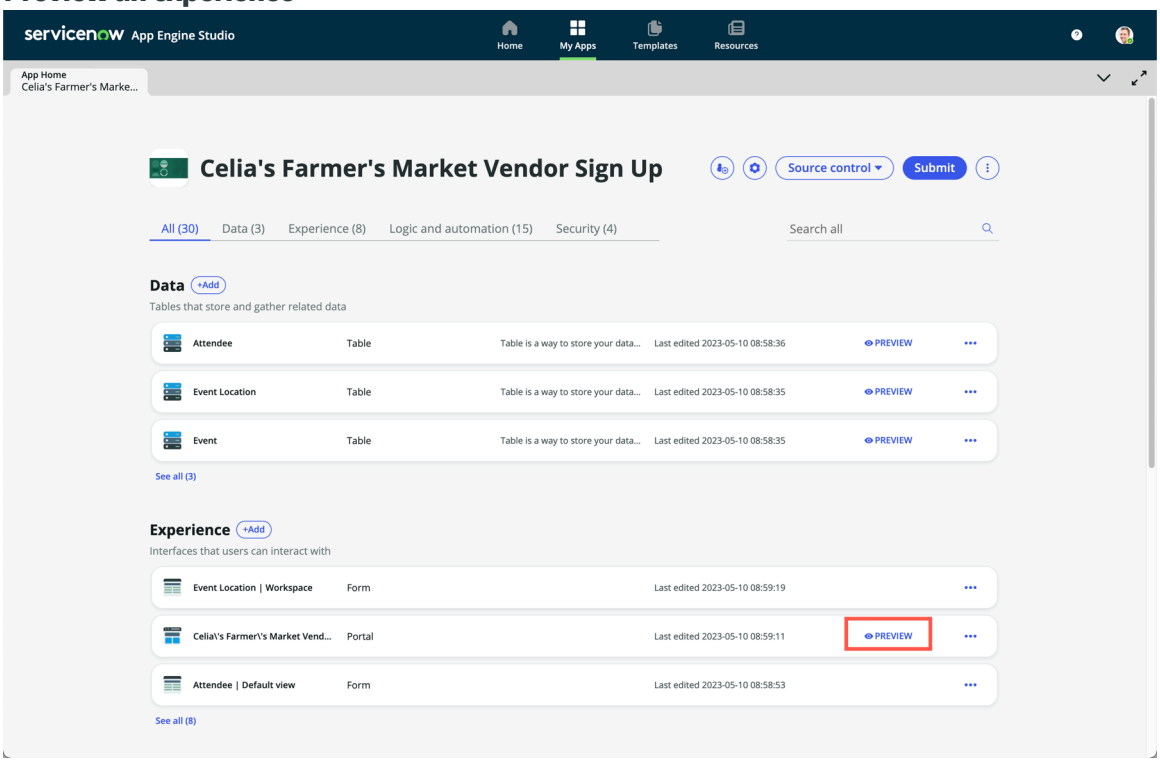
Add an experience to your application. For more information, see Add an application experience.

Role required: sn_app_eng_studio.user or delegated_developer. For more information, see Delegate developers using AES.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. Next to an experience, select **Preview**.

Preview an experience



4. On the browser tab that opens, review the experience.

What to do next

If the experience looks or behaves differently than expected, edit the experience.

Delete an experience

Delete an experience, such as a portal or record producer form, that you no longer need within App Engine Studio (AES).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

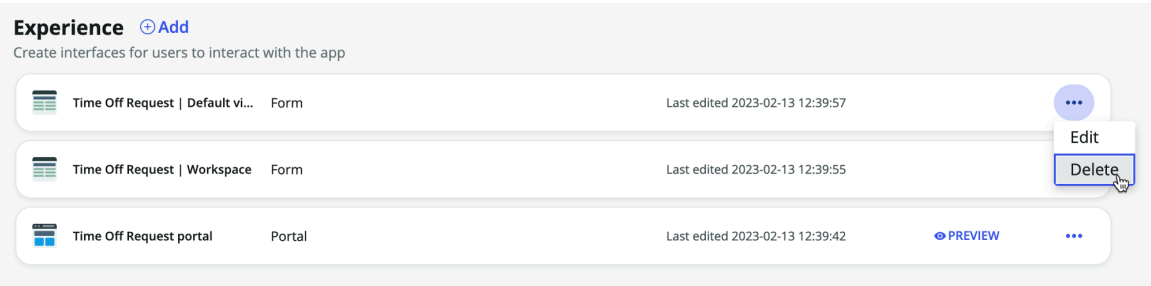
Before you begin

Role required: sn_app_eng_studio.user or delegated_developer. For more information, see Delegate developers using AES.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. Next to an experience, select the menu icon (•••) and then select **Delete**.

Delete an experience



4. On the dialog box that appears, enter `delete` and then select **Delete**.

Add logic and automation

Replace manual work by adding logic and automation to your app in App Engine Studio (AES).

Application templates automatically add logic and automation to your application.

Types of automation

Automation	Definition	Example	Reference
Decision	Decisions decouple logic from code by creating decision rules, such as inputs that evaluate data from existing records, external data sources, or data provided at runtime.	Automatically apply cascading discounts to different products based on the increasing number of units ordered.	• Add a decision • Modify decision table structure in Workflow Studio
Email notification	Notifications are emails that are automatically sent when the specified conditions are met.	Automatically send an email when a new support record is created.	• Add an email notification • Modify an email notification
Flow	A flow is a sequence of reusable actions, initiated by a trigger event, and passing variables between actions.	Automatically generate a Twitter post when an incident is closed or send an email when a record in a specific table is	• Add a flow from scratch • Modify a flow

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Types of automation (continued)

Automation	Definition	Example	Reference
		updated or created (record-based flow).	• Work with record-based flows in Table Builder **Note:** This feature is only available if your licensing entitles you to "exclusive low code capability" and you have Table Builder for App Engine installed. Contact your Solutions consultant for more information.
Process	Processes are cross-enterprise workflows that enable you to create a single, unified process. Processes can include multiple flows.	Standardize and automate how agents handle chat interactions with VIP users.	• Create a playbook • Edit a process

Note: If you're using source control to collaborate with other developers, only changes that have been checked in are available to other developers. For example, if an admin creates a new flow for an app that's linked to Git, the new flow won't be available in the app for other AES users until the admin checks the flow into Git.

Add a pre-built flow using a template

Use a pre-built flow to quickly automate manual work in the app you're building in App Engine Studio (AES).

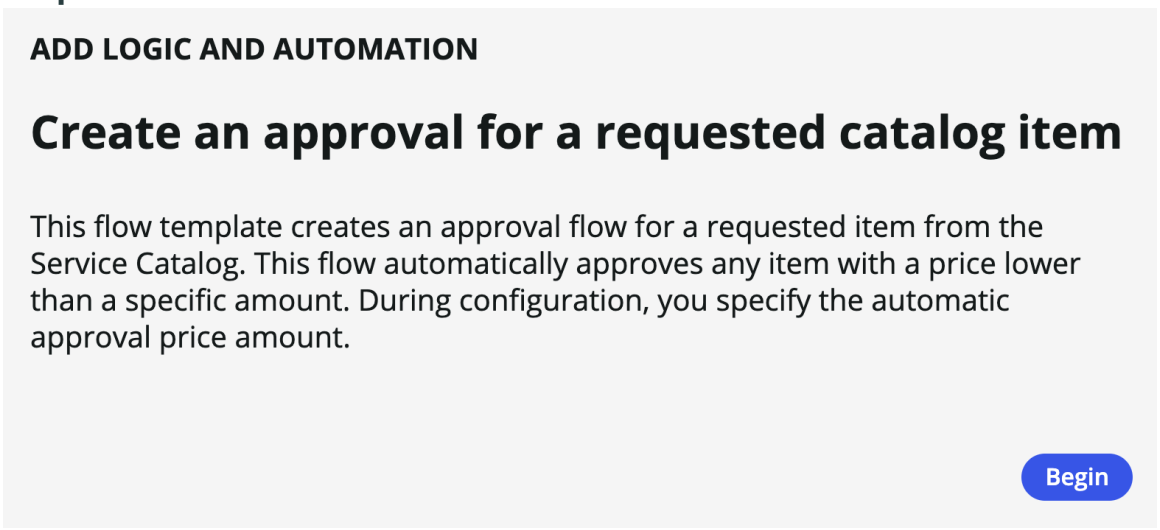
Before you begin

Role required: sn_app_eng_studio.user or delegated_developer. For more information, see Delegate developers using AES.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. In your application, in the Logic and Automation section, select **Add**.
4. Select the card for **Flow**.
5. From the gallery of automation templates, select a template and then select **Begin**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



6. Complete the wizard for the template that you selected.
7. On the summary screen, select **Done**.

What to do next

You can edit the flow to tailor it to your business needs. For more information on working with automation, see Modify a flow.

Add a flow from scratch

If an existing automation template doesn't fit your application goal, define custom automation in App Engine Studio (AES) by building a new flow.

Before you begin

Role required: sn_app_eng_studio.user or delegated_developer. For more information, see Delegate developers using AES.

About this task

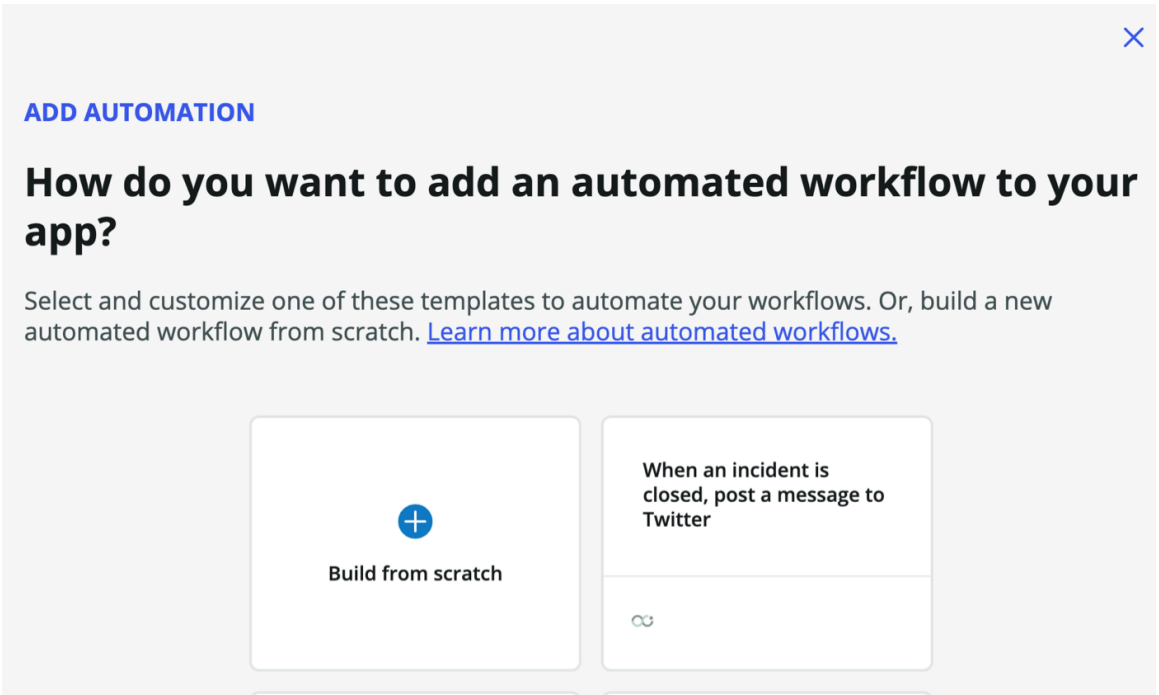
If a pre-built flow already exists, you can easily add its template to your app and then customize it. For more information, see Add a pre-built flow using a template.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. In your application, in the Logic and Automation section, select **Add**.
4. Select the card for **Flow**.
5. Select **Build from scratch**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Build from scratch



6. On the form, fill in the fields and then select **Continue**.

Field	Description
Name	Name to identify your flow.
Description	Description of your flow.
Protection	Access settings for the flow. You can select one of the following options: **-- None --** Option to enable other users to edit the flow. **Read-only** Option to enable other users to view the flow but not edit it.
Run as	Selection to specify if the flow runs as a system user or the user who initiates the session. If updates should come from the user who triggered the flow, select **User who initiates session**. For example, when you want incident record comments to come from the current user, or approval emails to originate from the approver.
Run with roles	Roles that the flow runs with. This option is available only when **Run as** is set to **User who initiates session**. To use a custom role for your flow, you must create one in **Security** first. For more information, see Add application security.

7. On the summary screen, select **Edit this flow** to open your flow.
8. Add a trigger to your flow.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. Under the TRIGGER section, select **Add a trigger**.

b. From the Trigger list, select a trigger that will start running your flow.
For more information on trigger types, see Workflow Studio flow trigger types.
The system displays a set of fields depending on the type of trigger that you've selected.

c. Set up your trigger by filling in the fields.
For a record-based trigger, for example, select a table and set field conditions that, when met, will start running your flow.

d. Click **Done**.

9. To add actions, flows, subflows, or flow logic, select **Add an Action, Flow Logic, or Subflow**.

a. Select an option.

Option	Description
Action	Select the desired action. Workflow Studio includes Workflow Studio actions that are available to flows and subflows. Alternatively, a user with the action_designer role can create additional actions to add to flows. The Integration Hub and Spokes plugins install additional actions. To add draft actions from the More Actions menu, set **Show draft actions** to **true**. To view spokes that are available in the ServiceNow Store, set **Show store spokes** to **true** from the More Actions menu. **Note:** Under **Not Installed Spokes**, the system displays spokes that are available in the ServiceNow Store based on compatibility with the ServiceNow version and application dependency on Workflow Studio.
Flow Logic	Select an option to specify conditional or repeated operations.
Subflow	Select a published subflow and define the input values. In addition to adding a subflow as a flow action, you can enable the **Show triggered flows** option from the More Actions menu to select an activated flow and define the required inputs. Running a triggered flow ignores its trigger conditions and runs all actions.

To change the order of an action in a flow, drag the handle on the left side of the action to the desired location.

The system displays a set of fields depending on the option that you selected.

b. To configure the action, flow logic, or subflow, fill in the fields.

c. Select **Done**.

d. Repeat adding actions until complete.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

10. To specify how the app will handle errors, enable the **ERROR HANDLER** switch and fill in the fields.
    For more information on configuring how errors are handled in flows, see Flow error handler.
11. Select **Save**.
    The ServiceNow AI Platform saves a draft of the flow, trigger, and actions.

What to do next

Test your flow until you're ready to activate it. For more information on testing and editing flows, see Activate a flow.

Note: Your application can trigger only activated flows.

Modify a flow

To update or change an existing flow for an app you built in App Engine Studio (AES), edit it.

Before you begin

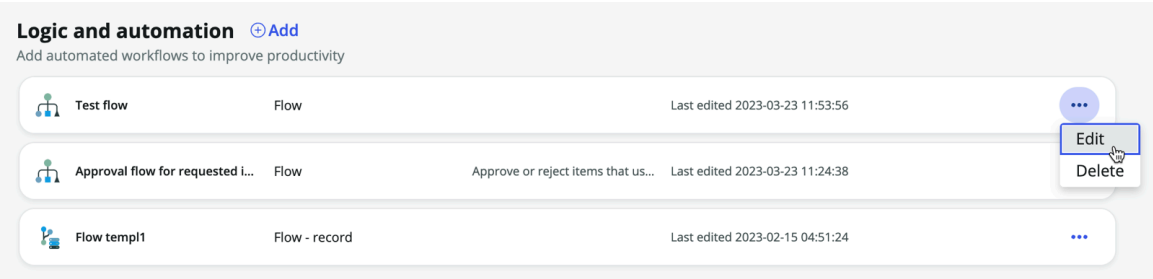
Role required: sn_app_eng_studio.user or delegated_developer. For more information, see Delegate developers using AES.

Add logic and automation to your application. If you created an application using a template, logic and automation may already be added to your application. For more information on adding custom logic and automation, see Add logic and automation.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. Next to a flow, select the menu icon (•••) and then select **Edit**.

Edit a flow



Note: When a flow is triggered by a record being created or updated in a data table, it's referred to as a record-based flow. Record-based flows display in the list of logic and automation with a distinct icon ( ). These flows open up within the **Flows** tab in Table Builder if you are licensed for Table Builder for App Engine and have it installed. Contact your account team for details. For more information on flows functionality in Table Builder, see Flows in Table Builder.

4. Take the appropriate actions to edit the flow.

Option	Description
Change the flow name, description, or roles	In the main header, select **Properties**, enter the values you want into the appropriate fields, and then select **Update**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Description
To edit the trigger	In your flow, select the trigger description, fill in the fields as desired, and then select **Done**. **Note:** Modifying triggers can result in the deletion of referenced action configurations.
To edit an existing action	In your flow, select the action description, fill in the fields as desired, and then select **Done**.
To add a new action	In your flow, select the plus icon in the ACTION section, then proceed as you would for adding an action to a new flow.

For more information on editing flows, see Edit a flow.

5. To save your changes, select **Save**.

What to do next

Test a flow to make certain it works the way you expect. For more information on testing flows, see Test a flow.

Add a decision

Create automation for decisions in App Engine Studio (AES). The automation decouples decision logic from code by enabling you to create and manage decision rules.

Before you begin

Role required: sn_app_eng_studio.user or delegated_developer. For more information, see Delegate developers using AES.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. In your application, in the Logic and Automation section, select **Add**.
4. On the Add logic and automation screen, select **Decision** and then **Get started**.
5. Specify the name and scope of the decision table.

Decision Table form

Field	Description
Name	Name for the decision table.
Accessible From	Scopes that can use the decision table. Available values are as follows: ○ **All application scopes** ○ **This application scope only**

6. Select **Done**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

What to do next

After you create the decision table, you must edit it to specify inputs and triggers. For more information on editing decision tables, see Modify decision table structure in Workflow Studio.

Learn more about decisions in AES	Additional ServiceNow resources
ServiceNow provides several additional resources on adding decisions in App Engine Studio.	Getting started with Decision Builder YouTube video Build my first App Engine Studio application ServiceNow University Introduction to App Engine Studio on-demand course

Edit a decision

To update or change an existing decision for an app built in App Engine Studio (AES), edit it in Workflow Studio.

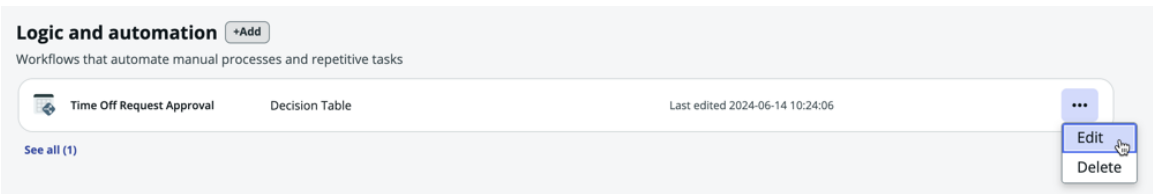
Before you begin

Add logic and automation to your application. If you created an application using a template, logic and automation may already be added to your application. For more information on adding custom logic and automation, see Add logic and automation.

Role required: sn_app_eng_studio.user or delegated_developer. For more information, see Delegate developers using AES.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. Next to a decision, select the menu icon (•••) and then select **Edit**.



4. Take the appropriate actions to edit the decision.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For more information on editing decision tables, see Modify decision table structure in Workflow Studio.

5. To save your changes, select **Save**.

Add an email notification

Build notifications in App Engine Studio (AES) to create automated emails for events using templates or custom messages.

Before you begin

You can perform this task if you're either an administrator or a developer for the application. The required developer permission is **All File Types**. For more information on being a developer for an application, see Delegated development in App Engine Studio.

Role required: admin or delegated_developer

About this task

Email notifications enable you to send emails to the selected users about activities in the system, such as updates to incidents or change requests.

Note: You can't currently create any other types of automatic notifications, such as SMS messages.

Creating an email notification involves specifying when to send it, who receives it, and what it contains.

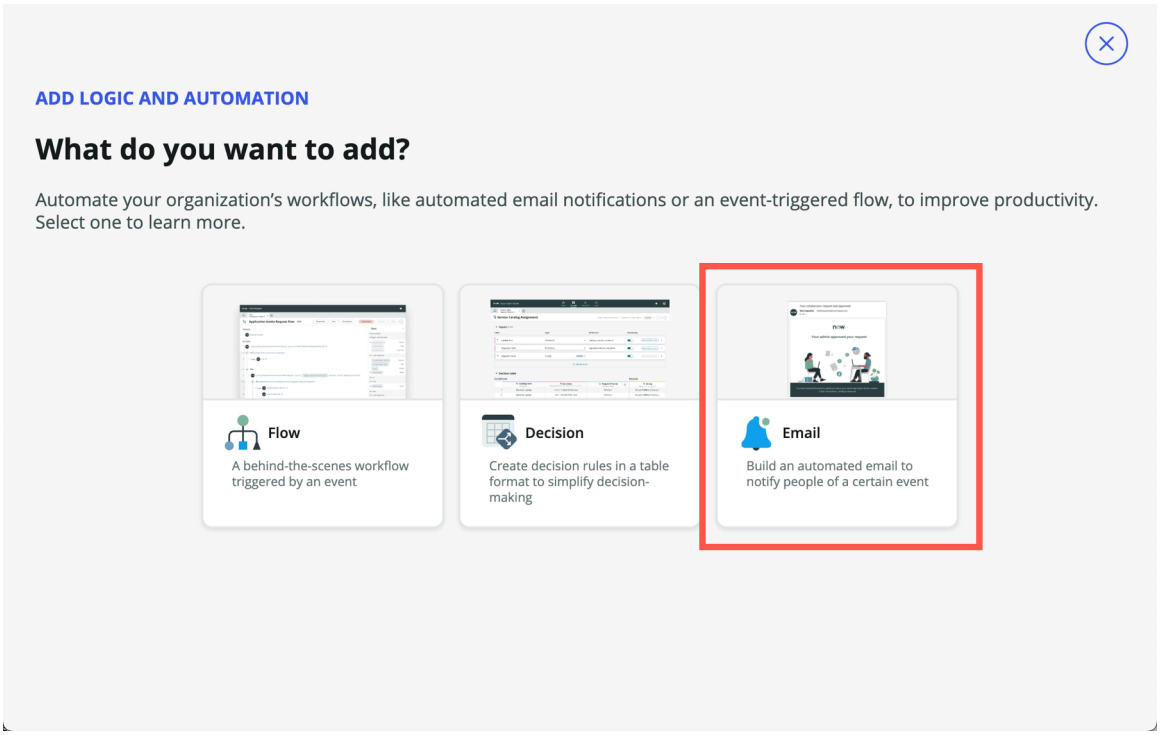
If your administrator has created an email template in the ServiceNow AI Platform, you can use it as is or override its content. For more information on email templates, see Email Templates. Also check with your administrator to see if they are using email layouts to control the header, body, and footer. For more information on email layouts, see Email layouts.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. In your application, in the Logic and Automation section, select **Add**.
4. Select the card for **Email**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add email notification



5. On the form, fill in the fields for the basic settings.

Add notification form basic settings

Field	Description
Name	Name of the email notification. Descriptive names help identify the purpose of the email notification. For example, `Incident Opened & Unassigned`.
Short description	A brief explanation of the purpose of the notification.

6. On the Notification trigger settings section of the form, fill in the fields.

Add notification form Notification trigger settings

Field	Description
Send email notification	Situation where a notification is sent, such as after something happens or when a table is modified. Choices are as follows: ○ **When a record is created or updated** ○ **Only when a record is created** ○ **Only when a record is updated** ○ **When triggered via Flow Action** ○ **When event is fired**

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
	This field is automatically set to **When a record is created or updated**. After you create an email notification with a Flow Action as the trigger, you specify the notification when creating a flow in Workflow Studio. For more information, see Notification step.
Table	The database table that hosts the records to link the notification trigger to. For example, the Incident [incident] table. If you set the **Send email notification** field to one of the following values, the table is required: ○ **When a record is created or updated** ○ **Only when a record is created** ○ **Only when a record is updated** You can select from two sections: Tables that are already available in your app, and all tables outside the app scope.
Event name	Event that triggers the notification. This field appears only when **When event is fired** is selected from the **Send email notification** field.

If the same trigger generates multiple notifications, the system only sends one notification. The system considers all other notifications, even if they have a different subject and body, as duplicates. The Ignore Duplicates business rule controls this functionality.

7. Make the trigger conditional by using the condition builder.

   Define the condition sets and values that must be met for the notification trigger. For more information, see Condition builder.

   Note: If you specify that the notification trigger is a Flow Action, the condition builder isn't available.

8. In the Recipients section of the form, specify who receives the notification.

   You can:

   ○ Search for users or groups.

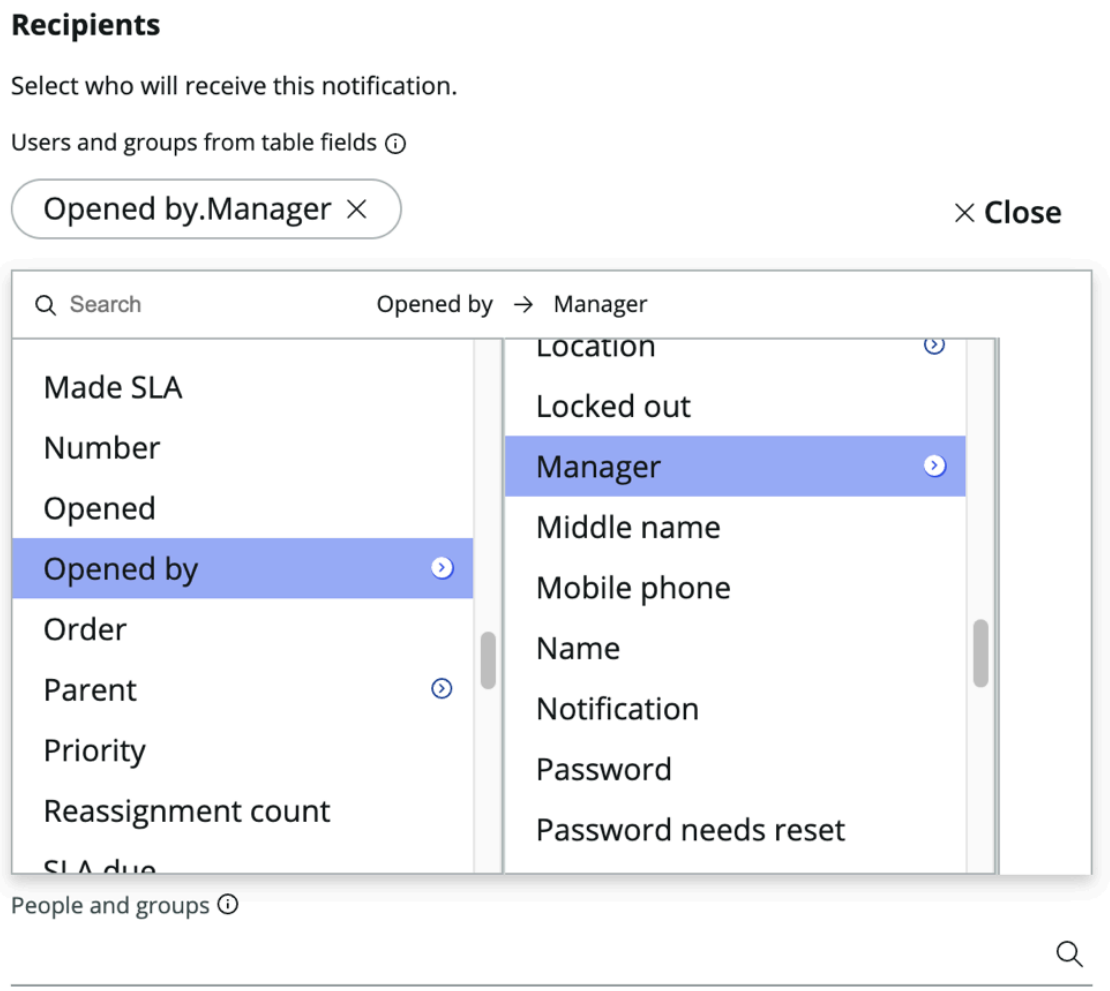
   ○ Define dynamic recipients by selecting **Add**, which requires you to specify a table in the notification trigger details.

Add notification form Recipients settings

Field	Description
Users and groups from table fields	Dynamic users and groups who receive the email based on the table. For example, if

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
	a notification uses the Incident [incident] table, you can select the **Opened by** field to send the notification to users or groups who opened the incident. For some fields, you can also select values contained by a reference field, such as **Opened by** > **Manager**. Select the **Add** button to search for dynamic users and groups based on fields from the specified table. **Note:** If you don't specify a table, the **Add** button isn't available.
People and groups	Static addresses for users and groups who receive the email notification. You can also search for users and groups.



9. In the Email content section of the form, select a template or write original content.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add notification form Email content settings

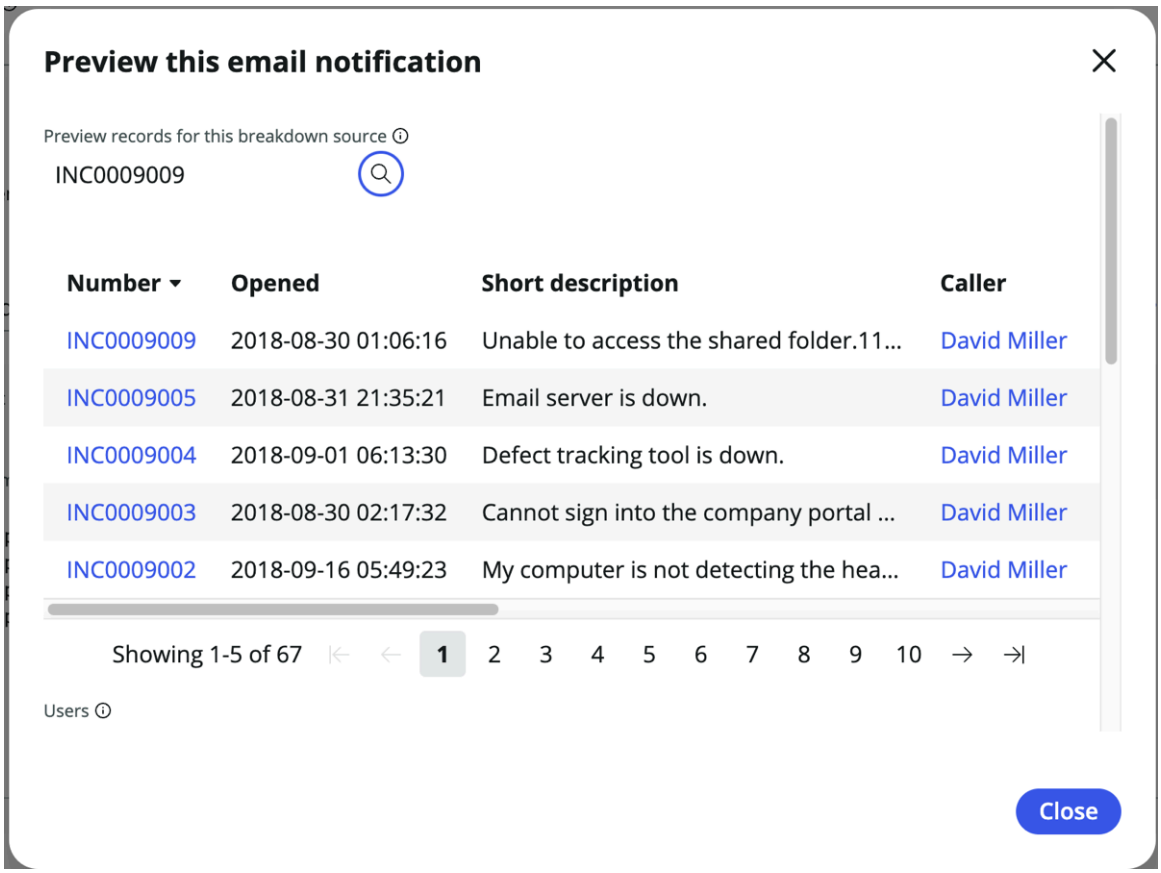
Field	Description
Email template	Predefined email template that you can use to apply existing content, which you can override. You can select only templates that have the following: ○ The same table as the notification ○ No specified table If you use a template and don't override it, then when your administrator updates the template, the content of your email notification is also automatically updated.
Override template	Option to override the contents of the selected email template. If you use an email template, any images from the template still belong to the template regardless of whether you override it. Images in the notification are automatically updated when administrators update the template. If you override an email template and then change the selected template, the system removes the overrides and replaces them with content from the new template.
Subject	Subject line for the email. To make the subject dynamic, use the **Email variables** field next to the **Email message** to generate variable syntax in the message body. You cam then copy and paste the variable syntax in the subject line. If empty, the system uses the **Subject** value from the email template. If you enter a value in this field, it overrides the template.
Email message	Content of the email notification message, which can include variables. You can format the contents of the email message using the rich text editor icons. If empty, the system uses the **Message** value from the email template. If you enter a value in this field, it overrides the template value.
Add variables	Dynamic variables, or values, derived from fields in the specified table. **Note:** You must specify a table for the **Add variables** button to appear.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
	Select variables in the **Email variables** field and use the arrow to move them into the **Email message**. Use variables to include values from a record in the table, such as an incident short description or comments and work notes. Variables are available fields from the specified table.

10. Select **Create**
    The notification is saved, and you can preview it.
11. View how the email notification appears by selecting **Preview**.

    Check the preview that appears. To see how a notification renders, select the search icon in the preview and select the record you want to preview.



Result

When you create a notification in AES, the notification is saved only to the application that you're currently building.

Modify an email notification

Edit an email notification created for an app in App Engine Studio (AES) to change it. For example, modify when to send the email notification, who receives it, and what it contains.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

You can perform this task if you're either an administrator or a developer for the application. The required developer permission is **All File Types**. For more information on being a developer for an application, see Delegated development in App Engine Studio.

Role required: admin or delegated_developer

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. Next to a notification, select the menu icon (•••) and then select **Edit**.
4. Edit the notification by taking the appropriate actions.
   For more information on working with notification fields, see Add an email notification.
5. Change the template you're using.

   a. Select a different template from the **Email template** field.

      If you override an email template and then change the selected template, the system removes the overrides and replaces them with content from the new template.

   b. In the dialog box that appears, select **Change template** to confirm that you want to lose the changes you made by overriding the previous template.

6. To save your changes, select **Save**.

Edit a process

To update or change an existing process for an app built in App Engine Studio (AES), edit it in Playbooks.

Before you begin

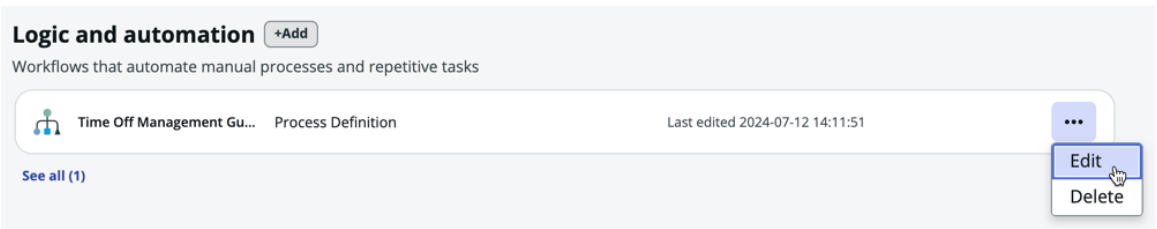
Before you edit a process, you must add a process in Playbooks. For more information, see Create a process definition.

You can edit processes in App Engine Studio, but you must activate Playbooks for App Engine Studio and enable the **Process Automation Designer for App Engine [com.glide.pad.license]** plugin to get started. For more information, see Activate Playbooks.

Role required: sn_app_eng_studio.user or delegated_developer. For more information, see Delegate developers using AES.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. Next to a process, select the menu icon (•••) and then select **Edit**.'



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

4. Take the appropriate actions to edit the process.
5. To save your changes, select **Save**.

What to do next

Test a process to make certain it works the way you expect. For more information on testing processes, see Test a process.

Add application security

Control who is permitted to use or edit your application by configuring roles in App Engine Studio (AES).

Working with roles

You add security to your application by defining roles. A role is used to limit and control access to your application. The access that you define for a role is granted to all users who are assigned to the role.

Note: If you don't have permission to work with roles for an app, the **Add** button for the Security section appears, but you can't select it.

An administrator assigns a role to the appropriate user based on the kind of work that the role permits. For example, to enable a manager to read and approve employee requests, an administrator would assign a role that has access to read and update records from the relevant table. The role may also need access to the portal or workspace where the manager would see the request.

Application templates automatically add security to your application. If you use a template to create your application, you can edit the roles that were added or add different roles.

Learn more about AES roles and security	Additional ServiceNow resources
App Engine Studio includes default roles that you can use for your application. Alternatively, you can build custom roles for your application. After you submit your application, an administrator on-boards the requesting team by assigning the roles that you've defined.	 What is Security? - ServiceNow Developers site  Security and Roles - ServiceNow Developers site  Elevating Privileges - ServiceNow Developers site

Build a new role for your application

Create a custom role for your application in App Engine Studio (AES).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

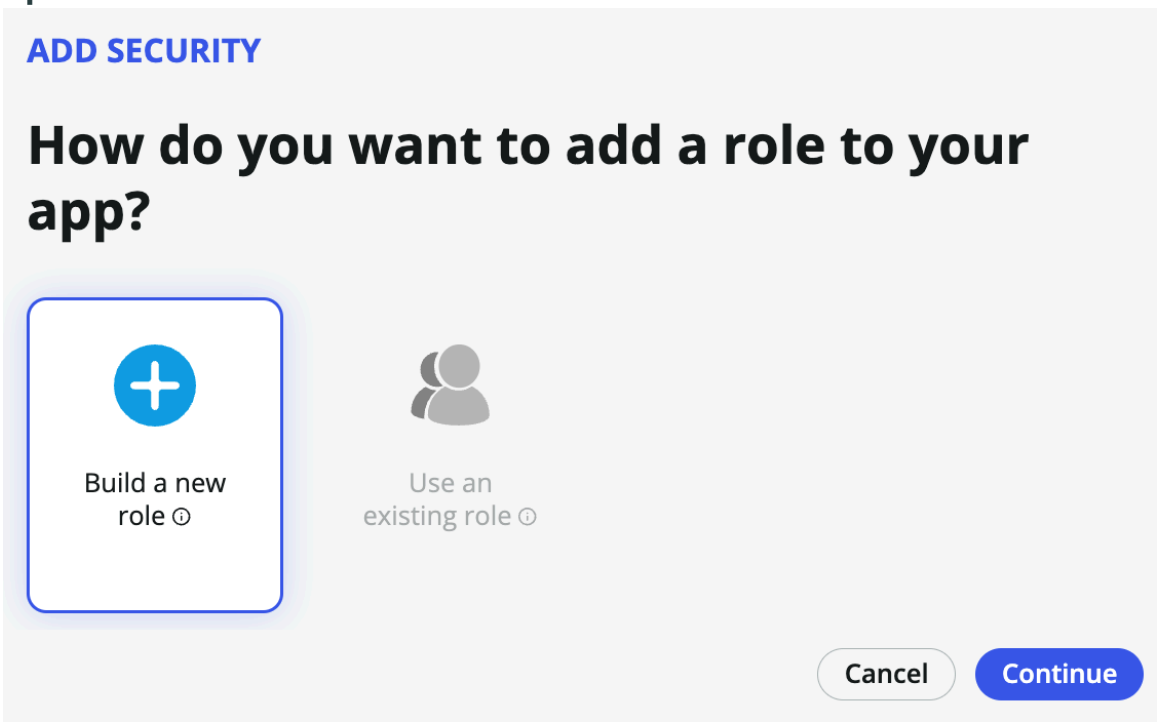
Role required: admin, security admin, or delegated_developer (with the **Security Management** permission)

Note: If you are an admin, be sure to elevate your role.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. In your application, next to Security, select the add icon ( ).
4. Select **Build a new role** and then select **Continue**.

Option to build a new role



5. Enter a name and description for your role, and then select **Continue**.

Tip: For consistency with existing role names, enter a role name that is all lower-case and uses underscores instead of spaces. For example, for an office art requester role, you would enter `office_art_requester`.

6. For each table in your application, select one or more access controls for your role.

Create

Users with the role can create new records on the data table.

Read

Users with the role can read the data that is stored in the table.

Write

Users with the role can update existing records in the table.

Delete

Users with the role can delete table records.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

7. Select the **Experience** tab, and then select the experiences that users with the role can access.
8. Select **Continue**, and then select **Done** on the summary screen.

Use an existing role for your application

Use a previously created role for your application in App Engine Studio (AES).

Before you begin

Your application must contain either data or experiences in order to add an existing role. For more information about adding data or experiences to your application, see the following guidance:

- Create a data model for your application
- Add an application experience

Role required: admin, sn_app_eng_studio.user, or delegated_developer

About this task

App Engine Studio includes default roles that you can use for your application. For more information on the default roles, see Base system roles.

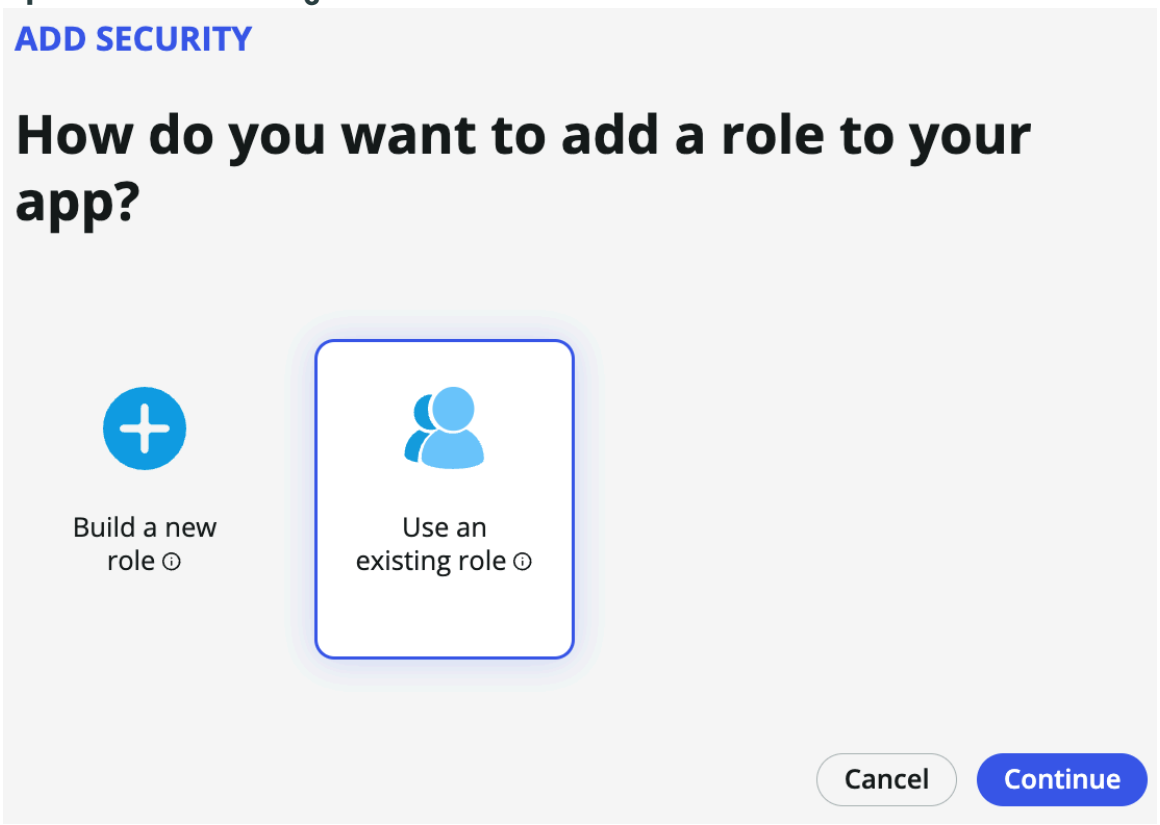
Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. In your application, next to Security, select the add icon (⊕).
4. Select **Use an existing role**, and then select **Continue**.

> **Important:** If your application doesn't contain data or experiences, the **Use an existing role** option will appear, but you cannot select it.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option to use an existing role



5. Select the search field and then select a role from the list.
6. Select **Edit**.
   A new tab opens to set access controls for the existing role.
7. In the Experience section, select the experiences that users with the role can access.
8. In the Data section, set access controls for each available table.

   Create

   Users with the role can create new records on the data table.

   Read

   Users with the role can read the data that is stored in the table.

   Write

   Users with the role can update existing records in the table.

   Delete

   Users with the role can delete table records.
9. Select **Save**.

Change access settings for a role

Control the application permissions for a role in App Engine Studio (AES).

Before you begin

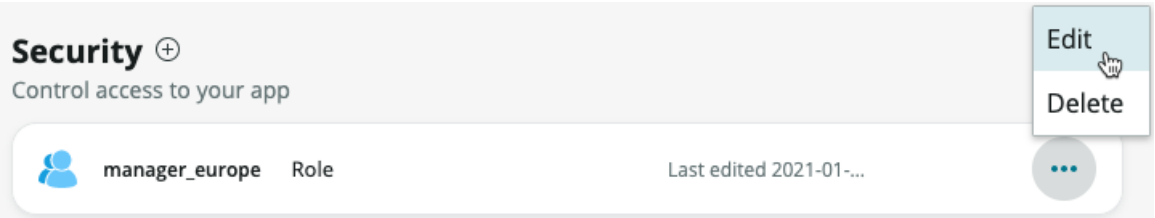
Role required: admin, sn_app_eng_studio.user, or delegated_developer

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. Next to a role, select the menu icon (•••) and then select **Edit**.

Editing a role



A new tab opens to edit the role.

4. In the Experience section, select the experiences that users with the role can access.
5. In the Data section, set the access controls for each available table.

Create

Users with the role can create new records on the data table.

Read

Users with the role can read the data that is stored in the table.

Write

Users with the role can update existing records in the table.

Delete

Users with the role can delete table records.

6. Select **Save**.

Delete a role

Delete a role that you no longer need in App Engine Studio (AES).

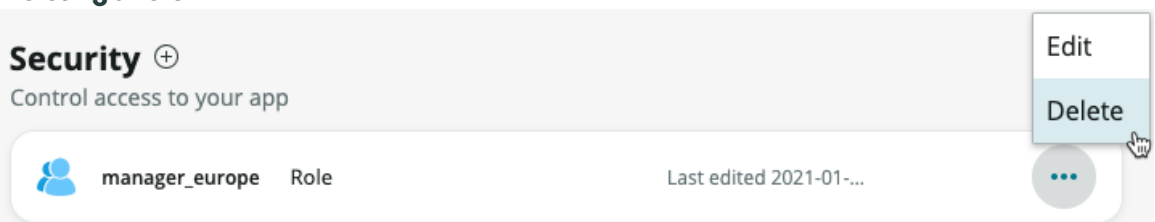
Before you begin

Role required: admin, sn_app_eng_studio.user, or delegated_developer

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. Next to a role, select the menu icon (•••) and then select **Delete**.

Deleting a role



4. On the dialog box that appears, enter `delete` and then select **Delete**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Collaborate with other developers

In App Engine Studio (AES), you can add or remove users and groups to be collaborators on an application.

Users have varying levels of permissions for actions on the Collaboration feature. At a high level:

Users with these permissions	Can perform these actions
Manage Collaborators delegated development permission to the application	Can: • See a list of collaborators and their descriptors • Search for users or groups • Select or change the collaboration descriptor for a user or group (but users can't change customized users or groups) • Add collaborators by sending invitations • Remove users or groups (but users can't remove customized users or groups) Can't: • Customize permissions for a user or group • Deploy via update sets
Invite Collaborators delegated development permission to the application	Can: • See a list of collaborators and their descriptors • Search for users or groups • Add collaborators by sending invitations Can't: • Select or change the collaboration descriptor for a user or group • Remove users or groups • Customize permissions for a user or group
No Collaborator-related roles to the application	Can see a read-only list of collaborators and their descriptors. Can't: • Search for users or groups • Select or change the collaboration descriptor for a user or group • Add collaborators by sending invitations • Remove users or groups • Customize permissions for a user or group

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Users with these permissions	Can perform these actions
Administrators	Admins must elevate to a security_admin role to work with collaboration features. For more information, see Elevated privilege roles. Administrators can: • See a list of collaborators and their descriptors • Show/hide members of a group • Search for users or groups • Select or change the collaboration descriptor for a user or group (includes changing customized users or groups and changing the owner) • Add collaborators by sending invitations • Remove users or groups (includes removing customized users and groups and removing the owner even if there is only one owner) • Customize permissions for a user or group

Note: You should create collaboration descriptors in addition to Owner and Editor in the global scope. If you want collaboration descriptors to appear and be used in AES, you should also set them to `standard = TRUE`. AES doesn't support collaboration descriptors that are created in custom scopes, and non-standard collaboration descriptors don't render in AES.

When you add a user or group, a collaboration task is generated and an approval flow kicks off. To find all collaboration tasks, navigate to **All** > **App Engine** > **Collaboration** > **Collaboration Tasks**. The collaboration task provides information on which application a developer is being added to, and what permissions are granted. Approvers sometimes need to review these task records before they add developers to the application.

If you're an admin, you can modify the Collaboration Request flow. The base system Collaboration Request flow handles collaboration requests as follows:

- If the user has AES or delegated developer permissions and isn't new to the platform, the collaboration request approval record is auto-approved.
- If the user does not have AES or delegated developer permissions and is new to the platform, approval is required.

If you're an admin, you can modify the collaboration descriptors that developers use to assign delegated development permissions. The base table provides Owner and Editor collaboration descriptors. By default, Owners have the manage collaborator delegated development permission set, and Editors have the invite collaborator delegated development permission set.

Note: If you’re using source control to collaborate with other developers, only changes that have been checked in are available to other developers. For example, if an admin creates a new flow for an app that’s linked to Git, the new flow won’t be available in the app for other AES users until the admin checks the flow into Git.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add a user or group to collaboration

Give users or groups permission to collaborate with other developers in App Engine Studio (AES) using the Collaboration feature.

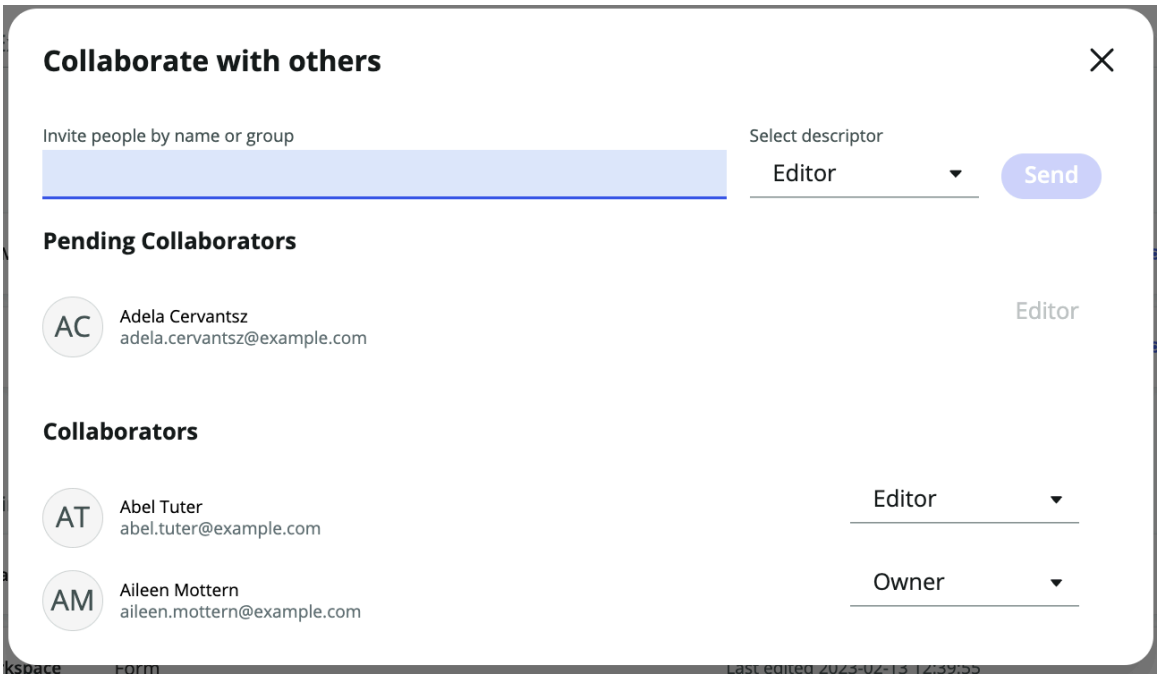
Before you begin

Role required: admin, manage collaborator delegated development role, or invite collaborator delegated development role

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. Select the Manage collaborators button.

Collaborate with others



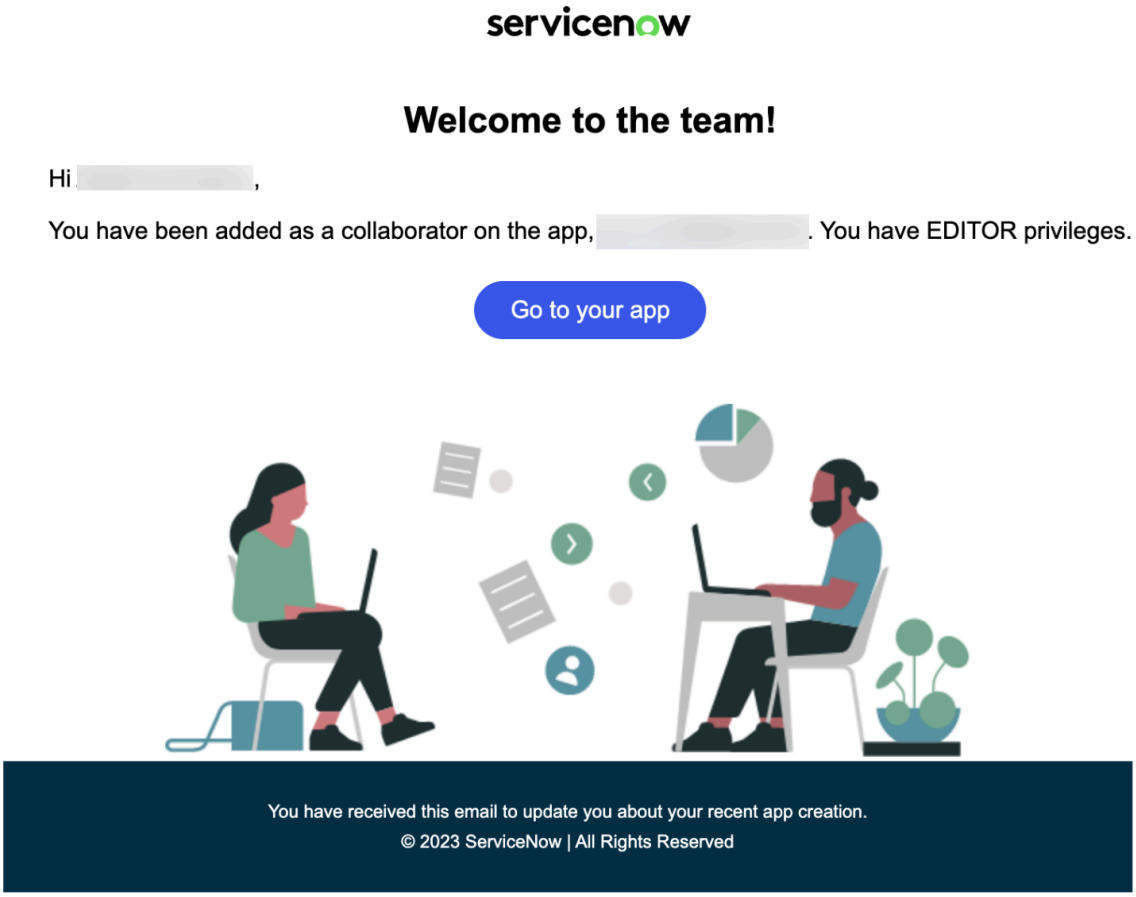
4. To add another user or group as a collaborator, enter the user name or group name in the **Invite people by name or group** field.
5. Search for a specific user or group by entering the first few characters of the name.
   A drop-down list with matching user names and groups appears where you can select the user or group you want to add. If a user or group appears in the drop-down list but you can't select it, it's already been added as a collaborator and can't be re-selected.
6. Select the collaboration descriptor for the user or group that you're adding.
   For a list of collaboration permissions, see Delegated development and collaboration permissions.

   Note: Users with Invite Collaborators permissions can't do this, and will default to the Editor option.

   Only App Collaboration Descriptors that are defined in the global scope and have the standard option selected appear in the list.
7. Select **Send**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- If the user has AES or delegated developer permissions and is new to the platform, the user is listed under the pending requests section and approval is required. After the request is approved, both the requester and the user receive an email indicating that the user has been added to the application.



- If the user has AES or delegated developer permissions and is not new to the platform, the collaboration request is auto-approved. Both the requester and the user receive an email indicating that the user has been added to the application.

Note: If a valid controller has been configured on the instance from which the collaboration request originated, a collaboration request task is created on the controller instance. If the originating instance isn't configured on the controller, work notes are automatically added to the request record indicating that the instance must be configured before approval is granted. If the originating instance is the controller or a valid controller has not been configured, the collaboration request task is created on the requesting instance. For more information, see Configure your controller instance.

Change collaborator permissions

Change the permissions that are assigned to a collaborator or create a custom collaboration permission to enable what people can do in App Engine Studio (AES).

Before you begin

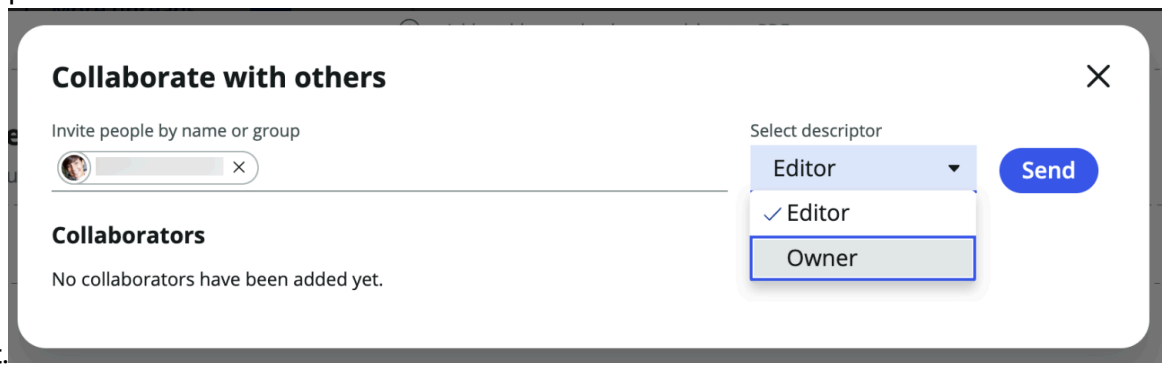
Role required: admin or manage collaborator delegated development role

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

3. Select the Manage collaborators button.
4. For the user or group you want to change, select a different collaborator descriptor from the drop-down list.



Customized users and groups are not editable by non-admins.
5. Admin only: Create custom collaboration permissions for specific users in an application.
   a. In AES, select **Customize Permissions** from the drop-down list across from the user's name.
   b. Select or clear delegated development permissions for the user or group.
   For a list of collaboration permissions, see Delegated development and collaboration permissions.

   If the selection matches that of another standard Collaboration descriptor, the system will provide a prompt before proceeding. For details on each delegated development permission, see Delegate development and deployment permissions to personnel.
6. Select **Save**.

Delegated development and collaboration permissions

Collaboration permissions define what users can do in App Engine Studio (AES).

File type access permissions

File type access permissions grant access to application file types.

File type access permissions

Permission	Description
All file types	Grants access to collaborate on all file types. **Note:** This option includes access to additional file types not listed in separate permissions.
Integrations	Grants access to web service APIs, REST APIs, data sources, and Integration Hub - Import.
Reporting	Grants access to reports and scheduled reports.
Mobile builders	Grants access to build mobile experiences, such as with Mobile App Builder.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

File type access permissions (continued)

Permission	Description
UI Builder	Grants access to work with UI Builder to build more complex interfaces.
Workflow	Grants access to the Workflow Editor and Activity Creator.
Service Portal	Grants access to work with Service Portal editors and tools.
Workflow Studio	Grants access to the Flows design environment in Workflow Studio to create flows and actions. Script action steps require the **Allow Scripting** permission.
Service Catalog	Grants access to work with catalog-related file types such as catalog items, record producers, and variables to add catalog items to apps.
Tables and forms	Grants access to model and layout-related file types such as table columns, form layout, and list layout.
Playbooks	Grants access to work with the Playbooks design environment to create processes. Editing activity subflows or actions requires the **Flow Designer** permission.
Decision Tables	Grants access to work with Decision Tables to create decision logic based on multiple if-then rules.
Notifications	Grants access to create automatic email notifications in apps.

Security/Entitlement permission

The **Manage ACLs and Roles** permission grants access to security management files, such as Access Control Lists and roles.

Programming tools permission

The **Allow scripting** permission grants access to script fields, such as scripting in business rules, UI actions, and client scripts.

Application management permissions

The application management permissions grant access to basic app management functions, like managing collaborators.

Application management permissions

Permission	Description
Delete application	Grants the collaborator within a scoped app rights to delete the application.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Application management permissions (continued)

Permission	Description
Manage collaborators	Grants access to manage and invite collaborators for apps.
Source control	Grants full access to use source control.
Invite collaborators	Grants access to invite developers to collaborate on an app.

Deployment permissions

The deployment permissions grant access to installing, upgrading, and publishing apps.

Deployment permissions

Permission	Description
Upgrade app	Grants access to upgrade the associated application after it has been installed in the current instance.
Submit for deployment	Grants access to submit the associated application for review and deployment.
Publish to app repo	Grants access to publish the associated application to the application repository in the current instance.
Publish to app store	Grants access to publish the associated application to the ServiceNow Store in the current instance.

Remove a user or group from collaboration

Remove users and groups from collaboration to restrict them from owning or editing an app.

Before you begin

Role required: admin or manage collaborator delegated development role

About this task

You can remove the final owner and editor for an app. If an app doesn't have any owners and needs one, an admin must add them.

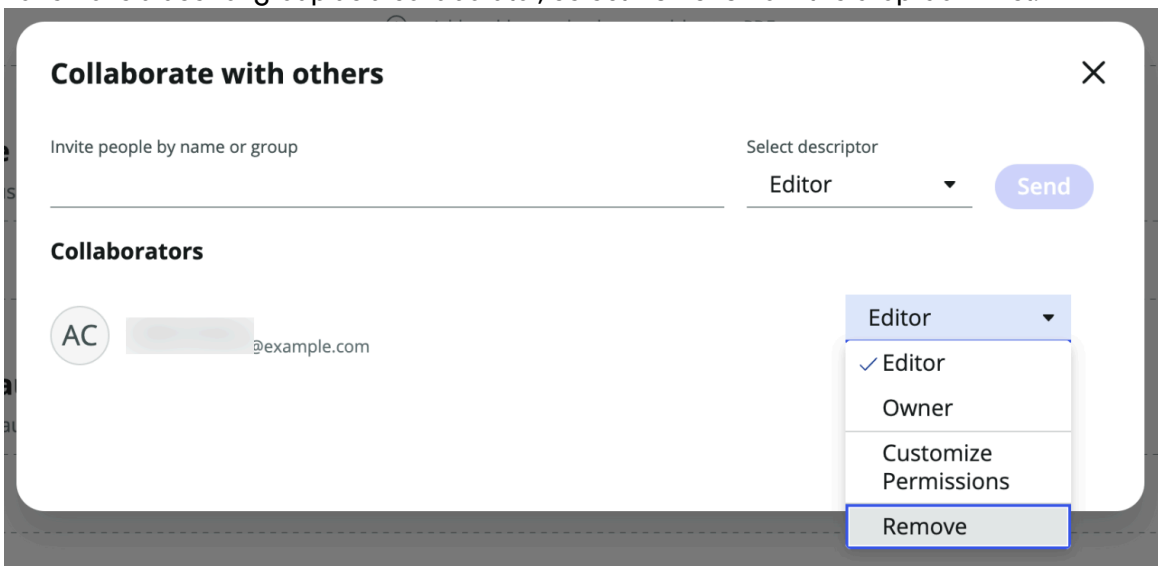
Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. Select the Manage collaborators  button.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The Collaborate with others modal appears with a list of the current collaborators.

4. To remove a user or group as a collaborator, select **Remove** from the drop-down list.



Use AES with a Git source control repository

App developers working in App Engine Studio (AES) can manage their data repository in numerous ways.

After your admin has linked an application to Source control, all application developers on a non-production instance can perform these actions:

- Import applications from a Git repository.
- Pull and apply remote changes from a Git repository.
- Commit all local changes on the instance to a Git repository.
- Create tags to permanently link to a given version of an application.
- Create branches to maintain multiple versions of an application simultaneously.

Note: If you're using source control to collaborate with other developers, only changes that have been checked in are available to other developers. For example, if an admin creates a new flow for an app that's linked to Git, the new flow won't be available in the app for other AES users until the admin checks the flow into Git.

Options available from App Engine Studio

After an application has been linked to source control, application developers can use App Engine Studio to manage the repository. From App Engine Studio, developers can:

- Edit the application repository credentials.
- Commit all local changes on the instance.
- Apply remote changes from the repository.
- Create a branch.
- Switch branches.
- Import an application from a remote repository.

Source control integration does not support managing applications on a production instance. Instead, you can manage applications on a production instance using the application repository,

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

an update set, or App Engine Studio. For more information about managing applications on a production instance, see Application sharing.

Options available from a Git repository

The ServiceNow platform offers limited support for modifying linked application files outside of an instance. From Git, developers can:

- Move application files to a different Git directory structure.
- Edit application files outside of App Engine Studio.

The system generates a properties text file called `sn_source_control.properties` at the root level of the repository. To move application files to a different Git directory structure, application developers can set the *path* parameter to specify the subfolder path containing their application files. For example, if you moved your application to the `src/app` subfolder, set the *path* to `path=src/app`.

The system generates a `checksum.txt` file in the Git repository to determine if any application files have been changed outside of App Engine Studio. When the checksum value from the file matches the current checksum value, the integration skips the validation and sanitization process. When the checksum values do not match, the integration validates and sanitizes the application files as part of the source control operation. The sanitization process:

- Creates upgrade log entries for each sanitization action taken.
- Removes unsupported folders and files from the repository.
- Aborts all source control operations when a system application file fails XML schema validation. For example, if a database dictionary record fails XML schema validation, the system aborts all operations.
- Skips the current source control operation when a non-system application file fails XML schema validation.

The source control integration sanitizes only content within the application path listed in the `sn_source_control.properties` file. Repository content outside the application path is ignored.

MID Server support

Use an existing MID Server to connect to a source control repository. Accessing an application through a MID Server enables access to repositories behind a firewall.

Source control operations in App Engine Studio

The source control integration primarily supports operations from App Engine Studio (AES), but can also support some operations directly from the Git repository.

Available source control operations

Operation	Description	Available from
Import from source control	Imports an application from the repository to the local instance. For more information, see Import application or application-customization from source control.	App Engine Studio
Link to source control	Allows developers to manage application changes from a Git repository. For more information, see Link an application or application-customization to source control.	App Engine Studio

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Available source control operations (continued)

Operation	Description	Available from
Edit Repository Configuration	Updates the Git repository user credentials. For more information, see Edit a Git repository configuration.	App Engine Studio
Apply Remote Changes	Updates the local version of the application to match the repository version. For more information, see Pull changes from a repository.	App Engine Studio
Commit Changes	Updates the repository version of the application to match the local version. For more information, see Commit changes to a repository.	App Engine Studio
Stash Local Changes	Removes and saves local changes for later work. For more information, see Stash local changes.	App Engine Studio
Switch Branch	Updates the local version of the application to match the repository branch version. For more information, see Switch branches.	App Engine Studio
Create Branch	Creates a branch in the repository to save a different version of the application. For more information, see Create a branch.	• App Engine Studio • Git repository
Create Tag	Creates a tag in the repository to link to a particular application version. For more information, see Create a tag to link to a particular application version.	• App Engine Studio • Git repository
Manage Stashes	Allows developers to apply or delete stashed changes. For more information, see Manage stashes.	App Engine Studio
Create repository	Creates a repository to store application changes.	Git repository
Create credentials	Creates credentials to the repository.	Git repository
Grant access to repository	Provides read and write access to the repository tied to a specific set of credentials.	Git repository

Import application or application-customization from source control

Import an app or app customization from a source control repository into App Engine Studio (AES) to continue developing it on this instance.

Before you begin

- Role required: admin
- Verify that the non-production instance has network access to the Git repository.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

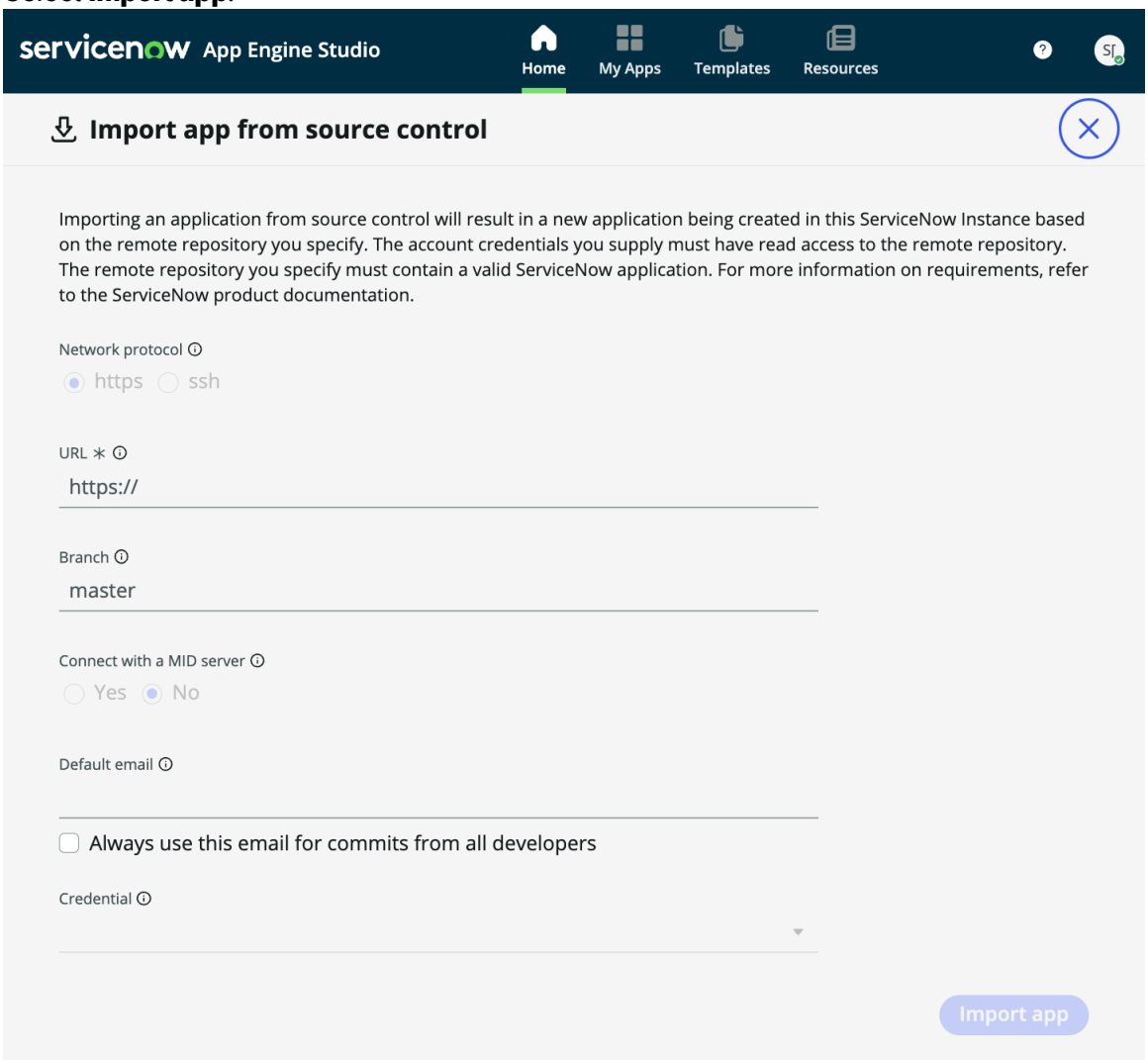
- Verify that the repository contains a valid application.
- Ensure that users add the email address to their respective Users table [sys_user] record that they use in their commits to the Git repository.
- Learn more about application-customizations Manage customizations to applications.

About this task

The source control integration does not support importing an application on a production instance. Instead install applications on a production instance from the application repository, an update set, or the App Engine Studio.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. Select **Import app**.



3. On the form, fill in the fields.

Import from source control fields

Field	Description
Network protocol	Https or ssh credential type that enables secure channel data exchange.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
URL	The URL to the Git repository where the application files reside. **Note:** If the Git repo URL for SSH provided by your Git server does not work, check with your Git server owner or provider for the correct URL. There may be additional specifications such as scheme protocol prefixes, port numbers, and so on, required for your Git repo URL to function.
Branch	The repository branch to work on within the application. **Note:** The default branch is named after your instance. If you do not choose a name, the branch defaults to **master**.
MID Server Name	Select an existing MID Server to link to a Git repository stored behind your corporate firewall. **Note:** Use a separate MID Server to prevent conflicts with Discovery activities. See MID Server for more information.
Default email	The committer email address is defined by the sys_user record if available. But if a committer's sys_user record email field is empty, the system generates an alternate email (username@instancename.service-now.com). You can also enter a default email address and change it later. To use that default email address in all cases, select the check box.
Credential	Select the credential for your Git repository. For more information, see Getting started with Credentials. **Note:** If you select the ssh network protocol, enter a valid credential of the SSH Private Key type. If you select the https protocol, enter a valid credential of the Basic Auth Credentials type.

Note: All application developers on the instance share the credential used to link a Git repository to an application.

4. Select **Import app**.
   The system compares the checksum in the `checksum.txt` file to current checksum. When the checksum values match, the integration skips validation and imports the application. When the checksum values do not match, the integration first validates and sanitizes the application files before importing them.
5. Select **Select Application**.
   App Engine Studio displays the application as a new choice in the Switch Applications window.

What to do next

- Review the upgrade logs for any sanitization applied to application files during the import.
- Select the imported application to edit it.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Related topics

MID Server

Getting started with credentials

Work with changes in Git

Developers using App Engine Studio (AES) can pull and commit changes in their Git repository.

Note: If you're using source control to collaborate with other developers, only changes that have been checked in are available to other developers. For example, if an admin creates a new flow for an app that's linked to Git, the new flow won't be available in the app for other AES users until the admin checks the flow into Git.

Pull changes from a repository

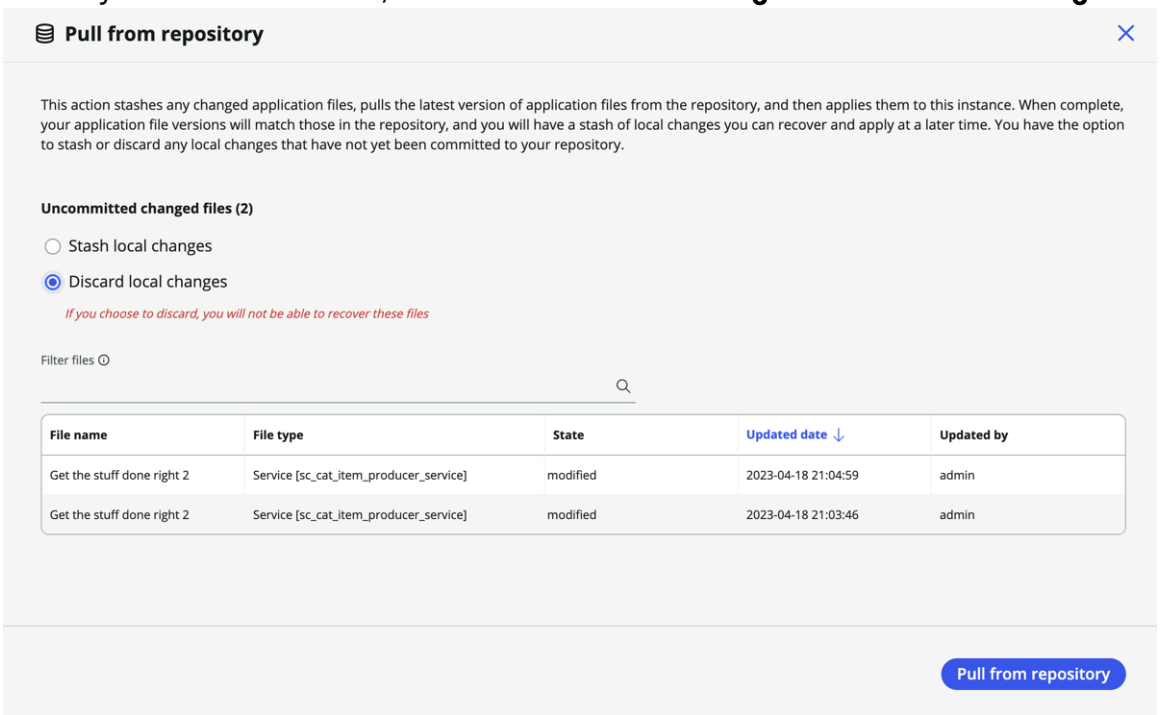
App developers using App Engine Studio (AES) can pull changes from a linked Git repository to apply remote changes to the local instance.

Before you begin

- Role required: admin
- Link an application or application-customization to source control

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. Select **Source control** > **Pull from repository**.
4. Review your uncommitted files, then select **Stash local changes** or **Discard local changes**.



5. Select **Pull from repository**.
   The following operations occur:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- The system fetches the most recent changes from the remote repository.
- The system applies the remote changes to the instance.
- The system identifies any change conflicts requiring resolution.
  If there are conflicts, the system displays the **Resolve Conflicts** window.

Delta loading is enabled by default in sys properties so your data isn't removed. You can disable this feature if you want data automatically deleted.

What to do next
Resolve any change conflicts.

Commit changes to a repository

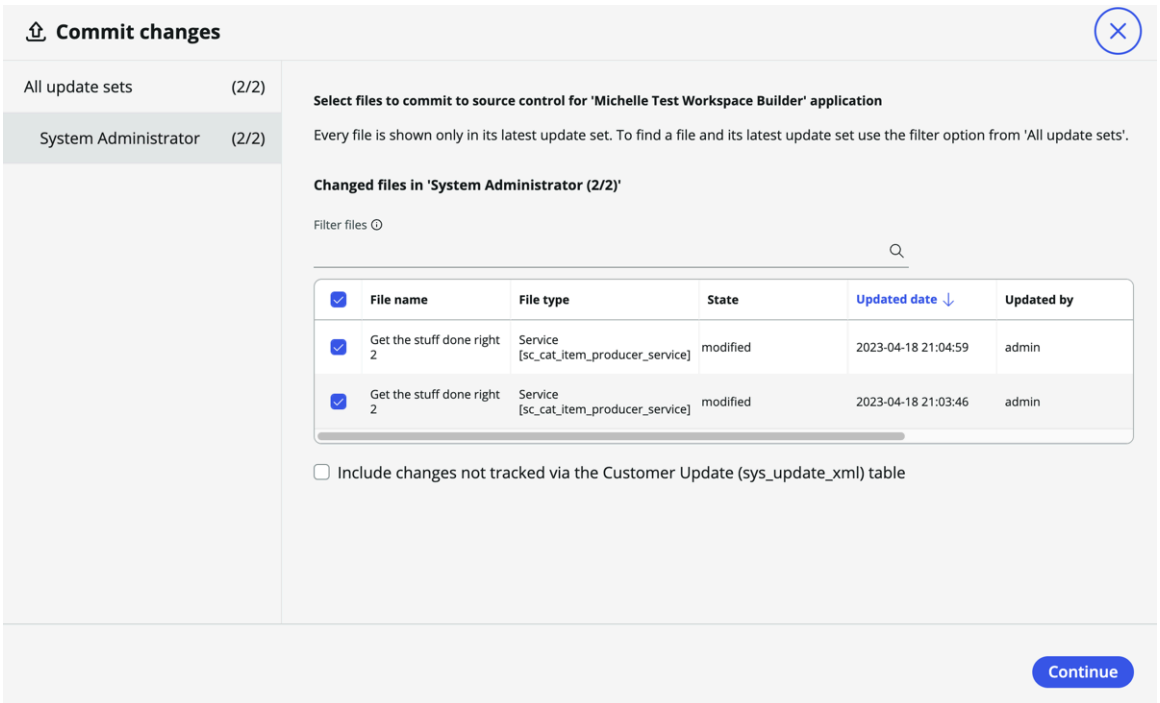
Commit changes made in your application in App Engine Studio (AES) to a linked Git repository. You can either select a few changes to commit, or commit all changes on the instance at once.

Before you begin

- Role required: admin
- Link an application or application-customization to source control

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. Select **Source control** > **Commit changes**.
   The system displays the **Select files to commit to source control** window. The file changes from all the updates sets display. By default, the file changes from the current update set display.



4. Select the file changes you wish to commit.
5. To include untracked changes, select the **Include changes not tracked via the Customer Update [sys_update_xml] table** check box.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- The default for this check box is set via the **glide.sourcecontrol.default_commit_mode** property.
  - Property can be set to **include_untracked** or **exclude_untracked**.
  - The **include_untracked** mode commits the updates to the application that do not generate sys_update_xml records, as well as any user-selected updates.
  - The **exclude_untracked** mode commits only updates selected by the user in the **Select files to commit to source control** dialog.
- The base system setting for the property is **exclude_untracked**.
- To hide the check box and use the value of the **glide.sourcecontrol.default_commit_mode** property, create the **sn_devstudio.vcs.allow_commit_mode_selection** property and set it to false.
- Checking this check box may incur a performance penalty.

Note:

Commits always occur in **include_untracked** mode in the following cases:

- Linking to source control for the first time. (To learn more, see Link an application or application-customization to source control.)
- Publishing an application that's linked to source control from App Engine Studio.
- Selective commit mode is disabled.

6. Select **Continue**.
7. In **Commit comment**, enter a comment for the changes.
8. Select **Commit Files**.
   The following operations occur:
   - The system identifies all local changes.
   - The system commits all local changes to the remote repository.

Note: For list of known files that don't have customer update records and are untracked, see Customer Updates table.

Create versions and branches in Git

App developers using App Engine Studio (AES) can create versions and branches in their Git repositories.

Create a tag to link to a particular application version

Create a tag in the repository to link to a particular app version in App Engine Studio (AES).

Before you begin

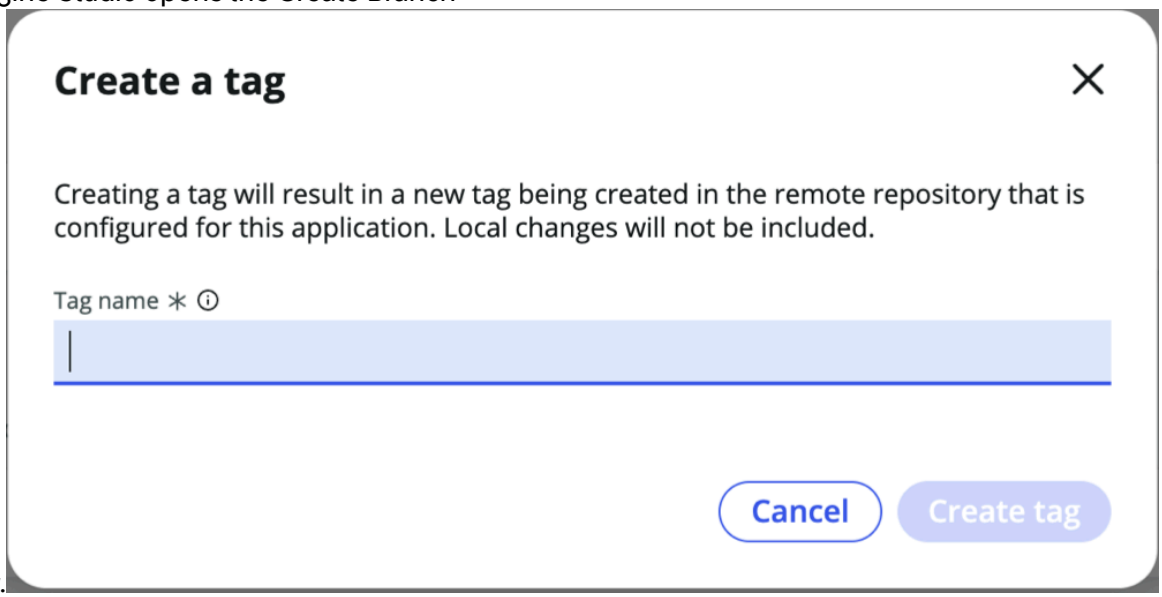
- Role required: admin or sn_group_creator.app_creator
- Link an application or application-customization to source control

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. Select **Source control** > **Create tag**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

App Engine Studio opens the Create Branch window.



4. Enter the **Tag Name**.
5. Select **Create Tag**.
   App Engine Studio creates the tag.
6. Select **Close**.

What to do next

Commit changes to the new branch.

Switch branches

Application developers can switch to a different repository branch to work on another version of the application using App Engine Studio (AES).

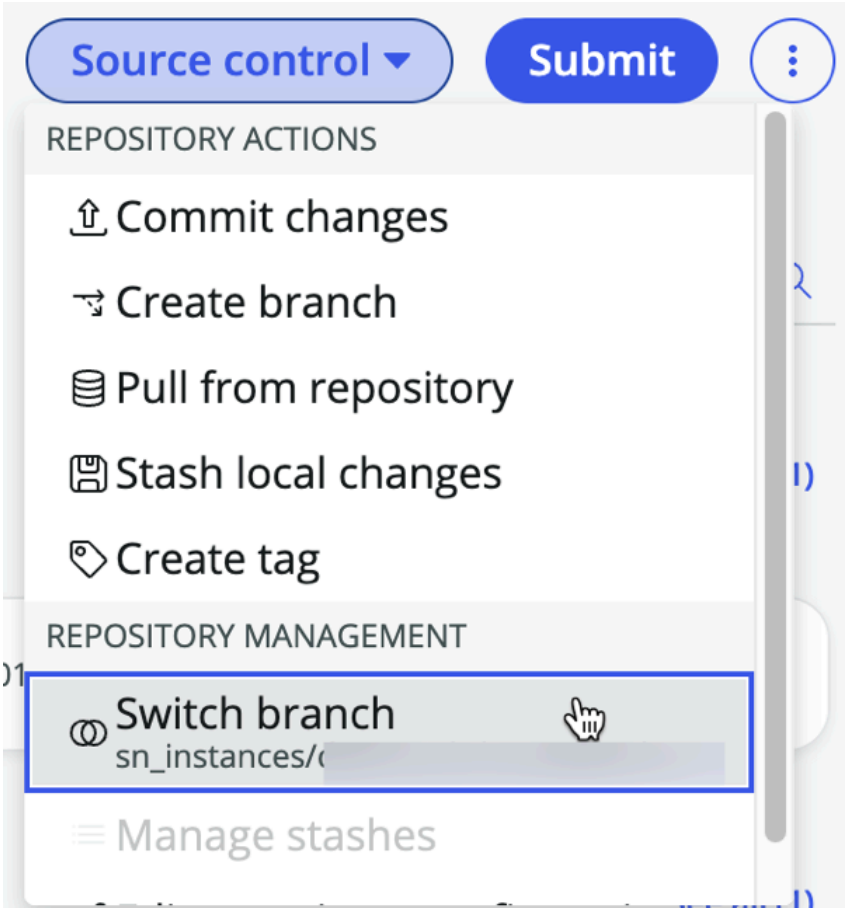
Before you begin

- Role required: admin
- Git repository with one or more available branches.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. Select **Source control** > **Switch branch**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



The system displays the Switch Branch window.

4. **Optional:** If there any local changes on the instance, you can save or discard them.

Option	Description
Save local changes	Saves local changes before switching to an alternate branch. You can later merge or dis card the saved changes.
Discard local changes	Permanently deletes all local changes before switching to an alternate branch.

Note: Use caution when discarding local changes. Since all application developers share repository credentials, there is no way to discard just one set of user changes. Note you cannot later restore discarded changes.

5. Select the branch you want to switch to.
6. Select **Switch Branch**.
   App Engine Studio updates the local application to match the branch version from repository.

Create a branch

Application developers can create a branch to work on a new version of an existing app in App Engine Studio (AES).

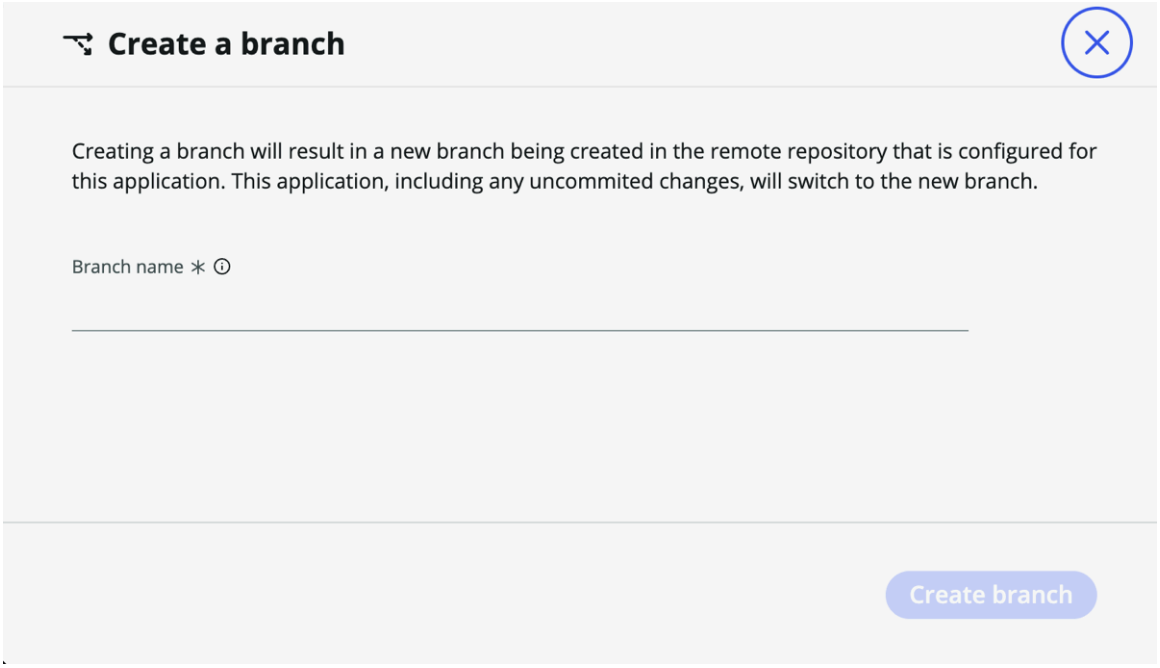
Before you begin

- Role required: admin or sn_group_creator.app_creator
- Link an application or application-customization to source control

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

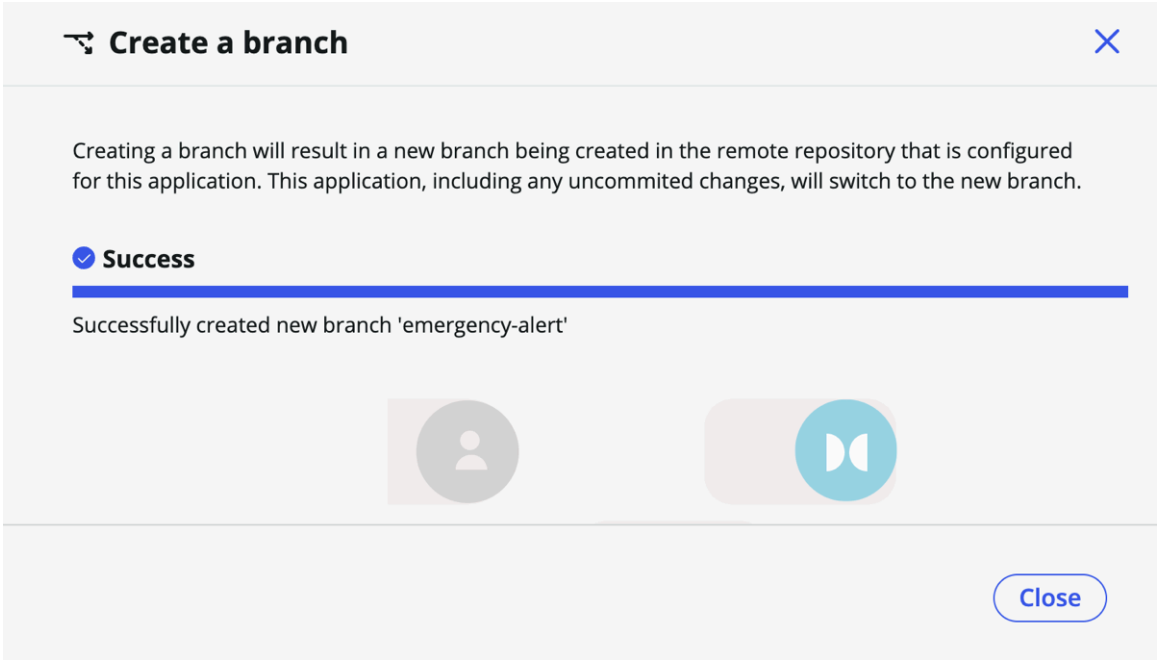
Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. Select **Source control** > **Create branch**.
   App Engine Studio opens the Create Branch window.



4. Enter the **Branch Name**.
5. **Optional:** To create a branch from a tag, select the **Create from Tag** drop-down list and select an existing tag.
6. Select **Create Branch**.
   App Engine Studio creates the branch.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



7. Select **Close**.

What to do next

Commit changes to the new branch.

Set the default branch

Set a default branch when you want to use a branch other than main for new changes or for your main App Engine Studio (AES) development repository.

Before you begin

- Role required: admin
- Link an application or application-customization to source control

Procedure

1. Follow the steps to Add a system property.
2. Add the **glide.source_control.default_branch_name** property, and specify the default branch name of the Git source control repository to work from (pull requests, code commits, etc.). Application developers' work is managed from and saved into the default branch if not otherwise specified. If not changed, this value defaults to sn_instances/<instance_name>.

Stash changes

Developers can remove and save changes locally to apply them later, and manage stashed changes from App Engine Studio (AES).

Stash local changes

App developers can remove and save changes locally to apply them later in App Engine Studio (AES).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

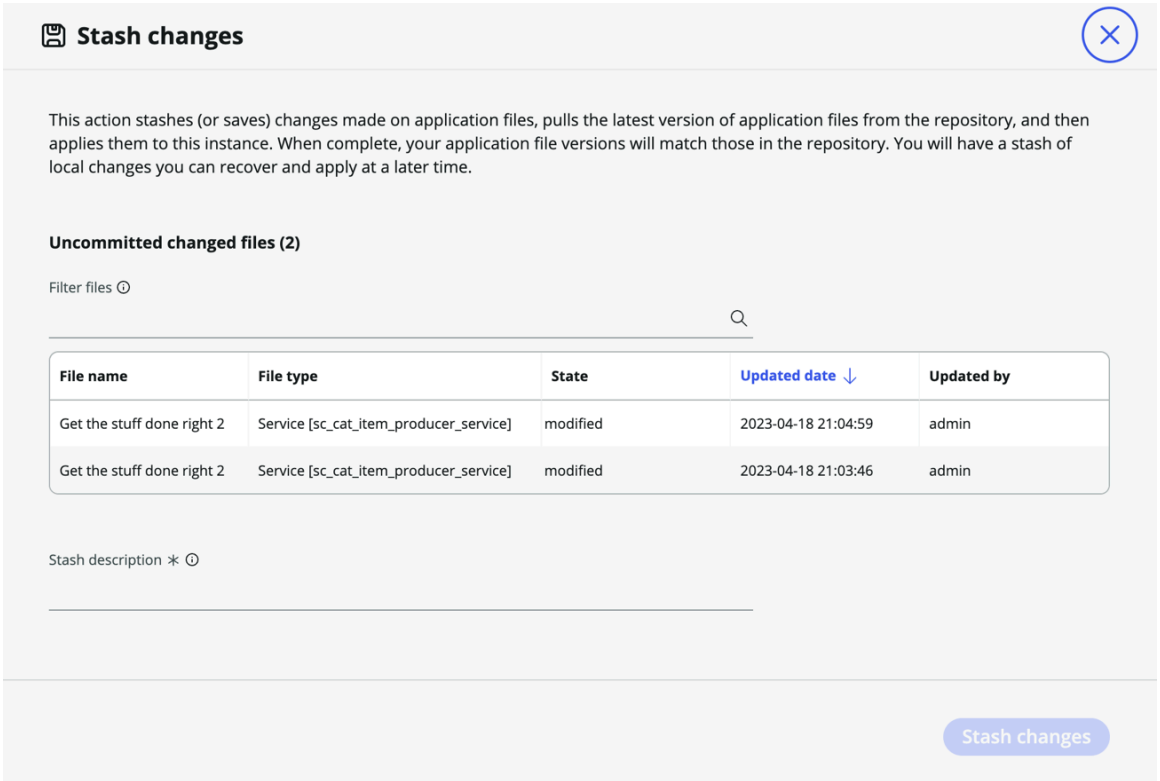
- Role required: admin
- Link an application to source control
- Change one or more application files

About this task

Stashing changes removes them from the current application and saves them for a developer to later apply or delete.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. Select **Source control** > **Stash local changes**.
   The system displays a list of locally changed files.
4. Enter your description.
5. Select **Stash Changes**.
   The system saves the current changes and displays a success message.



What to do next

- Close dialog
- Manage stashes

To learn more, see Getting started with credentials

Manage stashes

App developers can apply or delete stashed changes from App Engine Studio (AES).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

- Role required: admin
- Link an application to source control.
- Stash one or more application file changes.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. Select **Source control** > **Manage stashes**.
   The system displays a list of locally stashed changes.
4. Select the action next to the stash you want to manage.

Option	Description
Apply	Commits the stashed changes to the application and checks for conflicts.
Delete	Removes the stashed changes.

Resolve conflicts

App developers can select the app file version to use when applying remote or stashed changes in App Engine Studio (AES).

Before you begin

- Role required: admin
- Link an application to source control
- Apply a stashed change

About this task

Conflicts occur when there are multiple change versions of the same application file: one set of changes in the remote or stashed version and another set of changes in the local version. App Engine Studio requires developers to resolve conflicts before applying remote or stashed changes.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. Pull from a repository or stash local changes.
   If the system identifies a conflict, it displays the Resolve Conflicts dialog.
4. Select how to resolve the conflict.

Option	Description
Select an action	Apply or discard all stashed changes. Go to step 3.
Manually merge changes	Individually select which changes to apply. Go to step 6.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

5. If you want to apply or discard all stashed changes, select an **Action**.

Option	Description
Take Stashed Changes	Applies the application file version from the stashed changes.
Discard Stashed Changes	Applies the application file version from the most recent pull from the repository.

6. Select **Apply Stashed Changes**.
   The system applies the selected changes.
7. Select **Close Dialog**.
8. If you want to merge the conflicting changes, select **Manually Apply**.
   The system displays a list of version differences by field.
9. Select the field values you want the merged application file to have.
10. Select **Save Merge**.
    The system applies the selected changes.

View commit history

App developers can view the commit history of apps linked to a source control repository in App Engine Studio (AES).

Before you begin

- Role required: admin
- An existing link to a Git repository

Procedure

1. Navigate to **All** > **Source Control** > **View History**.
   The system displays the History window.
2. Select the commit sort order type.

Option	Description
Date	Sort by commit date.
Committer	Sort by user name.

3. Select the sort order direction.

Option	Description
Descending	Sort dates from the most recent to oldest date. Sort user names reverse-alphabetically from Z to A.
Ascending	Sort dates from the oldest to most recent date. Sort user names alphabetically from A to Z.

   The system sorts commits by the selected sort order.
4. Select a commit.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The system displays the commit details for the selected commit.

5. Review the commit details.

Commit Details

Field	Description
Committer	The user who committed the change.
Date	The date-time stamp of the commit.
SHA-1	The secure hash value identifying this commit in the repository.
Message	The commit message associated with this commit.
Files	The list of application files changed in this commit.

6. Close the History window.

Move application files in a Git repository

Move application files linked to source control to any folder of the repository when working in App Engine Studio (AES). Allow application developers to store supporting content such as automated tests in the same repository as the applications they support.

Before you begin

- Link an application or application-customization to source control
- Role required: Source control credentials with write access

About this task

Linking an application to source control generates a properties text file called `sn_source_control.properties` at the root level of the repository. The properties file specifies the folder containing the application files. The integration tracks changes to these application files by generating a `checksum.txt` file. When the checksum matches, the integration skips the validation and sanitization process. When the checksum does not match, the integration validates and sanitizes the application files as part of the source control operation. The integration ignores all repository content outside the application path.

Note: You can set system properties **glide.source_control.checksum_required** to enable optional checksum validations and sanitizations and **glide.source_control.checksum_quick_install** to bypass sanitization steps on checksum matches. See Available system properties for more information.

Procedure

1. Login to the source control repository linked to the application.
2. Create the destination folder where you will move the application files.

   Example
   For example, create the folders `src/app`.
3. Move the folder containing your application files to the destination folder.

   Example
   For example, move the folder `demo_my_app` to `src/app`.
4. Navigate to the root level of the repository.
5. Open the `sn_source_control.properties` text file in a text editor.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

6. For the `path=` property, enter the folder path where you moved the application files.

   Example
   For example, enter `path=src/app`.

7. Save the properties file.

What to do next
Login to your instance and perform Source control operations in App Engine Studio from App Engine Studio.

Publish your app

When you have completed your app in App Engine Studio (AES), you need to submit it for approval and publishing. The process you use depends on whether you are linked to source control.

Submit your app for approval and publishing

Start the process of getting the application you built in App Engine Studio (AES) published by submitting it for administrator review.

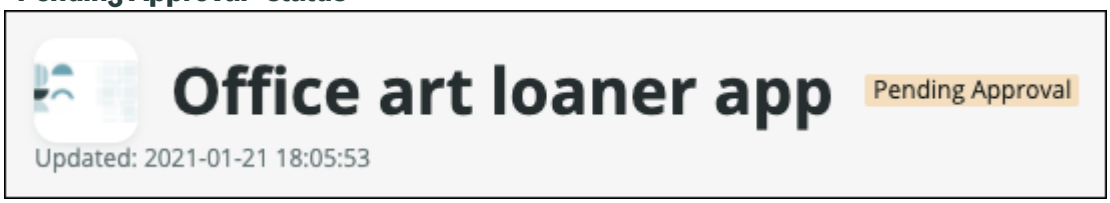
Before you begin
Role required: delegated developer with permissions (**Submit for deployment**, **Publish to app repo**, or **Publish to app store**, depending on the desired action)

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. From the app home, select **Submit**.
4. On the dialog box that appears, review the submission details and then select **Submit**.

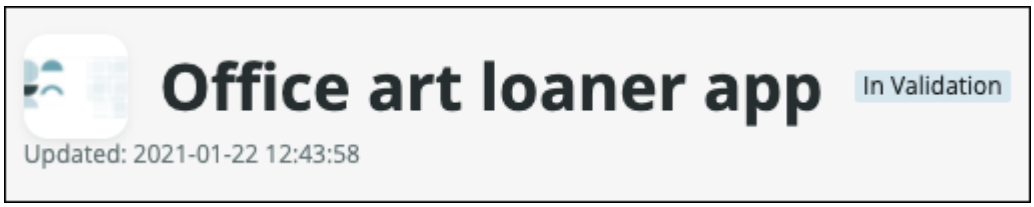
What to do next
The administrator reviews the submitted application and checks for potential issues. You can check the status of your application from the application home in App Engine Studio.

"Pending Approval" status



After a reviewer begins testing the application, the status changes to `In Validation`.

"In Validation" status



An administrator may provide you with test accounts for different roles in your application. Log in with each of these accounts to check that the application works as expected.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

If the reviewer rejects the application, the status changes to `Rejected`. You may also receive an email that includes the reviewer's comments. Use the feedback to improve your application, and then submit the application again.

"Rejected" status



If the application passes testing, the administrator publishes the application and the status changes to `Published`. The administrator on-boards the team that requested the application. For example, if you specified a certain security role for the application, the administrator assigns that role to the relevant users.

"Published" status



Related topics

Delegate developers using AES

Publish an app from App Engine Studio when linked to source control

You can publish a custom application from App Engine Studio (AES) when linked to source control.

Before you begin

Role required: admin

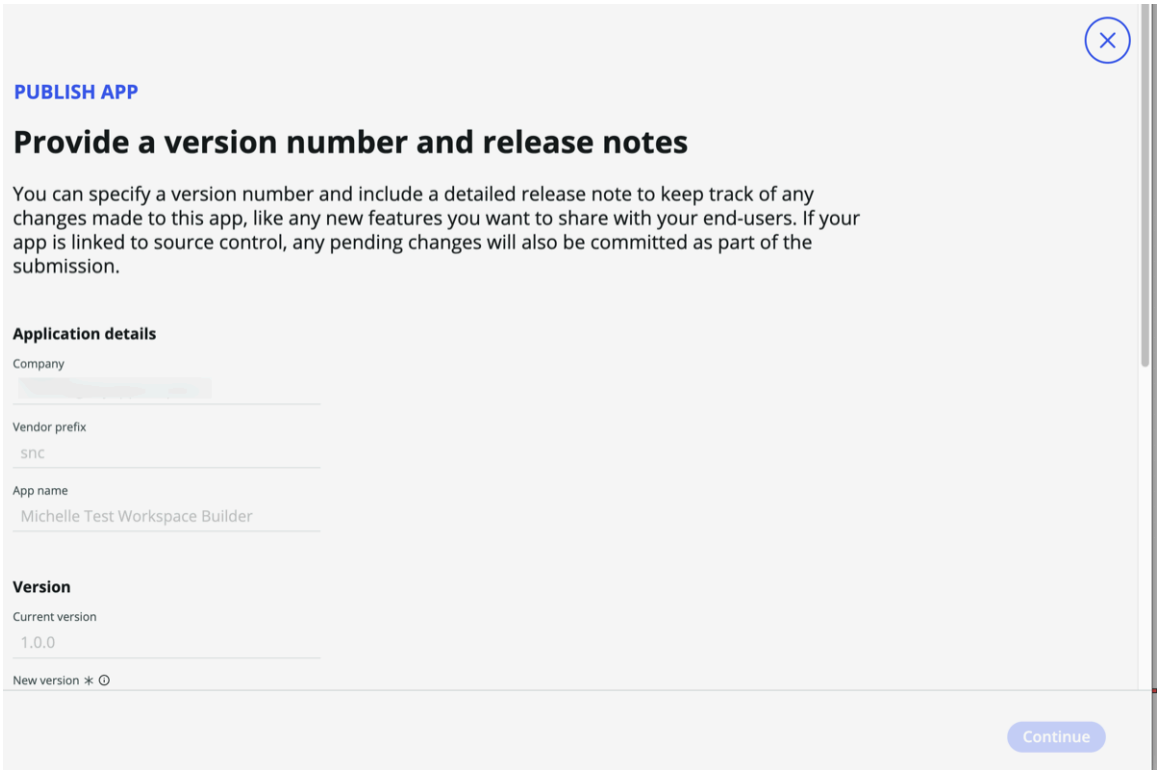
About this task

When you publish an application from App Engine Studio that is linked to source control, there is a different outcome than when you publish via the sys_app or sys_store_app **Publish** related link.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. From the app home, select **Publish**.
   App Engine Studio displays the publish app fields.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



4. Enter the following field values.

Provide a version number and release notes

Field	Description
Version	View your current app version and provide a new version number for your application. **Note:** To change the version number, you must have one of the following deployment permissions: **Submit for deployment**, **Publish to app repo**, or **Publish to app store**
Release notes	Provide a brief description of the changes made to your application.
Publish options	Select if you want the application to be published to your application repository or the ServiceNow Store.

Note: All application developers on the instance share the credential used to link a Git repository to an application.

5. Select **Continue**.
6. The current state of the application is committed to source control, including any untracked or uncommitted changes.
   The value of the **glide.sourcecontrol.default_commit_mode** property is ignored.
   This occurs because when the application is published, all the untracked and uncommitted changes are also published. Therefore, the state of the application in the Git repository matches what is published. See the Commit changes to a repository topic for more information about the **glide.sourcecontrol.default_commit_mode** property.
7. A source control tag is created for the new version and the application is published.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

If needed, the sys_app record is updated with the new store correlation ID.

Note: If your application is linked to source control and you publish a new version outside of App Engine Studio, a source control commit and tag are not created.

Remove an app from App Engine Studio

If you have the appropriate permissions, you can remove an application from App Engine Studio (AES).

Before you begin

Role required: admin, delete application, or delegated developer with permissions. For more information, see Delegate developers using AES.

About this task

Delegated developers can delete applications based on their permissions. To request delete permissions, contact your administrator.

Procedure

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open your application.
3. In your application, select the Edit application properties icon ( ).
4. Select **Delete application**.
5. On the dialog box, enter `delete` and then select **Delete**.

Using the App Engine Management Center

Track and manage your requests, deployments, applications, and collaborative developers using the App Engine Management Center (AEMC) in your production instance. Additionally, AEMC allows admins to manage app development from intake through production.

As of the Xanadu store 2 release, the App Engine Management Center (AEMC) can be used to manage app development for custom apps built in App Engine Studio, Creator Studio, and ServiceNow Studio. To learn more about managing app development using AEMC, see Managing app development using the App Engine Management Center.

App Engine Studio reference

Reference topics provide additional information about the lists and forms that you use to configure and administer App Engine Studio (AES).

AES glossary

Learn about terms and concepts that are unique to App Engine Studio (AES).

Application template

Provides predefined data, experience, logic and automation, and security to support a certain use case. For example, the Travel Request template provides application content for submitting and approving employee travel requests.

For more information on the available templates, see Available templates.

Logic and automation

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Automate all the work in your application by adding logic and automation. For example, you can build a flow that sends a notification to the admin when someone makes a request.

Data

Information that is stored in your application. For example, employee phone numbers or office locations. You configure application data using tables.

Deployment request

Ticket to track the review of submitted applications. From the deployment request form, a reviewer can deploy the application to different environments, accept or reject an application, and send feedback to a developer.

Environment

Instance that you use for the developing, testing, or launching an application. To set up App Engine Studio, you must define environments in each of the instances that you're using as environments.

Experience

Graphical interface that your users interact with. For example, you can create a portal where users find information, submit requests, or complete business tasks. For more information on the available application experiences, see Add an application experience.

Pipeline

Configuration for deploying an application to different environments.

Security

Roles and access controls to limit who can use your application.

AES properties

System properties control system behavior. The properties in this section are specific to the App Engine Studio (AES) application. You can access system properties for AES by navigating to **All** > **App Engine Studio** > **Configuration** > **Properties**.

sn_app_eng_studio.show_servicenow_studio_banner

Controls whether the try ServiceNow Studio banner appears on the App Engine Studio home page. Set to true to show the banner or set to false to hide the banner.

- Type: true | false
- Default value: true

sn_app_eng_studio.aes_admin_contact

Defines the contact email for the App Engine Studio administrator. The system sends deployment request notifications to each email address listed in this property. For more information, see Add users to the App Engine Admin group.

- Type: string
- Default value: none

sn_app_eng_studio.illustration_supported_content_types

Defines the supported content types for Taxonomy Category illustrations. This property is read-only.

- Type: string
- Default value: image/svg+xml

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

sn_app_intake.instance_can_provision_users

Determines if the current instance can support the provisioning of users. The users on this instance will be provisioned to another instance. For more information on provisioning users, see Configure Application Intake.

- Type: true | false
- Default value: false

Supported features and metadata in custom templates

Custom templates in App Engine Studio (AES) support most features and metadata, including, but not limited to tables, forms, roles, ACLs, record producers, simple workspaces, flows, and actions.

Custom templates can be created in two ways, either from scratch or using an existing scoped application as a starting point.

When you create a custom template, the pre-scan checks to make sure that everything in the application is supported. There are three possible outcomes.

- All items are supported: The custom template is created. All the objects are available in the custom template and in apps created from the custom template.
- Some items aren't supported: You can continue to create the template but a few items are not available in the custom template. There's no impact on the custom template or the apps created from the template.
- Some items are denied: You encounter an error message and cannot continue further until the errors are fixed. The author must change the app, if it was created from an existing app or the template, if it was created from scratch.

Note: Creating an AES template using an application that contains a workspace is not supported.

For more information on the template creation process, see Build a custom template.

Supported metadata

Label	Name
Access Control	sys_security_acl
Access Roles	sys_security_acl_role
Action Assignment	sys_declarative_action_assignment
Action Category	sys_hub_category
Action Definition	sys_declarative_action_definition
Action Exclusion	sys_workspace_declarative_action_exclusion
Action Inputs	sys_hub_action_input
Action Instance	sys_hub_action_instance
Action item	sys_sg_write_back_action_item
Action Model Field	sys_declarative_action_model_field
Action Outputs	sys_hub_action_output

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Label	Name
Action parameters mapping	sys_sg_action_param_map
Action Payload Definition	sys_declarative_action_payload_definition
Action Payload Field	sys_declarative_action_payload_field
Action Payload Mapping	sys_declarative_action_payload_mapping
Action Type	sys_hub_action_type_definition
Activity Definition	sys_pd_activity_definition
Activity Stream Screen	sys_sg_activity_stream_screen
Analytics preview	sys_sg_chart
Analytics section	sys_sg_chart_section
Application Menu	sys_app_application
Application Menu	sys_ui_application
Application Search Sources	m2m_search_context_config_search_source
Audience	sys_ux_applicability
Basic Auth Configuration	sys_auth_profile_basic
Builder Template	sys_uib_template
Business Rule	sys_script
Card	sys_sg_view_config
Card element	sys_sg_view_config_element
Card element attribute	sys_sg_view_config_element_attribute
Card template	sys_sg_view_template
Card template element	sys_sg_view_template_slot
Card template element attribute	sys_sg_view_template_slot_attribute
Catalog Client Scripts	catalog_script_client
Catalog Item	sc_cat_item
Catalog Item Available for	sc_cat_item_user_criteria_mtom
Catalog Item Category	sc_cat_item_category
Catalog Item Not Available for	sc_cat_item_user_criteria_no_mtom
Catalog Items Catalog	sc_cat_item_catalog
Catalog UI Policy	catalog_ui_policy
Catalog UI Policy Action	catalog_ui_policy_action
Catalog Variable Set	io_set_item
Category Property	sys_properties_category_m2m
Chart Colors	sys_report_chart_color
Chart screen	sys_sg_chart_screen
Choice	sys_choice

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Label	Name
Choice Set	sys_choice_set
Client Script	sys_script_client
Color Definition	sys_report_color
Complex Object	sys_complex_object
Composite Data Broker	sys_ux_data_broker_composite
Contained Role	sys_user_role_contains
Cross scope privilege	sys_scope_privilege
Data Broker Scriptlet	sys_ux_data_broker_scriptlet
Data Broker Server Script	sys_ux_data_broker_transform
Data Item	sys_sg_data_item
Data Parameter	sys_sg_item_parameter
Data Policy	sys_data_policy2
Data Policy Rule	sys_data_policy_rule
Decision	sys_decision_question
Decision Condition	sn_decision_table_decision_condition
Decision Input	sys_decision_input
Decision Table	sys_decision
Decision Table Multiple Result	sys_decision_multi_result
Details screen	sys_sg_details_screen
Dictionary Entry	sys_dictionary
Dictionary Entry Override	sys_dictionary_override
Dynamic Filter Options	sys_filter_option_dynamic
Email Address Filter	sys_email_address_filter
Email Address Filter Domain	sys_email_address_filt_domain
Email Client Configuration	sys_email_client_configuration
Email Client From Address	sys_email_client_from_address
Email Client Template	sys_email_client_template
Email Filter	sys_email_filter
Email Layout	sys_email_layout
Email Reply Separator	sys_email_reply_separator
Email Script	sys_script_email
Email Template	sysevent_email_template
Embedded Help Role Priority	sys_embedded_help_role
Empty State	sys_sg_empty_state
EVAM Data Filter	sys_ux_composite_data_filter

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Label	Name
EVAM Data Source Filter	sys_ux_composite_datasource_filter
EVAM Datasource	sys_ux_composite_datasource
EVAM Datasource M2M	sys_ux_composite_data_m2m_datasource
EVAM Definition	sys_ux_composite_data
EVAM View Config	sys_ux_composite_data_template_predicate
EVAM View Config Action Assignment M2M	sys_ux_predicate_m2m_action_assignment
EVAM View Config Bundle	sys_ux_composite_data_template_predicate_bundle
EVAM View Config Bundle M2M	sys_ux_composite_data_m2m_predicate_bundle
Event Registration	sysevent_register
Export Definition	sys_export_definition
Export Set	sys_export_set
Export Target	sys_export_target
Extended Flow Logic Inputs	sys_hub_flow_logic_ext_input
Extended Step Input Variable	sys_hub_step_ext_input
Extended Step Output Variable	sys_hub_step_ext_output
Extension Instance	sys_extension_instance
Extension Point	sys_extension_point
Facet	sys_search_facet
Field Label	sys_documentation
Field Map	sys_impex_entry
Filter	sys_filter
Filter Action	sys_email_filter_action
Flow	sys_hub_flow
Flow catalog variable	sys_flow_cat_variable
Flow Inputs	sys_hub_flow_input
Flow Logic Instance	sys_hub_flow_logic
Flow Snapshot	sys_hub_flow_snapshot
Flow Stages	sys_hub_flow_stage
Flow Variables	sys_hub_flow_variable
Form	sys_ui_form
Form Header	sys_aw_form_header
Form Section	sys_ui_section
Function	sys_sg_button
Function instance	sys_sg_button_instance
GraphQL Data Broker	sys_ux_data_broker_graphql

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Label	Name
Highlighted Value	sys_highlighted_value
Highlighted Value Condition	sys_highlighted_value_condition
HTTP Headers	sys_rest_message_headers
HTTP Headers	sys_rest_message_fn_headers
HTTP Method	sys_rest_message_fn
HTTP Query Parameters	sys_rest_message_fn_param_defs
Icon	sys_sg_icon
Icon section destination launcher	sys_sg_navigation_section_destination_applet_launcher
Images	db_image
Import Export Map	sys_impex_map
Inbound Email Actions	sysevent_in_email_action
Indexes	sys_index
Launcher header title	sys_sg_applet_launcher_header
Launcher screen	sys_sg_applet_launcher
Launcher section mapping	sys_sg_applet_launcher_m2m_section
Launcher tab	sys_sg_applet_launcher_tab
Layout Applicability	sys_ux_applicability_m2m_layout
Legacy card	sys_sg_item_view
Legacy icon section	sys_sg_icon_section
List	sys_ui_list
List Applicability	sys_ux_applicability_m2m_list
List Control	sys_ui_list_control
List item config	sys_sg_master_item
List screen	sys_sg_list_screen
List stream	sys_sg_item_stream
M2m Action Assig Ux Action Config	sys_ux_m2m_action_assignment_action_config
M2m Workspace Header Ux Header Config	sys_ux_m2m_workspace_header_ux_header_config
Many to Many Definition	sys_m2m
Message	sys_ui_message
Mobile UI Rule	sys_sg_ui_rule
Mobile UI Rule Action	sys_sg_ui_rule_action
Module	sys_ui_module
Module	sys_app_module
Navigation Tab Map	sys_sg_navigation_tab_map
New Record Menu Item	sys_aw_new_menu_item

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Label	Name
Notification	sys_notification
Notification	sysevent_email_action
Notification Category	sys_notification_category
Notification Common Content	sys_notification_common_content
Notification Route	notification_route
Notification Workspace Content	sys_notification_workspace_content
Number	sys_number
Playbook Activity Override	sys_playbook_activity_renderer
Playbook Activity UIs	sys_playbook_experience_activity_ui
Playbook Experience Action Assignment Map	sys_playbook_experience_action_assignment_map
Playbook Experience Record Generator	sys_playbook_experience_record_generator
Playbook Experiences	sys_playbook_experience
Process Definition	sys_pd_process_definition
Process Definition Swim Lane Images	sys_pd_swim_lane_image
Process Flow Trigger Plan	sys_flow_trigger_plan
Public Pages	sys_public
Push Notification	sys_sg_push_notification
Push Notification Installation	sys_push_notif_app_install
Push Notification Message	sys_push_notif_msg
Push Notification Message Content	sys_push_notif_msg_content
Question Choice	question_choice
Quick actions menu	sys_sg_alp_quick_action_map
Recipient Qualifier	sys_recipient_qualifier
Record Producer	sc_cat_item_producer
Record screen	sys_sg_form_screen
Record screen segment	sys_sg_form_segment
Record section	sys_sg_item_section
Redirection destination fields	sys_sg_redirect_dest_field
Related List	sys_ui_related_list
Relationship	sys_relationship
Relationship Data Item	sys_sg_relationship_data_item
Report	sys_report
Report Users and Groups	sys_report_users_groups
REST Data Broker	sys_ux_data_broker_rest
REST Message	sys_rest_message

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Label	Name
Result	sys_decision_multi_result_element
Role	sys_user_role
Route Configuration	sys_ux_dynamic_route_config
Route Mapping	sys_ux_dynamic_route_mapping
Scheduled Script Execution	sysauto_script
Screen Applicability	sys_ux_applicability_m2m_screen
Screen fields	sys_sg_screen_field
Screen parameters mapping	sys_sg_screen_param_map
Screen segment	sys_sg_item_stream_segment
Screen tab	sys_sg_applet_tab
Screen UI Policy	sys_sg_ui_policy
Script Action	sysevent_script_action
Script Include	sys_script_include
Scripted REST API	sys_ws_definition
Scripted REST Query Parameter	sys_ws_query_parameter
Scripted REST Query Parameter Association	sys_ws_query_parameter_map
Scripted REST Resource	sys_ws_operation
Search Application Configuration	sys_search_context_config
Search config	sys_sg_global_search
Search M2M item config	sys_sg_global_search_m2m_master_item
Search Source	sys_search_source
Secondary Values	sys_aw_form_header_secondary_values
Service Fulfillment Approval Step	sc_service_fulfillment_approval_step
Service Fulfillment Stage	sc_service_fulfillment_stage
Service Fulfillment Step	sc_service_fulfillment_step
Service Fulfillment Task Step	sc_service_fulfillment_task_step
Service Offering SLA	service_offering_sla
SLA Definition	contract_sla
SLA Timer Configuration	sla_timer_config
SLA timer configuration mapping	sla_timer_config_mapping
Step Instance	sys_hub_step_instance
Stream M2M item config	sys_sg_item_stream_m2m_master_item
Stream M2M segment	sys_sg_item_stream_m2m_segment
Style	sys_ui_style
Subflow Instance	sys_hub_sub_flow_instance

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Label	Name
Subflow Instance Inputs	sys_hub_sub_flow_instance_inputs
Suggestion Reader and Suggestion Reader Group Relations	m2m_sys_suggestion_reader_sys_suggestion_reader_group
Suggestion Reader Group	sys_suggestion_reader_group
Sys Ux M2m Action Layout Item	sys_ux_m2m_action_layout_item
Sys Ux M2m Highlighted Value Config	sys_ux_m2m_highlighted_value_config
Sys Ux M2m Workspace View Rule Ux View Rule Config	sys_ux_m2m_workspace_view_rule_ux_view_rule_config
System Property	sys_properties
System Property Category	sys_properties_category
Table	sys_db_object
Task Relationship Type	task_rel_type
Task Relationships Allowed	task_rel_allowed
Template	sys_template
Transform Script	sys_transform_script
Trigger Definition	sys_pd_trigger_definition
Trigger Execution	sys_flow_trigger_auto_script
Trigger Instance	sys_hub_trigger_instance
UI Action	sys_ui_action
UI Action Layout	sys_aw_form_uiaction_layout
UI Action Role	sys_ui_action_role
UI Formatter	sys_ui_formatter
UI Page	sys_ui_page
UI parameter	sys_sg_ui_parameter
UI Policy	sys_ui_policy
UI Policy Action	sys_ui_policy_action
UI Policy Rule	sys_sg_ui_policy_rule
UI Script	sys_ui_script
UI View	sys_ui_view
UIB Screen Test Values	sys_uib_screen_test_values
UX Actions Configuration	sys_ux_action_config
UX Add-on Event Mapping	sys_ux_addon_event_mapping
UX App Configuration	sys_ux_app_config
UX App Route	sys_ux_app_route
UX App Theme	m2m_app_config_theme
UX Application	sys_ux_page_registry

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Label	Name
UX Application Category M2M	sys_ux_registry_m2m_category
UX Asset	sys_ux_lib_asset
UX Client Script	sys_ux_client_script
UX Client Script Include	sys_ux_client_script_include
UX Custom Content Extension	sys_ux_custom_content_root_elem
UX Dashboard Route Permission	sys_ux_route_permission
UX Event	sys_ux_event
UX Form Actions	sys_ux_form_action
UX Form Actions Layout	sys_ux_form_action_layout
UX Form Actions Layout Group	sys_ux_form_action_layout_group
UX Form Actions Layout Item	sys_ux_form_action_layout_item
UX Header Configuration	sys_ux_header_config
UX Highlighted Value Configuration	sys_ux_highlighted_value_config
UX List	sys_ux_list
UX List Category	sys_ux_list_category
UX List Menu Configuration	sys_ux_list_menu_config
UX Macroponent Definition	sys_ux_macroponent
UX Page Element Permissions	sys_ux_page_element_m2m_role
UX Page Property	sys_ux_page_property
UX Ribbon Configuration	sys_ux_ribbon_config
UX Screen	sys_ux_screen
UX Screen Collection	sys_ux_screen_type
UX Screen Condition	sys_ux_screen_condition
UX Theme	sys_ux_theme
UX Theme Asset	sys_ux_theme_asset
UX Theme Assets	sys_ux_theme_m2m_asset
UX View Rules Configuration	sys_ux_view_rules_configuration
Variable	item_option_new
Variable Set	item_option_new_set
Variable Substitutions	sys_rest_message_fn_parameters
View Rule	sysrule_view_workspace
View Rule	sysrule_view
View Template	sys_ux_composite_data_template
Workspace	sys_aw_master_config
Workspace Global Search Configuration	sys_aw_global_search_config

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Label	Name
Workspace List	sys_aw_list
Workspace List Category	sys_aw_list_category
Workspace Module	sys_aw_module

Skipped features and metadata

The template creation process only skips items that should not impact the template or apps created from the template.

Label	Name
Activity Designer	wf_element_activity
Activity Variables	wf_activity_variable
Content Link	content_link
Content Themes	content_theme
Content Type	content_type
Database View	sys_db_view
Detail Pages	content_type_detail
Detailed Content	content_block_detail
Disallows "create from scratch" templates to be re-templatized if there are running executions	
Dynamic Content	content_block_programmatic
Flash Movie	content_block_flash
Flow Template Mapping	sys_app_flow_template_mapping
IFrames	content_block_iframe
Input Detail	sys_app_template_input_detail
Lists of Content	content_block_lists
Login Rule	content_page_rule
Menu Items	menu_item
Menu Sections	menu_section
Meta Tags	content_page_meta
Navigation Menu	content_block_menu
Pages	content_page
Payload	sys_app_scan_payload
Payload Loader Rule	sys_app_payload_loader_rule
Payload Rule	sys_app_payload_unloader_rule
Processor	sys_processor
Site	content_site

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Label	Name
Spoke Configuration	sys_app_template_spoke_configuration
Stage Default	wf_stage_default
Static Content	content_block_static
Style Sheet	content_theme_css
Subtemplate	sys_app_subtemplate
Template	sys_app_template
Template Dependency	sys_app_template_dependency
Template Dependency Definition	sys_app_template_dependency_definition
Template Details	sys_app_template_detail
Template Function Type	sys_app_template_function_type
Template Input Variable	sys_app_template_input_var
Template Manifest	sys_app_template_manifest
Template Output Variable	sys_app_template_output_var
Template Page	sys_app_template_page
Template Page Configuration	sys_app_template_page_configuration
Template Wizard	sys_app_template_wizard
Variable	sys_app_scan_variable
Variable Type	sys_app_scan_variable_type
View Field	sys_db_view_table_field
View Table	sys_db_view_table
Workflow	wf_workflow
Workflow Activity Definition	wf_activity_definition
Workflow Instance	wf_workflow_instance
Workflow SC Variable	wf_variable
Workflow Schedule	wf_workflow_schedule
Workflow UI Policy	wf_ui_policy

ServiceNow CLI

The ServiceNow CLI is a command-line interface that lets you perform instance operations from your local system. You can extend the CLI to include new commands that meet your application's needs.

Benefits

The ServiceNow CLI lets you:

- Perform basic CRUD operations on records in your instance.
- Develop custom components and deploy them to your instance to personalize a UI.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Create custom commands that enable you to manage custom applications from the command line.
- Use the ServiceNow CLI in scripts to simplify setup tasks and operational activities.

Architecture

Commands are stored in a table on the instance you are connected to. When the ServiceNow CLI connects to the instance, it receives all the available commands supported by that instance.

Commands map to a REST endpoint that executes asynchronously. For more information, see Create a custom command in ServiceNow CLI.

Activating ServiceNow CLI

Install ServiceNow CLI by requesting it from the ServiceNow Store. Visit the ServiceNow Store to view all the available apps, and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the ServiceNow Store version history release notes.

Configuration file

The ServiceNow CLI stores profile information in a `config.json` file which, by default, is stored in your home directory at the following path:

- Linux and Mac: `~/.snc/config.json`
- Windows: `%USERPROFILE%\.snc\config.json`

The CLI uses this file to determine what information to use to connect to an instance, and what settings to use to generate output. By default, the ServiceNow CLI uses the settings found in the default profile to connect to an instance. To use alternate settings, you can create and reference additional named profiles. For more information, see Configuring and managing your ServiceNow CLI connection profiles.

The following example shows a#configuration#file with a default profile and a named profile. Each profile can use different credentials and specify different hosts and output formats.

```
{
   "profiles":{
      "default":{
         "host":"https://myinstance.service-now.com",
         "loginmethod":"basic",
         "username":"admin",
         "output":"json",
         "hostversion":"Paris",
         "appversion":"1.0"
         },
      "user1":{
         "host":"https://otherinstance.service-now.com",
         "loginmethod":"basic",
         "username":"user1",
         "output":"yaml",
         "hostversion":"Paris",
         "appversion":"1.0"
         }
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
      }
}
```

Note: Sensitive credential information is only stored in the OS keychain, not in the configuration file.

Command structure

ServiceNow CLI commands follow this structure:

1. The base call to the `snc` program.
2. The top-level command group followed by any child command groups.
3. The command that specifies which operation to perform.
4. General CLI arguments required by the operation. You can specify arguments in any order.

```
$ snc <command-group> <command> [arguments]
```

Arguments can take various types of input values, such as numbers, strings, and JSON objects. The types supported depend on the command you specify.

Argument values

Many argument values in the ServiceNow CLI are simple string or numeric values, such as the table and table name in the following example.

```
$ snc record create --table incident --data "{short_description:
 'New Incident'}"
```

You can surround strings that do not contain any space characters with quotation marks or not. However, you must use quotation marks around strings that include one or more space characters.

Output formats

The ServiceNow CLI supports four output formats:

- `json`: The output is formatted as JSON. This is the default.

```
{
   "default": {
      "appversion": "1.0.8",
      "host": "https://myinstance.service-now.com",
      "hostversion": "Paris",
      "loginmethod": "basic",
      "output": "json",
      "username": "admin"
   },
   "user1": {
      "appversion": "1.0.8",
      "host": "https://otherinstance.service-now.com",
      "hostversion": "Paris",
      "loginmethod": "basic",
      "output": "yaml",
      "username": "admin"
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    }
}
```

- `yaml`: The output is formatted as YAML. Use YAML to handle the output with services and tools that emit or consume YAML-formatted strings.

```
default:
   appversion: 1.0.8
   host: https://myinstance.service-now.com
   hostversion: Paris
   loginmethod: basic
   output: json
   username: admin
user1:
   appversion: 1.0.8
   host: https://otherinstance.service-now.com
   hostversion: Paris
   loginmethod: basic
   output: yaml
   username: admin
```

- `text`: The output is formatted as multiple lines of tab-separated string values. Use this output with traditional UNIX text tools such as grep, sed, and awk, and the text processing performed by PowerShell.

```
default   https://myinstance.service-now.com   Paris   1.0.8
 basic   admin   json
user1   https://otherinstance.service-now.com   Paris   1.0.8
 basic   admin   yaml
```

- `table`: The output is formatted as a table which presents the information in a human-readable format.

```
NAME   HOST   HOST   VERSION   APP   VERSION   LOGIN   METHOD
 USERNAME   OUTPUT
-------------------------------------------------------------------
----------------
default   myinstance   Paris   1.0.8   basic   admin   json
user1   otherinstance   Paris   1.0.8   basic   admin   yaml
```

- `none`: The CLI does not print the output to the console. Success, error, and progress messages still display.

You can specify command output in two ways:

Use the `output` option in a named profile in the configuration file

The following example sets the default output format to text.

```
{
    "profiles":{
       "default":{
          "output":"text"
       }
}
```

Use the `--output` argument on the command line

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The following example sets the output of a single command to JSON. This option on the command overrides any currently set value in the configuration file.

```
$ snc record query --table incident --query
 'active=true' --output json
```

Install the ServiceNow CLI

Install the ServiceNow CLI on a Mac, Windows, or Linux machine.

The latest version of the ServiceNow CLI is 1.1.0. To find your installed version and see if you must update, run `snc version`.

Installing the ServiceNow CLI requires:

- Admin rights to install software on your machine.
- Software to extract or unzip the downloaded package.

The sn_cli_metadata plugin needs to be installed on your instance to support full CLI functionality.

Role required: admin

Available for the following operating systems:

- Mac
- Windows
- Linux

Install the ServiceNow CLI on Mac

Install ServiceNow CLI on a Mac OS using the installer.

Before you begin

You must have an Apple-supported version of 64-bit Mac OS.
Role required: admin

Procedure

1. In your browser, download the installer bundle from the ServiceNow Store.
2. Extract the OS-specific installers.
3. Open the `snc-1.0.0-osx-installer` file.
4. Follow the on-screen instructions.
   a. When prompted, ensure the **Add to PATH** option is selected.
   This option requires root account access.
5. To verify that the installation succeeded, use the following commands.

```
$ which snc

/usr/local/bin/snc
```

```
$ snc version
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
{
 "extensions": {},
 "snc": "1.0.0"
}
```

Install the ServiceNow CLI on Windows

Install ServiceNow CLI on a Windows OS using the installer.

Before you begin

You must have a 64-bit version of Windows 10 or later.
Role required: admin

Procedure

1. In your browser, download the installer bundle from the ServiceNow Store.
2. Extract the OS-specific installers.
3. Open the `snc-1.0.0-windows-x64-installer.exe` file.
4. Follow the on-screen instructions.
   By default, the CLI installs to `C:\Program Files\ServiceNow CLI`.
5. To confirm the installation, check the version in the command line using the following command.

   ```
   C:\>snc version

   {
    "extensions": {},
    "snc": "1.0.0"
   }
   ```

   If Windows is unable to find the program, close and reopen the command prompt window to refresh the path.

Install the ServiceNow CLI on Linux

Install ServiceNow CLI on a Linux machine using the installer.

Before you begin

You must have a 64-bit version of a recent distribution of CentOS or Ubuntu.
Role required: admin

Procedure

1. In your browser, download the installer bundle from the ServiceNow Store.
2. Extract the OS-specific installers.
3. Open the `snc-1.0.0-linux-x64-installer.run` file.
4. Follow the on-screen instructions.

   a. When prompted, ensure the **Add to PATH** option is selected.
   This option requires root account access.

5. To verify that the installation succeeded, use the following commands.

   ```
   $ which snc
   ```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
~/ServiceNow CLI/bin/snc
```

```
$ snc version

{
 "extensions": {},
 "snc": "1.0.0"
}
```

Configuring and managing your ServiceNow CLI connection profiles

Create a connection profile to connect with your instance, view connection profiles, refresh your connection and available commands, or delete profiles you no longer need.

The ServiceNow CLI stores profile information in a `config.json` file which, by default, is stored in your home directory at the following path:

- Linux and Mac: `~/.snc/config.json`
- Windows: `%USERPROFILE%\.snc\config.json`

The CLI uses this file to determine what information to use to connect to an instance, and what settings to use to generate output. By default, the ServiceNow CLI uses the settings found in the default profile to connect to an instance. To use alternate settings, you can create and reference additional named profiles. For more information, see Configuring and managing your ServiceNow CLI connection profiles.

The following example shows a#configuration#file with a default profile and a named profile. Each profile can use different credentials and specify different hosts and output formats.

```
{
   "profiles":{
      "default":{
         "host":"https://myinstance.service-now.com",
         "loginmethod":"basic",
         "username":"admin",
         "output":"json",
         "hostversion":"Paris",
         "appversion":"1.0"
         },
      "user1":{
         "host":"https://otherinstance.service-now.com",
         "loginmethod":"basic",
         "username":"user1",
         "output":"yaml",
         "hostversion":"Paris",
         "appversion":"1.0"
         }
      }
}
```

Create a default profile

Create a connection profile that the ServiceNow CLI uses by default. You must create a default profile to set up the CLI's initial connection with an instance.

Before you begin
Role required: none.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

About this task

By default, the information in this profile is used when you run a command that does not explicitly specify a profile to use.

Note: Sensitive credential information is only stored in the OS keychain, not in the configuration file.

Procedure

1. Open your system's command-line tool and execute this command.

```
$ snc configure profile set
```

   The CLI prompts you for more information.

2. Enter the requested values.
   When prompting for information, the CLI displays current values in brackets []. To keep an existing value, press the Enter key.

Requested information	Description
Host	The host name of the instance to connect to. Supports both the full URL (https://my-instance.service-now.com) or just the host name (my-instance).
Login method	The login method to use to connect to the instance. Supports `Basic`, `OAuth`, and `OAuth + MFA`.
Username	The user name to use to connect to the instance.
Password	The password to use to connect to the instance.
Client id	The client ID to use to connect to the instance when the login method is OAuth or OAuth + MFA.
Client secret	The client secret to use to connect to the instance when the login method is OAuth or OAuth + MFA.
Authentication code	The authentication code to use to connect to the instance when the login method is OAuth + MFA.
Default output format	Specifies how to format the command results. Options are `json`, `yaml`, `text`, and `table`.

Example

```
$ snc configure profile set

Host:
https://myinstance.service-now.com

Login Method (Basic, OAuth, OAuth + MFA):
Basic

Username:
myusername

Password:
mypassword
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

3. Run any command using the values specified in your default profile by omitting the `--profile` attribute from your command.

   Example

   ```
   $ snc record create
   ```

   The command creates a record in the instance specified in the default profile with the specified connection options.

Create a named profile

Create a named connection profile to use with specific commands. This allows you to specify a different instance or connection protocol for a specific command.

Before you begin

A default profile must first exist to communicate with an instance.

Role required: none.

About this task

Note: Sensitive credential information is only stored in the OS keychain, not in the configuration file.

Procedure

1. Open your system's command-line tool and execute this command.
2. Specify the name of the named profile in the *`profile-name`* argument.

   ```
   $ snc configure profile set [--profile profile-name]
   ```

   The CLI prompts you for more information.
3. Enter the requested values.
   When prompting for information, the CLI displays current values in brackets []. To keep an existing value, press the Enter key.

Requested information	Description
Host	The host name of the instance to connect to. Supports both the full URL (https://my-instance.service-now.com) or just the host name (my-instance).
Login method	The login method to use to connect to the instance. Supports `Basic`, `OAuth`, and `OAuth + MFA`.
Username	The user name to use to connect to the instance.
Password	The password to use to connect to the instance.
Client id	The client ID to use to connect to the instance when the login method is OAuth or OAuth + MFA.
Client secret	The client secret to use to connect to the instance when the login method is OAuth or OAuth + MFA.
Authentication code	The authentication code to use to connect to the instance when the login method is OAuth + MFA.
Default output format	Specifies how to format the command results. Options are `json`, `yaml`, `text`, and `table`.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example

```
$ snc configure profile set --profile user1

Host:
https://mytestinstance.service-now.com

Login Method (Basic ,OAuth ,OAuth + MFA):
Basic

Username:
myusername

Password:
mypassword
```

4. Run any command using the values specified in your named profile by adding the`#--profile# <profile-name>` attribute to your command.

   Example

   ```
   $ snc record create --profile user1
   ```

   The command creates a record in the instance specified in the `user1` profile with the specified connection options.

View profiles

View all connection profiles set in the configuration file, or view information about a specific profile.

Before you begin

A default profile must first exist to communicate with an instance.

Role required: none.

About this task

For each profile, the CLI displays host information, version details, user name, login method, and preferred output format. It does not display sensitive information such as passwords or client IDs.

Procedure

Open your system's command-line tool and execute this command.

Include the `[--profile profile-name]` argument to view the connection information for a single profile. Otherwise, the CLI lists information for all profiles.

```
$ snc configure profile list [--profile profile-name]
```

Example

This command displays all configured profiles.

```
$ snc configure profile list
```

The CLI provides the following output.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
{
   "profiles":{
      "default":{
      "host":"https://myinstance.service-now.com",
      "loginmethod":"basic",
      "username":"admin",
      "output":"json",
      "hostversion":"Paris",
      "appversion":"1.0"
   }
}
```

Remove a profile

Remove a named connection profile that you no longer need from the configuration file.

Before you begin

A default profile must first exist to communicate with an instance.

Role required: none.

About this task

You cannot remove the default connection profile using this command. To remove the default connection profile, edit the configuration file manually.

Procedure

1. Open your system's command-line tool and execute this command.
2. Specify the name of the named profile in the *`profile-name`* argument.

```
$ snc configure profile remove [--profile profile-name]
```

Result

The CLI removes the specified profile from the configuration file.

Refresh your connection

Update the available commands from the instance for the given profile. Refresh your connection after modifying any of the commands on the corresponding instance in order to keep the CLI up-to-date.

Before you begin

A default profile must first exist to communicate with an instance.

Role required: none.

Procedure

Open your system's command-line tool and execute this command.

Include the `[--profile profile-name]` argument to refresh connection information from a named profile. If you do not include the argument, the CLI refreshes the default profile.

```
$ snc configure profile refresh [--profile profile-name]
```

Result

The CLI connects to the instance in the designated connection profile and updates any commands that may have changed.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Get help with ServiceNow CLI

See available commands, command options, and examples, or generate debug logging output.

Before you begin

- Install the ServiceNow CLI
- Configuring and managing your ServiceNow CLI connection profiles
- Role required: none

About this task

In addition to the debug logging output option, the CLI logs all command executions to a log file at the following locations:

- Linux and Mac: `~/.snc/.logs`.
- Windows:`%USERPROFILE%\.snc\.logs`.

Procedure

1. To see the available commands and arguments, open your system's command-line tool and enter the command you need help with, followed by the `--help` argument.

   Example

   The following example displays help for the available top-level commands.

   ```
   $ snc --help
   ```

   Example

   The following example displays the available profile-specific commands.

   ```
   $ snc configure profile --help
   ```

   The CLI displays help for the requested command divided into eight sections:

   Name

   The name of the command.

   ```
   Name
        set
   ```

   Description

   The description of what the command does and its return values.

   ```
   Description
        Configure connection profiles. This command
    is interactive and will prompt you for each
    configuration value.
   ```

   Synopsis

   The basic syntax for using the command and its options.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
Synopsis
     snc configure profile set [arguments]
```

Available commands

The commands available under the current command group.

```
Available Commands
 list    : Lists the configured connection profiles.
 refresh : Updates the available commands from the
 instance for the given profile.
 remove  : Removes the specified connection profile.
 set     : Configures connection profiles in order to
 communicate with an instance.
```

Command groups

The command groups available under the current command group.

```
Command Groups
     profile : Set, view, and remove connection
 profiles.
```

Arguments

A description of each of the arguments the command accepts.

```
Arguments
     -p, --profile string : Use a specific connection
 profile when executing a command.
```

Global arguments

The global arguments that the command accepts.

```
Global Arguments
     -d, --debug Print logs to console.
     -h, --help Display detailed help information.
     -o, --output string Set the format for printing
 command output.
```

Examples

Examples of the requested command.

```
Examples
   Create a new profile to save as the default:
      $ snc configure profile set
         Host:
         Login method:
         Username:
         Password:
         Client id:
         Client secret:
```

2. To generate debug logging output, open your system's command-line tool and enter the command you want to debug, followed by the `--debug` argument.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example

The following example generates debug logging output when trying to perform the operation.

```
$ snc record delete --table incident --sysid
 552c48888c033300964f4932b03eb092 --debug
```

The CLI generates debug logging output when executing the command.

Perform record operations using ServiceNow CLI

Create, read, update, delete, and query records in your instance using the ServiceNow CLI command-line tool.

- Install the ServiceNow CLI
- Configuring and managing your ServiceNow CLI connection profiles

Note: To perform record operations, install the CLI Metadata application from the ServiceNow Store on the instance you want to connect to.

Create a record

Inserts a single record in a specified table.

Before you begin

Role required: sn_cli_metadata.cli_admin, sn_cli_metadata.cli_user, or admin

About this task

Note: You cannot insert multiple records using this command.

Procedure

Open your system's command-line tool and execute this command.

```
$ snc record create [--table table --data data]
```

Pass in values for these arguments.

Parameter	Description
table	Required. Name of the table in which to save the record.
data	Required. Field name and the associated value for each field to define in the specified record in JSON string format.

Result

The system creates a record in the designated table and returns field-value pairs of the new record.

Example:

```
$ snc record create --table incident, --data
 "{'short_description': 'Unable to open file', 'impact':'3'}"
```

The system returns the record in JSON format.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
{
   "result": {
      "active": "true",
      "activity_due": "",
      "additional_assignee_list": "",
      "approval": "not requested",
      "approval_history": "",
      "approval_set": "",
      "assigned_to": "",
      "assignment_group": "",
      "business_duration": "",
      "business_service": "",
      "business_stc": "",
      "calendar_duration": "",
      "calendar_stc": "",
      "caller_id": "",
      "category": "inquiry",
      "caused_by": "",
      "child_incidents": "0",
      "close_code": "",
      "close_notes": "",
      "closed_at": "",
      "closed_by": "",
      "cmdb_ci": "",
      "comments": "",
      "comments_and_work_notes": "",
      "company": "",
      "contact_type": "",
      "contract": "",
      "correlation_display": "",
      "correlation_id": "",
      "delivery_plan": "",
      "delivery_task": "",
      "description": "",
      "due_date": "",
      "escalation": "0",
      "expected_start": "",
      "follow_up": "",
      "group_list": "",
      "hold_reason": "",
      "impact": "3",
      "incident_state": "1",
      "knowledge": "false",
      "location": "",
      "made_sla": "true",
      "notify": "1",
      "number": "INC0010005",
      "opened_at": "2021-01-27 22:49:11",
      "opened_by": {
         "link":
 "https://my-
instance.service-now.com/api/now/table/sys_user/6816f79cc0a80164
01c5a33be04be441",
         "value": "6816f79cc0a8016401c5a33be04be441"
      },
      "order": "",
```


© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
      "parent": "",
      "parent_incident": "",
      "priority": "5",
      "problem_id": "",
      "reassignment_count": "0",
      "reopen_count": "0",
      "reopened_by": "",
      "reopened_time": "",
      "resolved_at": "",
      "resolved_by": "",
      "rfc": "",
      "route_reason": "",
      "service_offering": "",
      "severity": "3",
      "short_description": "Unable to open file",
      "sla_due": "",
      "state": "1",
      "subcategory": "",
      "sys_class_name": "incident",
      "sys_created_by": "admin",
      "sys_created_on": "2021-01-27 22:49:11",
      "sys_domain": {
         "link":
 "https://my-
instance.service-now.com/api/now/table/sys_user_group/global",
         "value": "global"
      },
      "sys_domain_path": "/",
      "sys_id": "6a6a4f5cdbcae850d7055268dc9619d4",
      "sys_mod_count": "0",
      "sys_tags": "",
      "sys_updated_by": "admin",
      "sys_updated_on": "2021-01-27 22:49:11",
      "task_effective_number": "INC0010005",
      "time_worked": "",
      "universal_request": "",
      "upon_approval": "proceed",
      "upon_reject": "cancel",
      "urgency": "3",
      "user_input": "",
      "watch_list": "",
      "work_end": "",
      "work_notes": "",
      "work_notes_list": "",
      "work_start": ""
   }
}
```


Delete a record

Deletes the specified record from the specified table.

Before you begin

Role required: sn_cli_metadata.cli_admin, sn_cli_metadata.cli_user, or admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

Open your system's command-line tool and execute this command.

```
$ snc record delete [--table table --sysid sys_id]
```

Pass in values for these arguments.

Parameter	Description
table	Required. Name of the table in which to delete the record.
sysid	Required. Sys_id of the record to delete.

Example:

```
$ snc record delete --table incident --sysid
 552c48888c033300964f4932b03eb092
```

Get a record

Retrieves a single record based on the specified sys_id from the specified table.

Before you begin
Role required: sn_cli_metadata.cli_admin, sn_cli_metadata.cli_user, or admin

Procedure

Open your system's command-line tool and execute this command.

```
$ snc record get [--table table --sysid sys_id]
```

Pass in values for these arguments.

Parameter	Description
table	Required. Name of the table from which to retrieve the record.
sysid	Required. Sys_id of the record to retrieve.

Example:

```
$ snc record get --table incident --sysid
 552c48888c033300964f4932b03eb092
```

The system returns the record in JSON format.

```
{
   "result": {
      "active": "true",
      "activity_due": "",
      "additional_assignee_list": "",
      "approval": "not requested",
      "approval_history": "",
      "approval_set": "",
      "assigned_to": "",
      "assignment_group": "",
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
      "business_duration": "",
      "business_service": "",
      "business_stc": "",
      "calendar_duration": "",
      "calendar_stc": "",
      "caller_id": {
         "link":
 "https://my-
instance.service-now.com/api/now/table/sys_user/005d500b536073005e0addeeff7b12f4",
         "value": "005d500b536073005e0addeeff7b12f4"
      },
      "category": "inquiry",
      "caused_by": "",
      "child_incidents": "0",
      "close_code": "",
      "close_notes": "",
      "closed_at": "",
      "closed_by": "",
      "cmdb_ci": "",
      "comments": "",
      "comments_and_work_notes": "",
      "company": "",
      "contact_type": "",
      "contract": "",
      "correlation_display": "",
      "correlation_id": "",
      "delivery_plan": "",
      "delivery_task": "",
      "description": "",
      "due_date": "",
      "escalation": "0",
      "expected_start": "",
      "follow_up": "",
      "group_list": "",
      "hold_reason": "",
      "impact": "3",
      "incident_state": "1",
      "knowledge": "false",
      "location": "",
      "made_sla": "true",
      "notify": "1",
      "number": "INC0010112",
      "opened_at": "2019-07-29 18:48:43",
      "opened_by": {
         "link":
 "https://my-
instance.service-now.com/api/now/table/sys_user/6816f79cc0a8016401c5a33be04be441",
         "value": "6816f79cc0a8016401c5a33be04be441"
      },
      "order": "",
      "parent": "",
      "parent_incident": "",
      "priority": "5",
      "problem_id": "",
      "reassignment_count": "0",
```


© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
      "reopen_count": "0",
      "reopened_by": "",
      "reopened_time": "",
      "resolved_at": "",
      "resolved_by": "",
      "rfc": "",
      "route_reason": "",
      "service_offering": "",
      "severity": "3",
      "short_description": "Assessment :  ATF Assessor",
      "sla_due": "",
      "state": "1",
      "subcategory": "",
      "sys_class_name": "incident",
      "sys_created_by": "admin",
      "sys_created_on": "2019-07-29 18:49:28",
      "sys_domain": {
         "link":
 "https://my-
instance.service-now.com/api/now/table/sys_user_group/global",
         "value": "global"
      },
      "sys_domain_path": "/",
      "sys_id": "552c48888c033300964f4932b03eb092",
      "sys_mod_count": "0",
      "sys_tags": "",
      "sys_updated_by": "admin",
      "sys_updated_on": "2019-07-29 18:49:28",
      "task_effective_number": "INC0010112",
      "time_worked": "",
      "universal_request": "",
      "upon_approval": "proceed",
      "upon_reject": "cancel",
      "urgency": "3",
      "user_input": "",
      "watch_list": "",
      "work_end": "",
      "work_notes": "",
      "work_notes_list": "",
      "work_start": ""
   }
}
```


Query records

Retrieves multiple records from a specified table.

Before you begin

Role required: sn_cli_metadata.cli_admin, sn_cli_metadata.cli_user, or admin

Procedure

Open your system's command-line tool and execute this command.

```
$ snc record query [--displayvalue displayValue --fields fields
 --limit limit --offset offset --query query --table table]
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Pass in values for these arguments.

Parameter	Description
displayValue	Include `--displayvalue` to retrieve the display value from the database for reference and choice fields. Do not include this parameter to retrieve the actual values.
fields	Comma-separated list of field names to return from the database.
limit	Maximum number of records to return.
offset	Starting record index for which to begin retrieving records. Use this value to paginate record retrieval.
query	Required. Encoded query used to filter the result set in the following format: `--query '<column_name><operator><value>'`.
table	Required. Name of the table in which to query the records.

Example:

```
$ snc record query --displayvalue --fields
 short_description,state --query '123TEXTQUERY321=email' --table
 incident
```

The CLI returns any records that match the query.

```
{
   "result": [
      {
         "short_description": "Unable to connect to email",
         "state": "Closed"
      }
   ]
}
```

Update a record

Updates the specified record with the given data attributes.

Before you begin

Role required: sn_cli_metadata.cli_admin, sn_cli_metadata.cli_user, or admin

Procedure

Open your system's command-line tool and execute this command.

```
$ snc record update [--sysid sys_id --table table --data data]
```

Pass in values for these arguments.

Parameter	Description
table	Required. Name of the table in which to save the record.
sysid	Required. Sys_id of the record to update.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Parameter	Description
data	Required. Field name and the associated value for each field to define in the specified record in JSON string format.

Example:

```
$ snc record update --sysid 9ef81de2db0d6090d7055268dc961978
 --table incident --data "{'short_description': 'Email servers
 down', 'urgency':'1'}"
```

The system returns field-value pairs for the updated record.


```
{
   "result": {
      "active": "true",
      "activity_due": "",
      "additional_assignee_list": "",
      "approval": "not requested",
      "approval_history": "",
      "approval_set": "",
      "assigned_to": "",
      "assignment_group": {
         "link":
 "https://my-
instance.service-now.com/api/now/table/sys_user_group/287ebd7da9
fe198100f92cc8d1d2154e",
         "value": "287ebd7da9fe198100f92cc8d1d2154e"
      },
      "business_duration": "",
      "business_service": "",
      "business_stc": "",
      "calendar_duration": "",
      "calendar_stc": "",
      "caller_id": "",
      "category": "inquiry",
      "caused_by": "",
      "child_incidents": "0",
      "close_code": "",
      "close_notes": "",
      "closed_at": "",
      "closed_by": "",
      "cmdb_ci": "",
      "comments": "",
      "comments_and_work_notes": "",
      "company": "",
      "contact_type": "",
      "contract": "",
      "correlation_display": "",
      "correlation_id": "",
      "delivery_plan": "",
      "delivery_task": "",
      "description": "",
      "due_date": "",
      "escalation": "0",
      "expected_start": "",
      "follow_up": "",
```


© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
      "group_list": "",
      "hold_reason": "",
      "impact": "2",
      "incident_state": "1",
      "knowledge": "false",
      "location": "",
      "made_sla": "true",
      "notify": "1",
      "number": "INC0010003",
      "opened_at": "2020-12-15 21:17:05",
      "opened_by": {
         "link":
 "https://my-
instance.service-now.com/api/now/table/sys_user/6816f79cc0a80164
01c5a33be04be441",
         "value": "6816f79cc0a8016401c5a33be04be441"
      },
      "order": "",
      "parent": "",
      "parent_incident": "",
      "priority": "2",
      "problem_id": "",
      "reassignment_count": "0",
      "reopen_count": "0",
      "reopened_by": "",
      "reopened_time": "",
      "resolved_at": "",
      "resolved_by": "",
      "rfc": "",
      "route_reason": "",
      "service_offering": "",
      "severity": "3",
      "short_description": "Email servers down",
      "sla_due": "",
      "state": "1",
      "subcategory": "",
      "sys_class_name": "incident",
      "sys_created_by": "admin",
      "sys_created_on": "2020-12-15 21:17:05",
      "sys_domain": {
         "link":
 "https://my-
instance.service-now.com/api/now/table/sys_user_group/global",
         "value": "global"
      },
      "sys_domain_path": "/",
      "sys_id": "9ef81de2db0d6090d7055268dc961978",
      "sys_mod_count": "1",
      "sys_tags": "",
      "sys_updated_by": "admin",
      "sys_updated_on": "2021-01-27 22:56:33",
      "task_effective_number": "INC0010003",
      "time_worked": "",
      "universal_request": "",
      "upon_approval": "proceed",
      "upon_reject": "cancel",
      "urgency": "1",
```


© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
      "user_input": "",
      "watch_list": "",
      "work_end": "",
      "work_notes": "",
      "work_notes_list": "",
      "work_start": ""
   }
}
```

Create a custom command in ServiceNow CLI

Manage your custom application from the command line by creating custom commands in the ServiceNow CLI.

Before you begin

Role required: admin

About this task

A ServiceNow CLI command maps to a scripted REST endpoint in the End Point [sn_cli_metadata_end_point] table. You can define a scripted REST endpoint to perform a function in your custom application, or use any existing REST endpoint. Then map a CLI command to execute the REST call.

Procedure

1. Make a REST endpoint available to a ServiceNow CLI command.

   a. Navigate to **Command Line Interface (CLI)** > **End Points**.

   b. Select **New** and complete the form.

Field	Description
Resource Path	Required. Path to the endpoint on the instance to map a command to. Can be the path to an inbound REST API, or a scripted REST API. For example, `api/now/table/{table}/{sysid}`. For more information, see Available REST APIs and Scripted REST APIs.
HTTP Method	Required. HTTP method to use when the user runs the associated command.
Application	Read-only application scope for the endpoint.

2. **Optional:** Create a command group.
   Alternatively, you can add your new command to an existing command group.

   a. Navigate to **Command Line Interface (CLI)** > **Command Groups**.

   b. Select **New** and complete the form.

Field	Description
Name	Required. Name of the command group.
Parent Group	Parent command group.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Reference Group	Command group to reference. For example, you can create a new command group as an alias for an existing command group. When the user calls a referenced command using the new command group, the original command executes. This enables you to create a command group specific to your custom application that includes both new and existing commands.
Short Description	Required. Short description for the command group.
Description	Description of the command group used as help text when the user runs the `--help` command on the command group.
Application	Read-only application scope for the command group.
Active	When selected, the command group is active.

3. Create a command.

   a. Navigate to **Command Line Interface (CLI)** > **Commands**.

   b. Select **New** and complete the form.

Field	Description
Name	Required. Name of the command.
Command Group	Required. Command group that the command is a part of.
Reference Command	Command to reference. For example, you can create a new command as an alias for an existing command. When the user calls the new command, the original command executes. This enables you to create a command specific to your custom application that executes existing functionality. **Note:** A command cannot reference a command that references another command, or reference a descendant command, an ancestor command, or a callback command.
API Endpoint	Required. API call to execute when the user runs the command.
Short Description	Required. Short description of the command.
Application	Read-only application scope for the command.
Active	When selected, the command is active.
Is Callback Command	When true, designates the command as a callback command. Select this option to hide the command from the CLI client and prevent users from calling it from the command line. Use this field with the **Callback** section of a primary command. The callback command executes when the primary command is complete.
Help Text	

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Description	Description of the command used as help text when the user runs the `--help` command.
Examples	Examples of the command used as help text when the user runs the `--help` command.
Expressions	
Success Expression	Expression used to evaluate the response from the server and determine if the command succeeded. For example, `result.code = 1`.
Failure Expression	Expression used to evaluate the response from the server and determine if the command failed.
Messages	
Success Message	Message displayed on the CLI client when the command is successful.
Progress Message	Message displayed on the CLI client when the command is in progress.
Failure Message	Message displayed on the CLI client when the command fails.
Callback	
Callback Expression	Expression used to determine whether to execute the callback command. For example, you can write an expression that checks on a long-running process. If the expression produces a certain result, the callback executes.
Callback Command	Command to execute when the **Callback Expression** is satisfied. Must be a command with the **Is Callback Command** field selected.
Callback Interval	The interval between callback command executions. Unit: Milliseconds Default: 1,000
Max Retries	Maximum number of times the callback command executes. Default: 10

c. In the Command Arguments related lists, create any command arguments needed for the command.

Create a command argument to allow users to set options when running a command.

Field	Description
Name	Required. Name of the command argument, for example `data`.
Short Name	Short dame of the command argument, for example `d`.
Data Type	Required. Type of data expected for the command argument. Options include:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
	▪ String: Allows users to enter a string as input. ▪ Integer: Allows users to enter a number as input. ▪ Boolean: Allows users to enter `true` or `false` as input. ▪ File Input: Allows users to upload a file as input. Map this argument to a Body **Parameter Type** in the API Endpoint Arguments related list in the next step. Users can pass plain text, YAML, JSON, or another file type accepted by the REST API. When the user submits a YAML file, the CLI client converts to file to JSON format by default. Use the **Skip Pre-processing** field to disable this behavior. ▪ Password: Allows users to securely enter a password as input. **Note:** For File Input data types, the file size limit is 10 MB by default. However, you change this limit using the *`glide.rest.scripted.max_inbound_content_length_mb`* system property.
Skip Pre-processing	Set this flag to prevent the CLI client from converting YAML files into JSON format before sending them in the request. When **Data Type** is set to **File Input** and the user submits a YAML file, the CLI client converts the file to JSON before executing the command by default. **Note:** Only applies when **Data Type** is set to **File Input**.
Default Value	Default value to use when the user does not pass a value.
Short Description	Required. Short description of the command argument. Used as help text when the user runs the `--help` command.
Application	Read-only application scope for the argument.
Command	Required. Command that the argument applies to.
Mandatory	When selected, the user must provide a value for the argument when running the associated command.
Prompt	Prompt to request information from the user. The CLI prompts for information when the user does not include a required argument in a command.
Visibility Expression	Expression used to determine whether the CLI should prompt for the argument. Typically used to display an argument based on the value provided to a previous argument.
Order	Order in which to prompt for the argument.

d. In the API Endpoint Arguments related lists, create any API endpoint arguments needed for the command.

Map command arguments from the Command Arguments related list to parameters in your REST endpoint.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Name	Required. Name of the argument.
Value	Required. The value from the command that you want to pass to the REST endpoint. You can pass a static value, or an expression such as `{flags.table}`. Use the `flags` global variable to access the command arguments.
Parameter Type	Required. The type of parameter in the REST endpoint that you want to pass the **Value** to. For example, if you select **Body**, the CLI passes the value of the **Value** field to the REST endpoint body. Options include: ▪ Body ▪ Header ▪ Path ▪ Query
Application	Read-only application scope for the argument.
Command	Required. Command that the argument applies to.

e. In the Return Values related lists, create any return values needed for the command.

Create return values to only return certain keys from the response.

Field	Description
Path Expression	Required. Expression representing the path to the key that you want to return.
Alias	Variable name to assign the return value to.
Application	Read-only application scope for the command.
Command	Required. Command that the return value applies to.

Result

When the user runs the ServiceNow CLI command, the system executes the associated REST API call and returns the result to the ServiceNow CLI.

Manage ServiceNow CLI extensions

Add extensions to the ServiceNow CLI to load additional functionality and commands, update existing extensions, or remove extensions you no longer need.

Before you begin

Role required: none.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Find available extensions.

   a. Open your system's command-line tool and execute this command.

   ```
   $ snc extension list-available -o table
   ```

   The system lists available extensions and associated details.

   ```
   NAME       DESCRIPTION
    VERSION            INSTALLED
   ------------------------------------------------------------------
   -------------------------
   ui-component        Build and deploy components
       19.0.0-alpha.15     false
   ```

2. Install an extension.

   a. Open your system's command-line tool and execute this command.

   ```
   $ snc extension add --name <extension-name>
   ```

3. **Optional:** Update an extension.

   a. Open your system's command-line tool and execute this command.

   ```
   $ snc extension update --name <extension-name>
   ```

4. **Optional:** Remove an extension.

   a. Open your system's command-line tool and execute this command.

   ```
   $ snc extension remove --name <extension-name>
   ```

ServiceNow CLI available commands

Commands and command arguments available to the base system ServiceNow CLI.

Global command arguments

Use command arguments to set options for any CLI commands.

--help

Provides help information for the specified command, which includes the description, supported arguments, and examples. For more information, see Get help with ServiceNow CLI.

```
$ snc --help
```

--debug

provides debug logging output when executing a command. For more information, see Get help with ServiceNow CLI.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
$ snc record delete --table incident --sysid
 552c48888c033300964f4932b03eb092 --debug
```

--profile

Specifies the named profile to use for a command. For more information, see Configuring and managing your ServiceNow CLI connection profiles.

```
$ snc configure profile set --profile <profilename>
```

--output

Specifies the output format to use for a command. The ServiceNow CLI supports the following output formats.

- `json`: The output is formatted as JSON. This is the default.

```
{
   "default": {
      "appversion": "1.0.8",
      "host": "https://myinstance.service-now.com",
      "hostversion": "Paris",
      "loginmethod": "basic",
      "output": "json",
      "username": "admin"
   },
   "user1": {
      "appversion": "1.0.8",
      "host": "https://otherinstance.service-now.com",
      "hostversion": "Paris",
      "loginmethod": "basic",
      "output": "yaml",
      "username": "admin"
   }
}
```

- `yaml`: The output is formatted as YAML. Use YAML to handle the output with services and tools that emit or consume YAML-formatted strings.

```
default:
   appversion: 1.0.8
   host: https://myinstance.service-now.com
   hostversion: Paris
   loginmethod: basic
   output: json
   username: admin
user1:
   appversion: 1.0.8
   host: https://otherinstance.service-now.com
   hostversion: Paris
   loginmethod: basic
   output: yaml
   username: admin
```

- `text`: The output is formatted as multiple lines of tab-separated string values. Use this output with traditional UNIX text tools such as grep, sed, and awk, and the text processing performed by PowerShell.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
default   https://myinstance.service-now.com   Paris
 1.0.8   basic   admin   json
user1   https://otherinstance.service-now.com   Paris
  1.0.8   basic   admin   yaml
```

- `table`: The output is formatted as a table which presents the information in a human-readable format.

```
NAME   HOST   HOST   VERSION   APP   VERSION   LOGIN
 METHOD   USERNAME   OUTPUT
-------------------------------------------------------
-------------------------
default   myinstance   Paris   1.0.8   basic   admin
 json
user1   otherinstance   Paris   1.0.8   basic   admin
  yaml
```

- `none`: The CLI does not print the output to the console. Success, error, and progress messages still display.

```
$ snc record query --table incident --query
 'active=true' --output json
```

--no-interactive

Prevents the CLI from prompting the user for argument values. If the user does not pass a value for a required argument, the system uses the default value. If no default value is defined, the system throws an error.

--no-verbose

The command executes silently without messages. Use this argument in automated testing scenarios.

Configure profile

Create a connection profile to connect with your instance, view connection profiles, refresh your connection and available commands, or delete profiles you no longer need. For more information about configuring your profile, see Configuring and managing your ServiceNow CLI connection profiles.

Set up a default profile

Create a connection profile that the ServiceNow CLI uses by default. You must create a default profile to set up the CLI's initial connection with an instance.

```
$ snc configure profile set
```

The system prompts you for the following information:

Requested information	Description
Host	The host name of the instance to connect to. Supports both the full URL (https://my-instance.service-now.com) or just the host name (my-instance).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Requested information	Description
Login method	The login method to use to connect to the instance. Supports `Basic`, `OAuth`, and `OAuth + MFA`.
Username	The user name to use to connect to the instance.
Password	The password to use to connect to the instance.
Client id	The client ID to use to connect to the instance when the login method is OAuth or OAuth + MFA.
Client secret	The client secret to use to connect to the instance when the login method is OAuth or OAuth + MFA.
Authentication code	The authentication code to use to connect to the instance when the login method is OAuth + MFA.
Default output format	Specifies how to format the command results. Options are `json`, `yaml`, `text`, and `table`.

Set up a named profile

Create a named connection profile to use with specific commands. This allows you to specify a different instance or connection protocol for a specific command.

```
$ snc configure profile set [--profile profile-name]
```

The system prompts you for the following information:

Requested information	Description
Host	The host name of the instance to connect to. Supports both the full URL (https://my-instance.service-now.com) or just the host name (my-instance).
Login method	The login method to use to connect to the instance. Supports `Basic`, `OAuth`, and `OAuth + MFA`.
Username	The user name to use to connect to the instance.
Password	The password to use to connect to the instance.
Client id	The client ID to use to connect to the instance when the login method is OAuth or OAuth + MFA.
Client secret	The client secret to use to connect to the instance when the login method is OAuth or OAuth + MFA.
Authentication code	The authentication code to use to connect to the instance when the login method is OAuth + MFA.
Default output format	Specifies how to format the command results. Options are `json`, `yaml`, `text`, and `table`.

View profiles

View all connection profiles set in the configuration file, or view information about a specific profile.

```
$ snc configure profile list [--profile profile-name]
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Remove a profile

Remove a named connection profile that you no longer need from the configuration file.

```
$ snc configure profile remove [--profile profile-name]
```

Refresh a profile

Update the available commands from the instance for the given profile. Refresh your connection after modifying any of the commands on the corresponding instance in order to keep the CLI up-to-date.

```
$ snc configure profile refresh [--profile
 profile-name]
```

Perform record operations

Create, read, update, delete, and query records in your instance using the ServiceNow CLI command-line tool. For more information about performing record operations, see Perform record operations using ServiceNow CLI.

Create a record

Inserts a single record in a specified table.

```
$ snc record create [--table table --data data]
```

Pass in values for these arguments.

Parameter	Description
table	Required. Name of the table in which to save the record.
data	Required. Field name and the associated value for each field to define in the specified record in JSON string format.

Delete a record

Deletes the specified record from the specified table.

```
$ snc record delete [--table table --sysid sys_id]
```

Pass in values for these arguments.

Parameter	Description
table	Required. Name of the table in which to delete the record.
sysid	Required. Sys_id of the record to delete.

Get a record

Retrieves a single record based on the specified sys_id from the specified table.

```
$ snc record get [--table table --sysid sys_id]
```

Pass in values for these arguments.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Parameter	Description
table	Required. Name of the table from which to retrieve the record.
sysid	Required. Sys_id of the record to retrieve.

Query records

Retrieves multiple records from a specified table.

```
$ snc record query [--displayvalue displayValue
 --fields fields --limit limit --offset offset --query
 query --table table]
```

Pass in values for these arguments.

Parameter	Description
displayValue	Include `--displayvalue` to retrieve the display value from the database for reference and choice fields. Do not include this parameter to retrieve the actual values.
fields	Comma-separated list of field names to return from the database.
limit	Maximum number of records to return.
offset	Starting record index for which to begin retrieving records. Use this value to paginate record retrieval.
query	Required. Encoded query used to filter the result set in the following format: `--query '<column_name><operator><value>'`.
table	Required. Name of the table in which to query the records.

Update a record

Updates the specified record with the given data attributes.

```
$ snc record update [--sysid sys_id --table table
 --data data]
```

Pass in values for these arguments.

Parameter	Description
table	Required. Name of the table in which to save the record.
sysid	Required. Sys_id of the record to update.
data	Required. Field name and the associated value for each field to define in the specified record in JSON string format.

Work with extensions

Add extensions to the ServiceNow CLI to load additional functionality and commands, update existing extensions, or remove extensions you no longer need. For more information, see Manage ServiceNow CLI extensions.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Find available extensions

```
$ snc extension list-available -o table
```

Install an extension

```
$ snc extension add --name <extension-name>
```

Update an extension

```
$ snc extension update --name <extension-name>
```

Remove an extension

```
$ snc extension remove --name <extension-name>
```

Use the ui-component extension

Add the ui-component extension

Add the ui-component extension to the ServiceNow CLI.

```
$ snc extension add --name ui-component
```

Set up your project

Create the component project and the set of files required to develop a component. You can connect to your instance and create an application scope for your component, or you can reserve a scope to verify later.

```
$ snc ui-component project [--name name --description
 description --scope scope --offline]
```

Pass in values for these arguments.

Name	Description
name	Required. The project name. Must be a valid and unique npm package name.
description	The project description to be available in the npm registry and the plugins list in your instance.
scope	Suggested application scope to assign to this project and its components. If provided, the instance validates the name. Use the namespace identifier guidelines for application development on the instance. For more information, see Application scope. Maximum: 18 characters. Case: snake case. Default: `x_customerprefix_componentname`, where: • `customerprefix` is the value in the *`glide.appcreator.company.code`* system property on your instance. • `componentname` is the value provided in the component's `name` parameter when you created the project.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Name	Description
	Alternatively, you can add a value to the `scopeName` parameter in the `now-ui.json` file.
offline	When true, creates and scaffolds a component while disconnected from your instance. Skips validation of the given scope name. Default: `false`.

Run the development server

Add your component code and test it using a local development server.

```
$ snc ui-component develop [--entry entry --open --port
 port --host host]
```

Pass in values for these arguments.

Name	Description
entry	Path to the test module in your component project. Default: `example/index.js`.
open	Opens the default browser and navigates to the test page. Default: false.
port	Port where the development server runs. Default: 8081.
host	Host address to use if you want your local development server to be accessible externally by others. Typically set to `0.0.0.0`

Deploy a component to an instance

Deploy your component to display in your instance as an application plugin.

```
$ snc ui-component deploy [--open --force]
```

Pass in values for these arguments.

Name	Description
open	When true, opens the default browser and navigates to UI Builder in your instance. Default: false
force	Deploys component changes and overwrites any existing component records. Default: false.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Commands installed with CMDB Application CLI and API

Commands and command groups available to the ServiceNow CLI when the app-cmdb-api-cli plugin is installed.

Request apps on the Store

Visit the ServiceNow Store to view all the available apps, and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the ServiceNow Store version history release notes.

To script critical operations which support automation across the enterprise, you can leverage APIs or run command line operations that the CMDB Application CLI and API store app provide instead of using the user interface. The CMDB Application CLI and API store app provides a robust framework which consolidates all the APIs that are related to application services and the command lines that let you access the interface to those APIs.

CMDB Application CLI and API commands enable the following tasks:

- Registering and creating an application service and establishing upstream relationships
- Getting details of a given application service and its upstream relationships
- Connecting higher level constructs such as business applications and business service offerings
- Populating an application service with a given population type
- Changing the state of an application service

For the REST API solution, see SG Services API.

Convert Application Service

Converts a manual or empty type application service to a calculated application service. During conversion, the application service record moves into the [cmdb_ci_service_calculated] table with the newly assigned class.

Command group:

- Parent group: service-graph
- Child group: app-service

Roles required

app_service_admin

If using a service mapping related service, the user must have the required roles for that service mapping related service.

Command structure for Mac OS

```
snc service-graph app-service convert --data '{JSON}'
```

Command structure for Windows OS

```
snc service-graph app-service convert --data "{JSON}"
```

Arguments

This command passes a JSON object using the *data* parameter.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The following properties for identifying a CI take precedence as follows:

1. sys_id – If sys_id, the system only uses the sys_id and ignores any additional values.
2. number – If provided without the sys_id, the system only uses the number and ignores any additional values.
3. <IRE field name> – The system only uses these values if the sys_id or number are not provided.

JSON object properties

Name	Description
<IRE field name>	One or more IRE fields identifying the application service. For example, name or version. Data type: String
levels	Number of levels to include in the conversion. Data type: String
number	Unique number that identifies the application service. Data type: String
sys_id	Sys_id of the application service. Data type: String

Example on Mac OS

```
snc service-graph app-service
 convert --data '{"name":"Test
 Register","environment":"Test","version":"1.0","levels
":8}'
```

Example on Windows OS

```
snc service-graph app-service
 convert --data "{"name":"Test
 Register","environment":"Test","version":"1.0","levels
":8}"
```

Return value

```
{
  "result": {
  "status": "success"
  }
}
```

Create Application Service Relationship

Constructs upstream relations such as business applications, business service offerings, and other application services. Running this command creates a relationship, taking input with a single parent and a corresponding child object.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Command group:

- Parent group: service-graph
- Child group: app-service

Roles required

app_service_admin

If using a service mapping related service, the user must have the required roles for that service mapping related service.

Command structure for Mac OS

```
snc service-graph app-service create-relationship
 --data '{JSON}'
```

Command structure for Windows OS

```
snc service-graph app-service create-relationship
 --data "{JSON}"
```

Arguments

This command passes a JSON object using the *data* parameter.

The following properties for identifying a CI take precedence as follows:

1. sys_id – If sys_id, the system only uses the sys_id and ignores any additional values.
2. number – If provided without the sys_id, the system only uses the number and ignores any additional values.
3. <IRE field name> – The system only uses these values if the sys_id or number are not provided.

JSON object properties

Name	Description
child	Information identifying the child application service with which to create a relationship. The child is located in the Service Instance [cmdb_ci_service_auto] table. A dynamic CI group can be added as a child but cannot be parent. Data type: Object `"child": { "<IRE field name>": "String", "number": "String", "sys_id": "String" }`
child.<IRE field name>	One or more IRE fields identifying the child application service. For example, name or version. Data type: String

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

JSON object properties (continued)

Name	Description
child.number	Unique number that identifies the child application service. Data type: String
child.sys_id	Sys_id of the child application service listed in the Service Instance [cmdb_ci_service_auto]. Data type: String
parent	Details identifying the parent application service with which to create a relationship. Data type: Object `"parent": {` `  "<IRE field name>": "String",` `  "number": "String",` `  "sys_id": "String",` `  "class_name": "String"` `}`
parent.<IRE field name>	One or more IRE fields identifying the application service. For example, name or version. Data type: String
parent.number	Unique number that identifies the application service. Data type: String
parent.sys_id	Sys_id of the application service listed in the Service Instance [cmdb_ci_service_auto]. Data type: String
parent.class_name	Name of the class that contains the application service. The parent class name should be from one of the following tables: • cmdb_ci_service_auto • cmdb_ci_service_discovered • cmdb_ci_service_by_tags • cmdb_ci_service_calculated • service_offering • cmdb_ci_business_app Default: cmdb_ci_service_auto Data type: String

Example on Mac OS

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
snc service-graph app-service
 create-relationship --data
 '{"child":{"name":"wdfsdf","environment":"Test","versi
on":"1.0"},"parent":{"sys_id":"abcdefg","name":"busin
ess App1","class_name":"service_offering"}}'
```

Example on Windows OS

```
snc service-graph app-service
 create-relationship --data
 "{"child":{"name":"wdfsdf","environment":"Test","versi
on":"1.0"},"parent":{"sys_id":"abcdefg","name":"busin
ess App1","class_name":"service_offering"}}"
```

Return value

```
{
  "result": {
  "status": "success"
  }
}
```

Delete Application Service

Deletes an application service.

Command group:

- Parent group: service-graph
- Child group: app-service

Roles required

app_service_admin

If using a service mapping related service, the user must have the required roles for that service mapping related service.

Command structure for Mac OS

```
snc service-graph app-service delete --data '{JSON}'
```

Command structure for Windows OS

```
snc service-graph app-service delete --data "{JSON}"
```

Arguments

This command passes a JSON object using the *data* parameter.

The following properties for identifying a CI take precedence as follows:

1. sys_id – If sys_id, the system only uses the sys_id and ignores any additional values.
2. number – If provided without the sys_id, the system only uses the number and ignores any additional values.
3. <IRE field name> – The system only uses these values if the sys_id or number are not provided.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

JSON object properties

Name	Description
<IRE field name>	One or more IRE fields identifying the application service. For example, name or version. Data type: String
number	Unique number that identifies the application service. Data type: String
sys_id	Sys_id of the application service listed in the Service Instance [cmdb_ci_service_auto] table. Data type: String

Example on Mac OS

```
snc service-graph app-service
 delete --data '{"name":"Test
 Register","environment":"Test","version":"1.0"}'
```

Example on Windows OS

```
snc service-graph app-service
 delete --data "{"name":"Test
 Register","environment":"Test","version":"1.0"}"
```

Return value

```
{
  "result": {
  "status": "success"
  }
}
```

Delete Application Service Relationship

Deletes an application service upstream relationship.

Command group:

- Parent group: service-graph
- Child group: app-service

Roles required

app_service_admin

If using a service mapping related service, the user must have the required roles for that service mapping related service.

Command structure for Mac OS

```
snc service-graph app-service delete-relationship
 --data '{JSON}'
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Command structure for Windows OS

```
snc service-graph app-service delete-relationship
 --data "{JSON}"
```

Arguments

This command passes a JSON object using the *data* parameter.

The following properties for identifying a CI take precedence as follows:

1. sys_id – If sys_id, the system only uses the sys_id and ignores any additional values.
2. number – If provided without the sys_id, the system only uses the number and ignores any additional values.
3. <IRE field name> – The system only uses these values if the sys_id or number are not provided.

JSON object properties

Name	Description
child	Information describing the child relationship to be deleted from the service application. Data type: Object `"child": { "<IRE field name>": "String", "number": "String", "sys_id": "String" }`
child.<IRE field name>	One or more IRE fields identifying the child application service. For example, name or version. Data type: String
child.number	Unique number that identifies the child application service. Data type: String
child.sys_id	Sys_id of the child application service listed in the Service Instance [cmdb_ci_service_auto]. Data type: String
parent	Details identifying the parent application service from which to remove a relationship. Data type: Object `"parent": { "<IRE field name>": "String", "number": "String", "sys_id": "String", "class_name": "String" }`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

JSON object properties (continued)

Name	Description
parent.<IRE field name>	One or more IRE fields identifying the application service. For example, name or version. Data type: String
parent.number	Unique number that identifies the application service. Data type: String
parent.sys_id	Sys_id of the application service listed in the Service Instance [cmdb_ci_service_auto] table. Data type: String
parent.class_name	Name of the class that contains the application service. The parent class name should be from one of the following tables: • cmdb_ci_service_auto • cmdb_ci_service_discovered • cmdb_ci_service_by_tags • cmdb_ci_service_calculated • service_offering • cmdb_ci_business_app Default: cmdb_ci_service_auto Data type: String

Example on Mac OS

```
snc service-graph app-service delete-relationship
 --data '{"child":{"name":"Test
 Register","environment":"Test","version":"1.0"},"paren
t":{"name":"business Service
 Offering1","class_name":"service_offering"}}'
```

Example on Windows OS

```
snc service-graph app-service delete-relationship
 --data "{"child":{"name":"Test
 Register","environment":"Test","version":"1.0"},"paren
t":{"name":"business Service
 Offering1","class_name":"service_offering"}}"
```

Return value

```
{
  "result": {
  "status": "success"
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    }
}
```

Find Application Service

Finds the details of a given application service and its upstream relationships.

Command group:

- Parent group: service-graph
- Child group: app-service

Roles required

app_service_admin – This role provides unlimited viewing of application services.

app_service_user – This role only provides viewing application services in Operational status.

If using a service mapping related service, the user must have the required roles for that service mapping related service.

Command structure for Mac OS

```
snc service-graph app-service find --data '{JSON}'
```

Command structure for Windows OS

```
snc service-graph app-service find --data "{JSON}"
```

Arguments

This command passes a JSON object using the *data* parameter.

The following properties for identifying a CI take precedence as follows:

1. sys_id – If sys_id, the system only uses the sys_id and ignores any additional values.
2. number – If provided without the sys_id, the system only uses the number and ignores any additional values.
3. <IRE field name> – The system only uses these values if the sys_id or number are not provided.

JSON object properties

Name	Description
<IRE field name>	One or more IRE fields identifying the application service. For example, name or version. Data type: String
number	Unique number that identifies the application service. Data type: String
sys_id	Sys_id of the application service listed in the Service Instance [cmdb_ci_service_auto] table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

JSON object properties (continued)

Name	Description
	Data type: String

Example on Mac OS

```
snc service-graph app-service find --data '{"name" :
 "Test App Service1"}'
```

Example on Windows OS

```
snc service-graph app-service find --data "{"name" :
 "Test App Service1"}"
```

Return value


```
{
  "result": {
    "aliases": null,
    "asset": null,
    "asset_tag": null,
    "assigned": "",
    "assigned_to": null,
    "assignment_group": null,
    "attestation_score": null,
    "attested": "0",
    "attested_by": null,
    "attested_date": "",
    "attributes": null,
    "bucket": null,
    "business_contact": null,
    "business_need": null,
    "business_relation_manager": null,
    "business_unit": null,
    "busines_criticality": "4 - not critical",
    "can_print": "0",
    "category": null,
    "change_control": null,
    "checked_in": "",
    "checked_out": "",
    "checkout": null,
    "comments": null,
    "company": null,
    "compatibility_dependencies": null,
    "consumer_type": "internal",
    "correlation_id": null,
    "cost": null,
    "cost_cc": "USD",
    "cost_center": null,
    "delivery_date": "",
    "delivery_manager": null,
    "department": null,
    "discovery_source": "Manual Entry",
    "dns_domain": null,
    "due": "",
```


© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    "due_in": null,
    "duplicate_of": null,
    "end_date": "",
    "environment": null,
    "fault_count": "0",
    "first_discovered": "2021-07-19 20:09:48",
    "fqdn": null,
    "gl_account": null,
    "hide_from_dashboard": "0",
    "install_date": "",
    "install_status": "1",
    "invoice_number": null,
    "ip_address": null,
    "justification": null,
    "last_discovered": "2021-07-19 20:09:48",
    "last_review_date": "",
    "layer": null,
    "lease_id": null,
    "life_cycle_stage": null,
    "life_cycle_stage_status": null,
    "location": null,
    "mac_address": null,
    "maintenance_schedule": null,
    "managed_by": null,
    "managed_by_group": null,
    "manufacturer": null,
    "model_id": null,
    "model_number": null,
    "monitor": "0",
    "monitoring_requirements": null,
    "name": "Test App Service1",
    "number": "SNSVC0001014",
    "operational_status": "2",
    "order_date": "",
    "owned_by": null,
    "parent": null,
    "portfolio_status": "pipeline",
    "po_number": null,
    "prerequisites": null,
    "price_model": "per_unit",
    "price_unit": null,
    "published_ref": null,
    "purchase_date": "",
    "schedule": null,
    "serial_number": null,
    "service_classification": "Application Service",
    "service_level_requirement": null,
    "service_owner_delegate": null,
    "service_status": "requirements",
    "severity": null,
    "short_description": null,
    "skip_sync": "0",
    "sla": null,
    "spm_service_portfolio": null,
    "spm_taxonomy_node": null,
    "stakeholders": null,
    "start_date": "",
```


© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    "state": null,
    "subcategory": null,
    "supported_by": null,
    "support_group": null,
    "sys_class_name": "cmdb_ci_service_auto",
    "sys_class_path": "/!!/!7/!(",
    "sys_created_by": "admin",
    "sys_created_on": "2021-07-19 20:09:48",
    "sys_domain": "global",
    "sys_domain_path": "/",
    "sys_id": "a2f0618040697410f87713b656474255",
    "sys_mod_count": "0",
    "sys_updated_by": "admin",
    "sys_updated_on": "2021-07-19 20:09:48",
    "unit_description": null,
    "unverified": "0",
    "used_for": "Production",
    "user_group": null,
    "vendor": null,
    "version": null,
    "view_service": "61e1cb757f23220002d31ccebefa9120",
    "warranty_expiration": "",
    "relationships": [
      {
        "name": "Test Biz App1",
        "sys_id": "0250a94040697410f87713b656474250",
        "number": "APM0001001",
        "class_name": "cmdb_ci_business_app",
        "relationship": "Consumes::Consumed by"
      },
      {
        "name": "Tech Service Offering1",
        "sys_id": "98d0ed4040697410f87713b6564742ef",
        "number": "BSN0001005",
        "class_name": "service_offering",
        "relationship": "Contains::Contained by"
      }
    ]
  }
}
```


Populate Application Service

Populates an application service with a service population method.

Command group:

- Parent group: service-graph
- Child group: app-service

Roles required

app_service_admin

If using a service mapping related service, the user must have the required roles for that service mapping related service.

Command structure for Mac OS

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
snc service-graph app-service populate --data '{JSON}'
```

Command structure for Windows OS

```
snc service-graph app-service populate --data "{JSON}"
```

Arguments

This command passes a JSON object using the *data* parameter.

The following properties for identifying a CI take precedence as follows:

1. sys_id – If sys_id, the system only uses the sys_id and ignores any additional values.
2. number – If provided without the sys_id, the system only uses the number and ignores any additional values.
3. <IRE field name> – The system only uses these values if the sys_id or number are not provided.

JSON object properties

Name	Description
<IRE field name>	One or more IRE fields identifying the application service. For example, name or version. Data type: String
number	Unique number that identifies the application service. Data type: String
population_method	Required. Identifies the population method and its accompanying property to identify the content for population. Only one accompanying object is valid per type. Data type: Object
population_method.group_id	Group ID of the CMDB group configured with the cmdb_group population type. Data type: "String" `"population_method": { "group_id": "String", "type": "cmdb_group" }` Associated population type: cmdb_group
population_method.levels	Number of levels to use in building the service. If the level value is

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

JSON object properties (continued)

Name	Description
	not provided, the system checks the sys_property for the value. If svc.manual.convert.levels.default_value is not populated, a default value of 3 is used. Data type: Number `"population_method": { "levels": Number, "type": "dynamic_service" }` Associated population type: dynamic_service Default: 3 if no level value is set for the sys_property
population_method.service_candidate	Unique identifier of the service candidate. Data type: String `"population_method": { "service_candidate": "String", "type": "tag_based_service_family" }` Associated population type: tag_based_service_family
population_method.service_relations	List of objects containing hierarchy data for the CIs within the application service. All CIs form pairs with a parent and child CI. The top-level CI, referred to as the entry point of an application service, does not have a parent CI. Data type: Array `"population_method": { "service_relations":[ { "child": "String", "parent": "String" } ],`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

JSON object properties (continued)

Name	Description
	<pre>"type": "service_hierarchy" }</pre> Associated population type: service_hierarchy
population_method.service_relations.child	Name of a child CI related to the CI. Data type: String
population_method.service_relations.parent	Name of a parent CI related to the CI. Data type: String
population_method.tags	List of objects containing tags to associate with the CI. This information is located in the Key Values [cmdb_key_value] table. Data type: Array <pre>"population_method": { "tags": [{ "tag": "String", "value": "String" }], "type": "tag_list" }</pre> Associated population type: tag_list
population_method.tags.tag	Tag name. Data type: String
population_method.tags.value	Tag value. Data type: String
population_method.type	Required. Population type to add to the application service. Data type: Object Valid values: • cmdb_group • service_hierarchy • dynamic_service

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

JSON object properties (continued)

Name	Description
	• tag_list • tag_based_service_family
sys_id	Sys_id of the application service listed in the Service Instance [cmdb_ci_service_auto] table. Data type: String

Example on Mac OS

```
snc service-graph app-service
 populate --data '{"name":"Test
 Register","environment":"Test","version":"1.0","popula
tion_method":{"group_id":"String","type":"cmdb_group"
}}'
```

Example on Windows OS

```
snc service-graph app-service
 populate --data "{"name":"Test
 Register","environment":"Test","version":"1.0","popula
tion_method":{"group_id":"String","type":"cmdb_group"
}}"
```

Return value

```
{
  "result": {
  "status": "success"
  }
}
```

Register Application Service

Creates an application service, tags and constructs upstream relationships such as business applications, business service offerings, and other application services.

Command group:

- Parent group: service-graph
- Child group: app-service

Roles required

app_service_admin

If using a service mapping related service, the user must have the required roles for that service mapping related service.

Command structure for Mac OS

```
snc service-graph app-service register --data '{JSON}'
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Command structure for Windows OS

```
snc service-graph app-service register --data "{JSON}"
```

Arguments

This command passes a JSON object using the *data* parameter.

The following properties for identifying a CI take precedence as follows:

1. sys_id – If sys_id, the system only uses the sys_id and ignores any additional values.
2. number – If provided without the sys_id, the system only uses the number and ignores any additional values.
3. <IRE field name> – The system only uses these values if the sys_id or number are not provided.

JSON object properties

Name	Description
<IRE field name>	One or more IRE fields identifying the application service. For example, name or version. Data type: String
number	Unique number that identifies the application service. Data type: String
relationships	Upstream relationships categorized by type. Data type: Object `"relationships": { "business_app": [Array], "business_service_offering": [Array], "parent_app_service": [Array], "technical_service_offering": [Array] }` Maximum number of relationships is 25.
relationships.business_app	List of objects representing Business Application relationship types. These values can be defined using one of the following items as key-value pairs. • <IRE field name> • number • sys_id

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

JSON object properties (continued)

Name	Description
	Data type: Array
relationships.business_service_offering	List of objects representing Business Service Offering relationship types. These values can be defined using the following items as key-value pairs. • <IRE field name> • number • sys_id Data type: Array
relationships.parent_app_service	List of objects representing Application Service relationship types. These values can be defined using the following items as key-value pairs. • <IRE field name> • number • sys_id Data type: Array
relationships.technical_service_offering	List of objects representing technology management offering (formerly technical service offering)relationship types. These values can be defined using the following items as key-value pairs. • <IRE field name> • number • sys_id Data type: Array
sys_id	Sys_id of the application service listed in the Service Instance [cmdb_ci_service_auto] table. Data type: String
tags	List of objects containing tag definitions as key-value pairs. `"tags": [ { "key": "String", "value": "String" }]`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

JSON object properties (continued)

Name	Description
	Data type: Array
tags.key	Tag category name. Data type: String
tags.value	Tag value. Data type: String

Example on Mac OS

```
snc service-graph app-service
 register --data '{"name":"Test
 Register","environment":"Test","version":"1.0","number
":"
 SNSVC0001014","relationships":{"business_applicatio
n":[{"sys_id":"0250a94040697410f87713b656474250"},{"num
ber":"APM0001002"},{"name":"Test Biz
 App1"}],"business_service_offering":[{"sys_id":"ed32e9
8040697410f87713b656474259"}],"technical_service_offeri
ng":[{"sys_id":"80e12d8040697410f87713b65647421c"},{"nu
mber":"BSN0001005"},{"name":"Tech Service
 Offering2"}],"parent_app_service":[{"sys_id":"a2f06180
40697410f87713b656474255"}]},"tags":[{"key":"key1","val
ue":"value1"},{"key":"key2","value":"value2"}]}'
```

Example on Windows OS

```
snc service-graph app-service
 register --data "{"name":"Test
 Register","environment":"Test","version":"1.0","number
":"
 SNSVC0001014","relationships":{"business_applicatio
n":[{"sys_id":"0250a94040697410f87713b656474250"},{"num
ber":"APM0001002"},{"name":"Test Biz
 App1"}],"business_service_offering":[{"sys_id":"ed32e9
8040697410f87713b656474259"}],"technical_service_offeri
ng":[{"sys_id":"80e12d8040697410f87713b65647421c"},{"nu
mber":"BSN0001005"},{"name":"Tech Service
 Offering2"}],"parent_app_service":[{"sys_id":"a2f06180
40697410f87713b656474255"}]},"tags":[{"key":"key1","val
ue":"value1"},{"key":"key2","value":"value2"}]}"
```

Return value

```
{
  "result": {
    "app_service": {
      "sys_id": "99b2a54040697410f87713b6564742ad",
      "name": "Test Register",
      "number": "SNSVC0001014"
    },
    "message": "Service registered successfully",
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    "status": "INSERT"
  }
}
```

Update Application Service

Updates an existing application service provided and creates tags for the given application service.

Command group:

- Parent group: service-graph
- Child group: app-service

Roles required

app_service_admin

If using a service mapping related service, the user must have the required roles for that service mapping related service.

Command structure for Mac OS

```
snc service-graph app-service update --data '{JSON}'
```

Command structure for Windows OS

```
snc service-graph app-service update --data "{JSON}"
```

Arguments

This command passes a JSON object using the *data* parameter.

The following properties for identifying a CI take precedence as follows:

1. sys_id – If sys_id, the system only uses the sys_id and ignores any additional values.
2. number – If provided without the sys_id, the system only uses the number and ignores any additional values.
3. <IRE field name> – The system only uses these values if the sys_id or number are not provided.

JSON object properties

Name	Description
<fields or tags to update>	Use key-value pairs to identify each field or tag to be updated. Only basic information can be updated, no upstream relationships can be updated. Data type: String
<IRE field name>	One or more IRE fields identifying the application service. For example, name or version. Data type: String
number	Unique number that identifies the application service.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

JSON object properties (continued)

Name	Description
	Data type: String
sys_id	Sys_id of the application service listed in the Service Instance [cmdb_ci_service_auto]. Data type: String

Example on Mac OS

```
snc service-graph app-service update --data '{"name":
 "Test Register", "version": "2.0"}'
```

Example on Windows OS

```
snc service-graph app-service update --data "{"name":
 "Test Register", "version": "2.0"}"
```

Return value

```
{
  "result": {
    "sys_id": "99b2a5404069741 0f87713b6564742ad",
    "name": "Test Register",
    "number": "SNSVC0001014",
    "version": "2.0"
  }
}
```

Update Application Service State

Changes the application service lifecycle state to activate, deactivate, or retire.

Command group:

- Parent group: service-graph
- Child group: app-service

Roles required

app_service_admin

If using a service mapping related service, the user must have the required roles for that service mapping related service.

Command structure for Mac OS

```
snc service-graph app-service update-state --data
 '{JSON}'
```

Command structure for Windows OS

```
snc service-graph app-service update-state --data
 "{JSON}"
```

Arguments

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

This command passes a JSON object using the *data* parameter.

The following properties for identifying a CI take precedence as follows:

1. sys_id – If sys_id, the system only uses the sys_id and ignores any additional values.
2. number – If provided without the sys_id, the system only uses the number and ignores any additional values.
3. <IRE field name> – The system only uses these values if the sys_id or number are not provided.

JSON object properties

Name	Description
<IRE field name>	One or more IRE fields identifying the application service. For example, name or version. Data type: String
number	Unique number that identifies the application service. Data type: String
state	Required. Lifecycle state of the application service. These values are updated in the Service Instance [cmdb_ci_service_auto] table. Valid values: • ACTIVATE – Life cycle is operational and in use. ○ operational_status=Operational ○ life_cycle_stage=Operational ○ life_cycle_stage_status=In Use • DEACTIVATE – Life cycle is not operational and is in the design stage. ○ operational_status=Non-Operational ○ life_cycle_stage=Design ○ life_cycle_stage_status=Build • RETIRE – End of life. ○ operational_status=Retired ○ life_cycle_stage=End Of Life ○ life_cycle_stage_status=Retired Data type: String
sys_id	Sys_id of the application service listed in the Service Instance [cmdb_ci_service_auto]. Data type: String

Example on Mac OS

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
snc service-graph app-service
 update-state --data '{"name":"Test
 Register","environment":"Test","version":"1.0"}'
```

Example on Windows OS

```
snc service-graph app-service
 update-state --data "{"name":"Test
 Register","environment":"Test","version":"1.0"}"
```

Return value

```
{
  "result": {
  "status": "success"
  }
}
```

Building pro-code applications

Create and manage custom applications from scratch using classic pro-code tools.

ServiceNow Studio

Use ServiceNow Studio to build apps and app files with integrated tools, access and edit metadata in scoped and global apps, and package app changes for deployment, all in one powerful development tool.

ServiceNow IDE

ServiceNow IDE is an implementation of Visual Studio Code for the Web on the ServiceNow AI Platform. With the ServiceNow IDE, you can create and develop scoped applications in source code, create JavaScript modules, and use third-party libraries.

ServiceNow SDK

ServiceNow SDK enables developers to create scoped applications in source code locally in Visual Studio Code Desktop and upload changes to a ServiceNow instance. You can also use the ServiceNow SDK to create JavaScript modules and use third-party libraries.

ServiceNow Extensions for Visual Studio Code

The ServiceNow extensions for VS Code editor enables you to edit applications offline and within ServiceNow instances.

Which app builder should I use?

Use the following table to choose the app building experience that matches your role and technical background.

App builders

Tool	Users	Features
Build Agent: Build an app using natural language prompts.	Developers of all experience levels (from low-code to Platform and source code)	Build full-stack applications via a conversational interface. You can use Build Agent in ServiceNow Studio and the

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

App builders (continued)

Tool	Users	Features
		ServiceNow IDE to generate a complete ServiceNow scoped or global application. For more information, see Exploring Build Agent.
ServiceNow Studio: Build and deliver apps in a unified environment.	Citizen developers, Platform developers	Build apps in a unified development environment. ServiceNow Studio provides streamlined navigation, integrated low-code tools, and built-in tracking and packaging so you can develop and ship apps faster. Use Build Agent in ServiceNow Studio to create and update apps with a conversational interaction. For more information, see Exploring ServiceNow Studio.

Scripting

Action Designer

Use Workflow Studio to automate a repeatable task within a flow as a sequence of related steps. Enable flow authors to add actions to multiple flows with minimal configuration.

Client-side scripting

Client scripts enable the system to run JavaScript on the client (web browser) when client-based events occur, such as when a form loads, after form submission, or when a field changes value.

ServiceNow Otto for Code

Now Assist for code generation allows developers to write scripts quickly with AI-generated code suggestions based on text or code prompts.

Now Code Editor

Now Code Editor provides a rich-text editor interface that supports Cascading Style Sheets (CSS), Hypertext Markup Language (HTML), JavaScript, Extensible Markup Language (XML), and JavaScript Object Notation (JSON). Use Now Code Editor to modify UI configuration, data resource configuration, styles, events, client-side and server-side scripts in Next Experience UI Builder components.

Server-side scripting

Server scripts run on the server or database. They can change the appearance or behavior of the ServiceNow AI Platform or run as business rules when records and tables are accessed or modified.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Core functionality

Groups

Simplify user administration by assigning roles to groups. Any business rules, assignment rules, system roles, or attributes that refer to the group apply to all group members automatically.

Notifications

Use ServiceNow Notifications to manage system email, create system notifications, and configure how your system responds to inbound email.

Roles

Roles control access to features and capabilities in applications and modules. The admin role provides access to all features and capabilities.

ServiceNow plugins

Plugins are software components that provide specific features and functionalities within a ServiceNow instance.

UI policies

UI policies dynamically change the behavior of information on a form and control custom process flows for tasks.

Users

User records establish a relationship between an individual and your ServiceNow instance. User records consist of a user name, a password, and information relating to the individual, such as contact information, location, and job title.

Related applications and features

JavaScript APIs

Use JavaScript APIs in scripts that you write to change the functionality of applications, or when you create new applications.

UI Builder

Use UI Builder to build web user interfaces for CRM Workspace, App Engine Studio generated workspaces and portals, or custom web experiences using Next Experience Components and custom web components.

Building applications in source code

Create and develop custom applications in source code using familiar development tools and processes.

Overview of development in source code

You can create applications in code using the ServiceNow IDE on the ServiceNow AI Platform or locally in Visual Studio Code Desktop with the ServiceNow SDK.

In either development environment, you use ServiceNow Fluent, a domain-specific programming language, to define the metadata that makes up applications. ServiceNow Fluent includes APIs for defining the different types of metadata.

With the ServiceNow IDE or ServiceNow SDK, you can also create JavaScript modules and use third-party libraries in your application to optimize code reuse in scripts within an application scope.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Developing and maintaining applications in source code enables you to work in familiar development environments, create and modify complex applications, manage code in source control more easily, and catch errors at build time.

ServiceNow IDE



Develop applications in code in an IDE on the ServiceNow AI Platform.

ServiceNow SDK



Develop applications in code locally and upload changes to an instance.

Comparison of the ServiceNow IDE and the ServiceNow SDK

Feature	ServiceNow IDE	ServiceNow SDK
Development environment	Online ServiceNow instance	Local development and the ability to work offline
User interface	IDE based on Visual Studio Code for the Web	Visual Studio Code Desktop
Collaboration	View any user's changes in real time in code or embedded ServiceNow AI Platform user interfaces. Collaborate with other developers on applications in source control.	Download changes from an instance and install local changes to an instance using the ServiceNow SDK CLI to collaborate with other users. Collaborate with other developers on applications in source control.
Source control	Supports the most common Git functionality and integrating with a Git provider of your choice. One concurrent branch per repository per instance (or developer sandbox).	Full support
Application conversion	Support for converting existing applications not created with the ServiceNow IDE or ServiceNow SDK.	Support for converting existing applications not created with the ServiceNow IDE or ServiceNow SDK.
ServiceNow Fluent	Full support The ServiceNow Fluent Language server is included with the ServiceNow IDE.	Full support The ServiceNow Fluent Language server can be installed from the Visual Studio Code Extension Marketplace.
JavaScript modules and	Full support	Full support

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Feature	ServiceNow IDE	ServiceNow SDK
third-party libraries		
ServiceNow Otto for Code	Code auto-complete support for scripts. For information about activation, see ServiceNow Otto for Code.	Not supported
Build Agent	Full support. Build Agent is accessed from a chat panel in the ServiceNow IDE. For information about activation, see Build Agent.	Build Agent skills are available beginning with ServiceNow SDK version 4.6.0.

Application structure

Applications created or converted with the ServiceNow IDE or ServiceNow SDK include source code files and metadata XML files. The `package.json` and `now.config.json` files define the application structure, which is similar to that of Node.js applications or Node Package Manager (npm) packages.

Default structure of an application created in the ServiceNow IDE



By default, applications include the following directories and files. You can modify certain aspects of the application structure to suit your needs in the `now.config.json` file.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Application structure in source code

Directory or file	Description
.vscode	Directory containing recommended Visual Studio Code extensions.
dist	Directory containing the build artifacts for packaging. This directory includes the following subdirectories: • `app`: Directory containing the built metadata XML files. • `static`: Directory containing the built static asset files.
metadata	Directory containing the application metadata (XML) of the application, such as table schemas and business rules, organized in the same directory structure as existing ServiceNow applications. **Note:** Application metadata shouldn't be edited from the XML files. Edit application metadata in the source code or on the ServiceNow AI Platform.
node_modules	Directory containing the third-party Node.js modules on which your application depends.
src	Directory containing the source code of your application. This directory includes the following subdirectories: • `client`: Directory containing the client-side files for developing user interfaces. • `fluent`: Directory containing ServiceNow Fluent code in `.now.ts` files. The `generated` subdirectory contains the application files converted to ServiceNow Fluent. • `server`: Directory containing JavaScript module code in `.js` or `.ts` files.
target	Directory containing an installable package (`.zip` file) to upload to an instance.
.eslintrc	File containing the ESLint configuration. ESLint helps identify and fix issues in the application code.
.gitignore	File containing a list of directories or files for Git to ignore. These files aren't tracked in source control.
now.config.json	File containing the ServiceNow application configuration. The `now.config.json` file must be in the base directory for an application. You can configure aspects of an application by adding support parameters. For more information, see Custom application configuration in source code.
now.prebuild.mjs	Auto-updated file containing complete information about dependencies and their versions. This file is only available with the ServiceNow SDK.
package-lock.json	Auto-updated file containing complete information about dependencies and their versions. This file is only available with the ServiceNow SDK.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Application structure in source code (continued)

Directory or file	Description
package.json	File containing information about your application and custom or third-party module dependencies. The `package.json` file must be in the base directory for an application. On an instance, the `package.json` path is specified in the **Package JSON** field of the custom application record [sys_app] in the format `<scope>/<package-name>/<version>/package.json`.

Related applications and features

JavaScript APIs

Use JavaScript APIs in scripts that you write to change the functionality of applications or when you create applications.

Related topics

ServiceNow Fluent

JavaScript modules and third-party libraries

User interface development with React

Custom application configuration in source code

ServiceNow Fluent

Define application metadata in source code using the ServiceNow Fluent domain-specific programming language.

Overview of ServiceNow Fluent

ServiceNow Fluent is a declarative, domain-specific language (DSL) based on TypeScript for defining the metadata files [sys_metadata] that make up applications and includes APIs for the different types of metadata, such as tables, roles, ACLs, business rules, and Automated Test Framework tests.

Developers define this metadata in a few lines of code instead of through a form or builder tool user interface. Applications created or converted with the ServiceNow IDE or ServiceNow SDK support development in ServiceNow Fluent.

ServiceNow Fluent supports two-way synchronization, which allows changes to metadata to be synced from other ServiceNow AI Platform user interfaces into source code and changes to source code to be synced back to metadata across the instance.

To get started using the ServiceNow IDE or ServiceNow SDK, see the ServiceNow IDE or ServiceNow SDK documentation.

ServiceNow Fluent APIs

ServiceNow Fluent includes APIs for many types of metadata. You can use the Record API to define application metadata that doesn't have a dedicated API. For the latest list of supported APIs and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub .

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: A limited number of metadata types, such as Metadata Snapshots [sys_metadata_link] and UX Assets [sys_ux_lib_asset], can't be represented as ServiceNow Fluent code and aren't transformed. These metadata types remain as metadata XML files in the `metadata` directory of your application.

ServiceNow Fluent usage

In files with the `.now.ts` extension, use objects in the ServiceNow Fluent APIs to define metadata in the application. You must also include the required imports for the APIs from `@servicenow/sdk/core`. For objects with server-side scripts, such as the *`BusinessRule`* object, you can import and use code from JavaScript modules.

The following example includes the definitions of a table, client script, and business rule in the application. The client script uses a script from the `client-script.js` file. The business rule uses a function from the `script.js` JavaScript module.

```
import '@servicenow/sdk/global'
import { BusinessRule, ClientScript, DateColumn, StringColumn,
 Table } from '@servicenow/sdk/core'
import { showStateUpdate } from '../server/script.js'

//creates todo table, with three columns (deadline, status and task)
export const x_snc_example_to_do = Table({
    name: 'x_snc_example_to_do',
    schema: {
        deadline: DateColumn({ label: 'Deadline' }),
        state: StringColumn({
            label: 'State',
            choices: {
                ready: { label: 'Ready' },
                completed: { label: 'Completed' },
                inProgress: { label: 'In Progress' },
            },
        }),
        task: StringColumn({ label: 'Task', maxLength: 120 }),
    },
})

//creates a client script that pops up 'Table loaded successfully!!' message everytime todo
 record is loaded
ClientScript({
    $id: Now.ID['cs0'],
    name: 'my_client_script',
    table: 'x_snc_example_to_do',
    active: true,
    appliesExtended: false,
    global: true,
    uiType: 'all',
    description: 'Custom client script generated by Now SDK',
    isolateScript: false,
    type: 'onLoad',
    script: Now.include('../client/client-script.js'),
})

//creates a business rule that pops up state change message whenever a todo record is
 updated
BusinessRule({
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    $id: Now.ID['br0'],
    action: ['update'],
    table: 'x_snc_example_to_do',
    script: showStateUpdate,
    name: 'LogStateChange',
    order: 100,
    when: 'after',
    active: true,
})
```

The client script referenced from the *ClientScript* object:

```
function onLoad() {
    g_form.addInfoMessage("Table loaded successfully!!")
}
```

The JavaScript module referenced from the *BusinessRule* object:

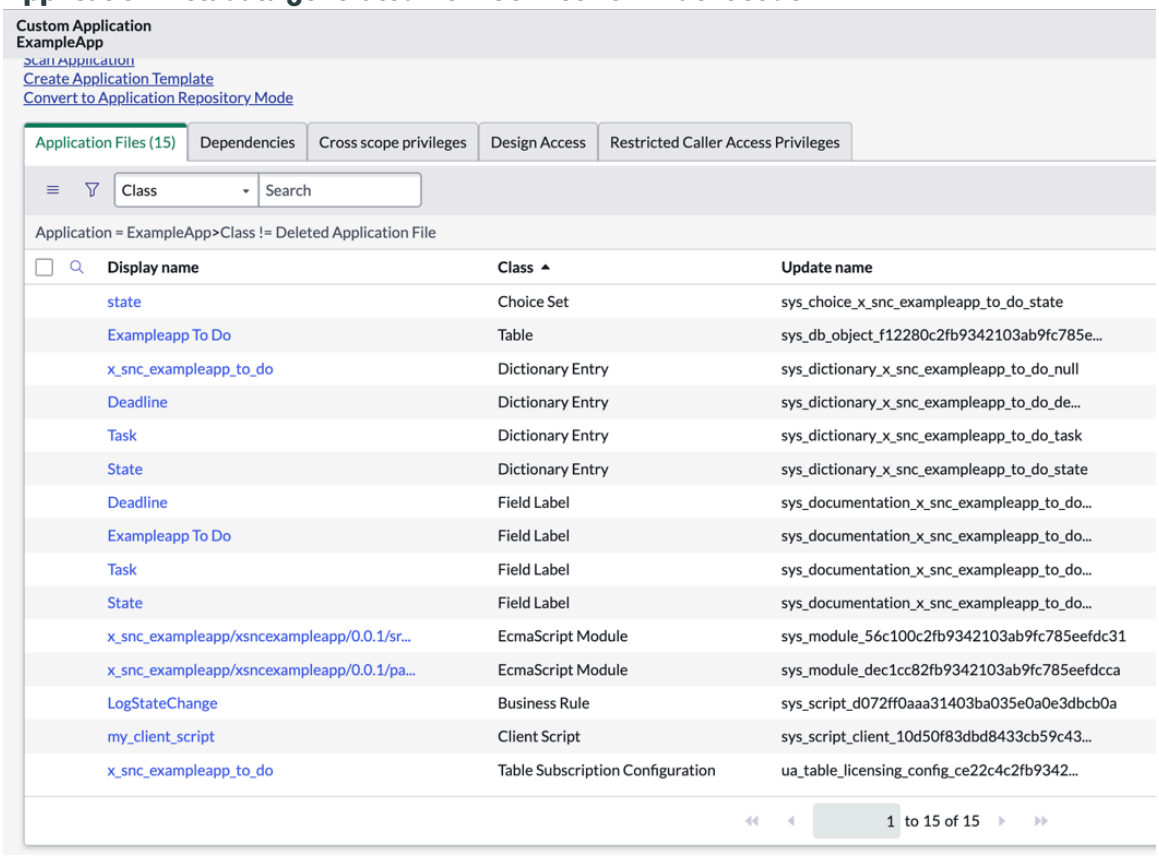
```
import { gs } from '@servicenow/glide'

export function showStateUpdate(current, previous) {
    const currentState = current.getValue('state')
    const previousState = previous.getValue('state')

    gs.addInfoMessage(`state updated from "${previousState}"
 to "${currentState}"`)
}
```

After building the application, this source code generates the following application metadata files on the instance.

Application metadata generated from ServiceNow Fluent code



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Tip: You can use the following directives in a code comment to help manage your code:

- `@fluent-ignore`: Suppresses ServiceNow Fluent diagnostic warnings and errors in the following line of code.
- `@fluent-disable-sync`: Turns off syncing changes to a ServiceNow Fluent object. Use before a call expression (for example, `Record({ ... })`) to turn off syncing for that object and its child objects. Only use this directive if you want to ignore changes made outside of the source code to the object and never update it when syncing.
- `@fluent-disable-sync-for-file`: Turns off syncing changes to a ServiceNow Fluent file (`.now.ts`). Use in the first line of the file to turn off syncing for all code in the file. Only use this directive if you want to ignore changes made outside of the source code to the file and never update it when syncing.

Related topics

ServiceNow Fluent API reference

Define application metadata in code with ServiceNow Fluent in the ServiceNow IDE

Define application metadata in code with ServiceNow Fluent and the ServiceNow SDK

JavaScript modules and third-party libraries

Optimize your code base using JavaScript modules to group related code or add third-party libraries and reuse their code within applications.

Overview of using JavaScript modules

A module is a JavaScript file that contains related code that's shared and reused within applications on an instance. You can add JavaScript modules and third-party libraries in applications that are created or converted with the ServiceNow IDE or ServiceNow SDK. You can also use TypeScript to create modules and compile them to JavaScript before building your application. After installing an application on an instance, JavaScript modules are stored in the EcmaScript Module [sys_module] table.

In a module, you identify code for reuse with *`export`* statements. Then, use *`import`* or *`require`* statements to reuse the code elsewhere in your applications. You must add third-party Node Package Manager (npm) libraries to applications as dependencies to use their module code. For general information about the syntax used to create JavaScript modules, see the JavaScript modules page on the MDN Web Docs website.

Note: To use global Glide Server APIs in modules, they must be imported from the `@servicenow/glide` package. For more information, see Importing server APIs.

Limitations

- Application customizations aren't supported.
- Modules can be used only within the application scope in which they're added. They can't be used across application scopes.
- A subset of ECMAScript features are supported in modules in accordance with the JavaScript engine feature support.
- Node.js APIs aren’t supported in modules. The ServiceNow SDK build process polyfills any Node.js built-in modules while packaging modules, otherwise modules are resolved from the `node_modules` directory.
- Global variables related to web APIs aren’t supported.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- CommonJS modules from third-party libraries aren't supported unless they define exports. Subpath imports aren't supported with CommonJS modules. ECMAScript modules from third-party libraries are supported.
- `import` and `export` statements are only supported in modules. To import module code in scripts, such as business rules or script includes, use `require` statements.
- JavaScript modules [sys_module] can be modified only in the ServiceNow IDE or in Visual Studio Code with the ServiceNow SDK.

Important: You can't use third-party libraries that rely on unsupported functionality, such as unsupported APIs or ECMAScript features. For more information, see Third-party library support in Brazil.

Exporting modules

In a module, identify code for reuse with `export` statements. You can use named exports or default exports. Named exports can be for variables, constants, functions, or classes whereas default exports can be for functions or classes only. The following example is one way of adding a named export for multiple features (a function and a variable) in a module:

```
export { myFunction, myVariable };
```

Importing modules

To import the module code you want to reuse, use `import` statements in other modules or `require` statements in server-side scripts.

The following example is one way that you could import an exported feature in a module:

```
import { feature } from 'path/to/module';
```

The following example is one way that you could import an exported feature in a script:

```
const { feature } = require('path/to/module');
```

Note: To import code from one TypeScript file to another TypeScript file, you must include the `.ts` file extension. For example, `import { feature } from './module.ts'`.

To use shorthand to import module code, you can use subpaths in the `imports` field of the application's `package.json` file. For example:

```
{
 "name": "math",
 "version": "1.0.0",
 "exports": {
   "./functions/*.js": "./src/functions/*.js",
   "./functions/private-functions/*": null
 },
 "imports": {
   "#calc": "calculus",
   "#derivative": "calculus/derivative"
 },
 "dependencies": {
   "calculus": "1.0.0"
 }
}
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Based on that example, instead of writing out the relative path to `derivative.js` every time you want to import it in the `math` application, you can use the `#derivative` shorthand instead. Subpaths can also be used in the *`imports`* field to use shorthand for dependencies, such as `#calc`.

```
import { derivative } from '#derivative';
import * as calculus from '#calc';
```

Adding third-party libraries

Applications must declare dependencies on third-party libraries to use their module code. In an application's `package.json` file, include the package name and version for any dependencies. For example, to use modules from the "math" library in the "test" application, add the "math" package as a dependency:

```
{
 "name": "test",
 "version": "1.0.0",
 "dependencies": {
  "math": "1.0.0"
 }
}
```

Importing server APIs

To import server APIs and use them in a module, use *`import`* statements. Glide APIs can be imported from the `@servicenow/glide` package or their namespace in the package.

For example:

```
import { API } from '@servicenow/glide';
import { API } from '@servicenow/glide/<namespace>';
```

In the following example, the *`gs`* (GlideSystem) and *`GlideRecord`* APIs are imported in a module:

```
import { gs } from '@servicenow/glide';
import { GlideRecord } from '@servicenow/glide';
```

In the following example, the *`RESTAPIRequest`* and *`RESTAPIResponse`* APIs are imported from the `sn_ws_int` namespace in a module because they run in that namespace:

```
import { RESTAPIRequest, RESTAPIResponse }
 from '@servicenow/glide/sn_ws_int';
```

To access server APIs in a third-party library module, you must add the module as a trusted module with the *`trustedModules`* parameter in your application's `now.config.json` file. For more information, see Custom application configuration in source code.

For more information about available server APIs, see Server API reference.

Importing script includes

To import script includes and use them in a module, use *`import`* statements. Script includes can be imported from their application scope or the global scope in the `@servicenow/glide` package.

For example:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
import { global } from '@servicenow/glide/global';
import { ScriptInclude } from '@servicenow/glide/<scope>';
```

For more information about script includes, see Script includes.

Related topics

- Create and use JavaScript modules in applications in the ServiceNow IDE
- Create and use JavaScript modules in applications with the ServiceNow SDK

Third-party library support in Brazil

Review which common third-party libraries are supported, partially supported, or unsupported by the ServiceNow IDE and ServiceNow SDK.

The following third-party libraries have been tested in the Brazil release. Libraries with partial support or no support are limited by server-side scripting features that aren't currently supported by the JavaScript engine. For more information about feature support for server-side scripts, see JavaScript engine feature support.

Note: The libraries tested were prioritized based on their number of Node Package Manager (npm) downloads, applicability to the ServiceNow AI Platform, and user interest. For questions or requests to test additional third-party libraries, see the ServiceNow IDE, SDK, and Fluent forum in the ServiceNow Community.

Supported third-party libraries

- @langchain/textsplitters
- acorn
- bn.js
- camelcase
- chance
- clone
- date-fns
- dayjs
- debug
- diff
- docxtemplater
- extend
- fast-csv
- html-parser-lite
- iconv-lite
- inherits
- ini
- js-yaml
- lodash
- lru-cache
- marked

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- mime
- mime-types
- moment
- moment-timezone
- ms
- openai
- path-to-regexp
- pdf-lib
- pizzip
- ramda
- reflect-metadata
- saxes
- semver
- strip-ansi
- tslib
- underscore
- uuid
- validator
- xtend

Third-party libraries with limited support

- @aws-sdk/client-s3
- effect
- jszip
- minimatch
- underscore.string
- util

Unsupported third-party libraries

- @faker-js/faker
- ajv
- archiver
- axios
- deep-equal
- eml-parser
- exceljs
- jws
- qs
- unzipper

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

User interface development with React

Develop a user interface (UI) with the React library to build a full-stack application in source code.

Overview of UI development with React

React is an industry-standard web framework for building UI components that you use to develop and render custom user interfaces. A React component is a self-contained, reusable JavaScript function, such as a button, and supports JavaScript (JS/JSX) and TypeScript (TS/TSX) syntax. For general information about React, see the React documentation on the react.dev website.

With the ServiceNow IDE or ServiceNow SDK, you can use React in an application to create a UI page in ServiceNow Fluent code. The ServiceNow Fluent UI Page API refers to an HTML entry point (`index.html`) that loads the page at the endpoint provided. After building and installing the application on an instance, the static assets are stored in the appropriate tables. For an example of a React application in source code, see the ServiceNow SDK examples GitHub repository. To get started using React, select one of the React templates when creating an application with the ServiceNow IDE or ServiceNow SDK.

Note: You can use some Next Experience Components in a React application with the React Wrapper Component Library Node Package Manager (npm) package on the public npm registry. To use the React Wrapper Component Library in an application, you can use Build Agent or you must install it from the ServiceNow IDE or ServiceNow SDK. For more information, see Use third-party libraries with the ServiceNow IDE or Use third-party libraries with the ServiceNow SDK.

UI development process

The following list is a high-level overview of the process to develop a React application with the ServiceNow IDE or ServiceNow SDK:

1. Create an application and select one of the React templates.

   The application includes the files and directories required for React development.

2. In the `src/client` directory, add client assets for a user interface, such as an HTML entry point (`index.html`), client scripts, and style sheets. The `index.html` file must contain the `<script>` tag as the JavaScript entry point.
3. Define a UI page [sys_ui_page] in code with the ServiceNow Fluent UI Page API. In the `now.ts` file that contains the UI page definition, you import the HTML and refer to it from the *`html`* property of the *`UiPage`* object.
4. Build the application to execute a pre-build Rollup script that bundles dependencies and packages the client assets in the `src/client` directory before executing the rest of the build process. The standard Rollup build process and plugins are used as the default JavaScript bundler.

   Static assets are output to the `dist/static` directory.

5. Install the application to add the static assets to the application files [sys_metadata] in the appropriate tables: UX Asset [sys_ux_lib_asset], Images [db_image], and UX Theme Asset [sys_ux_theme_asset] tables. These tables support adding files as attachments.

Important: UI development with React is an experimental feature due to how content is served from an instance and the content types and tables that are currently supported.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Limitations

- The UI page HTML should be modified only in source code. Changes to the HTML of a UI page [sys_ui_page] on an instance aren't synchronized into source code and are likely to result in unintended behavior.
- Audio, video, and WASM content types aren't supported.
- The maximum file size of uploaded attachments is limited by the size configured with *com.glide.attachment.max_size* system property. For more information, see Maximum allowed attachment size.
- Output paths must be deterministic.
- Pre-loading content linked from HTML isn't supported (`rel="preload"`).
- Relative style sheets linked from HTML aren't supported (`rel="stylesheet"`). Import style sheets into code instead (`@import 'path/to/style-sheet'`).
- Relative imports in CSS aren't supported.
- CSS modules aren't supported.
- Only hash routing is supported by UI pages.
- Server-side rendering and React server components aren't supported.

Example: UI page development with React

In a `.now.ts` file in the `src/fluent` directory, a UI page is defined in ServiceNow Fluent code. The HTML of the page is imported from the `index.html` file in the `src/client` directory.

```
//incident-manager.now.ts
import '@servicenow/sdk/global'
import { UiPage } from '@servicenow/sdk/core'
import incidentPage from '../../client/index.html'

UiPage({
    $id: Now.ID['incident-manager-page'],
    endpoint: '<scope>_incident_manager.do',
    description: 'Incident Response Manager UI Page',
    category: 'general',
    html: incidentPage,
    direct: true,
})
```

In the `index.html` file, the `<script>` tag refers to the `main.jsx` file in the `src/client` directory.

```
//index.html
<html>
<head>
  <title>Incident Response Manager</title>

  <!-- Initialize globals and Include ServiceNow's required scripts -->
  <sdk:now-ux-globals></sdk:now-ux-globals>

  <!-- Include your React entry point -->
  <script src="./main.jsx" type="module"></script>
</head>
<body>
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    <div id="root"></div>
</body>
</html>
```

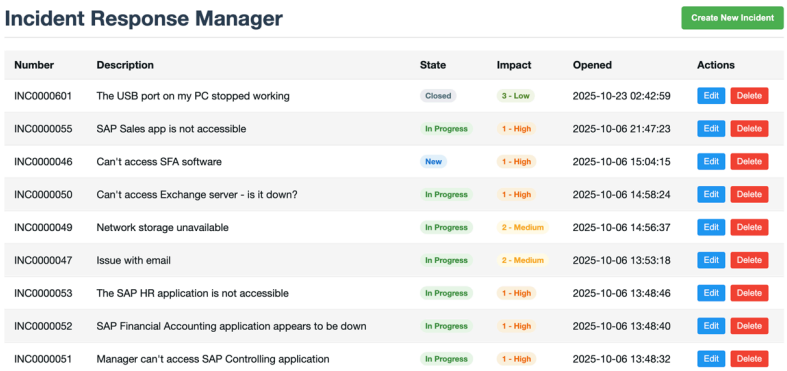
In the `main.jsx` file, the imports required for rendering the page and the main application component are included and the entry point of the React application is defined.

```
//main.jsx
import React from 'react'
import ReactDOM from 'react-dom/client'
import App from './app'

const rootElement = document.getElementById('root')
if (rootElement) {
    ReactDOM.createRoot(rootElement).render(
        <React.StrictMode>
            <App />
        </React.StrictMode>
    )
}
```

After building and installing the application, you can open the page from the endpoint provided in the *UiPage* object (`https://<instance>/<scope>_incident_manager.do`), which displays an interface for creating and managing incidents.

Example of a UI page created with React



Related topics

- UI Page API - ServiceNow Fluent
- ServiceNow Fluent

Custom application configuration in source code

Configure a custom application [sys_app] in the `now.config.json` file for an application in source code.

Note: For the latest ServiceNow SDK application configuration documentation, see the now.config.json reference on GitHub.

You can configure the application settings and aspects specific to developing the application in source code, such as the directory structure. In the `now.config.json` file, add the following parameters to configure the application settings.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Supported now.config.json parameters

Parameter	Type	Description
appOutputDir	String	Directory to output the build artifacts to for packaging. The pack and install commands refer to this directory to package the artifacts. Default: `dist/app`
clientDir	String	Directory containing the client-side files for developing user interfaces with React. Default: `src/client`
dependencies	Object	The items in another application scope on which your application depends. You must specify the application scope and the dependency type and names or sys_ids. `"dependencies": { "<scope>": { "<type>": ["<sys_id or name>"], ... }, ... }` For more information, see Download ServiceNow Fluent application dependencies.
excludeFilePatterns	Array	A list of file patterns to exclude when transforming metadata XML to ServiceNow Fluent code or downloading dependencies. Files that match are discarded and not written to the metadata directory.
fluentDir	String	Directory containing ServiceNow Fluent files (`.now.ts`) that define application metadata in source code. Default: `src/fluent`
generatedDir	String	Directory containing generated ServiceNow Fluent files, including existing application metadata converted into ServiceNow Fluent code. This directory is relative to the directory defined with the *`fluentDir`* parameter. Default: `generated`
ignoreTransformTables	Array	A list of tables to ignore when transforming application metadata into source code.
linter	Object	A configuration for ServiceNow Fluent and JavaScript module linting behavior. The module.enabled property controls whether to run linting on modules for compatibility with the JavaScript engine on the platform. If false, linting is turned off for local modules (in `src/server`) and third-party npm packages (in `node_modules`).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Supported now.config.json parameters (continued)

Parameter	Type	Description
		Default: `"linter": { "module": { "enabled": true } }`
metadataDir	String	Directory containing the application's metadata as XML files. Default: `metadata`
modulePaths	Object	A map of the module source files to their equivalent output files for if you use a custom transpilation step before building the application. For more information, see Using TypeScript in JavaScript modules with the ServiceNow SDK. **Warning:** You can't use this parameter and the `tsconfigPath` parameter. Configuring both results in an error.
packOutputDir	String	Directory to output the installable package (`.zip` file) when building the application. The install command refers to this directory to install the package. Default: `target`
serverModulesDir	String	Directory containing the JavaScript or TypeScript files to be built into JavaScript modules for use in server-side scripts. Default: `src/server`
serverModulesExcludeArray	Array of patterns	A list of file patterns to exclude when building JavaScript modules. Default: `["**/*.test.ts", "**/*.test.js", "**/*.d.ts"]`
serverModulesIncludeArray	Array of patterns	A list of file patterns to include when building JavaScript modules. Default: `["**/*.ts", "**/*.tsx", "**/*.js",`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Supported now.config.json parameters (continued)

Parameter	Type	Description
		<pre>"**/*.jsx", "**/*.cts", "**/*.cjs", "**/*.mts", "**/*.mjs", "**/*.json"]</pre>
staticContentDir	String	Directory to output the static asset files used for developing user interfaces. Default: `dist/static`
staticContentPaths	Object	A map of the client-side source files to the output paths for static asset files.
defaultLanguage	String	The BCP 47 code of a default language for field labels [sys_documentation] in a table or column. The default language is used to resolve field labels with multiple languages. Default: en
tableOutputFormat	String	The type of build artifacts for table metadata XML generated from ServiceNow Fluent code. Valid values: • bootstrap: The build process outputs a bootstrap XML file with the `<database>` root element for the table, field label XML files [sys_documentation], licensing configuration XML files [ua_table_licensing_config], and auto-numbering XML files [sys_number]. • component: The build process outputs XML files for each component of the Table API. Default: `bootstrap`
taxonomy	Object	A configuration for organizing generated ServiceNow Fluent files, which maps table names to directories and defines a fallback directory. The default taxonomy configuration uses ServiceNow standard table classifications to add generated ServiceNow Fluent files in a logical directory structure within the `fluent/generated` directory when metadata is initially transformed into ServiceNow Fluent code. For example: • Business rules [sys_script] are added to the `fluent/generated/server-development/business-rule` directory. • Script includes [sys_script_include] are added to the `fluent/generated/server-development/script-include` directory.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Supported now.config.json parameters (continued)

Parameter	Type	Description
		You can override the default mappings or configure additional ones. In the following example, the configuration overrides the default directory for business rules [sys_script] and the fallback folder and configures an additional mapping for metadata from a custom table. `"taxonomy": { "mapping": { "sys_script": "scripts/server/rules", "custom_table": "my-custom-folder/my-nested-folder" }, "fallbackFolderName": "unclassified" }` • mapping: An object that maps table names to directories. Directory paths are relative to the directory configured with the *generatedDir* parameter and must include only lowercase letters, numbers, hyphens, underscores, and slashes to separate subdirectories. • fallbackFolderName: A name for a directory to use for tables that don't have a default or a custom mapping configured. Within this directory, ServiceNow Fluent files are added to subdirectories named after the table, such as `src/fluent/generated/other/x-unmapped-table/`. Default: Default taxonomy mappings are defined for all standard ServiceNow tables and are automatically applied when no custom configuration is defined in the `now.config.json` for an application. The default value of *fallbackFolderName* is `other`.
trustedModules	Array	A list of npm packages to identify as trusted (or internal). Trusted modules have access to ServiceNow APIs. For example: `"trustedModules": [ "<package-name>", // Specific package "@servicenow/*" // All packages from an organization ]` In the EcmaScript Module [sys_module] table, the **External source** field is set to false for trusted modules. ⚠ **Warning:** Only add dependencies that you trust completely as trusted modules.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Supported now.config.json parameters (continued)

Parameter	Type	Description
		Valid patterns: • Fully qualified package names, such as `'@servicenow/sdk'`. • Organization prefixes with a wildcard, such as `'@servicenow/*'` or `'@mycompany/*'`.
tsconfigPath	String	A path to a `tsconfig.json` file with custom options for transpiling TypeScript into JavaScript during the build process. Specifying a `tsconfigPath` generates diagnostic results from TypeScript using the `tsconfig.json` file. **Warning:** You can't use this parameter and the `modulePaths` parameter. Configuring both results in an error. Default: .
applicationRuntimePolicy	String	The runtime policy. For more information about runtime policies, see Runtime access tracking. Valid values: • none: The system does not track runtime access requests. Application scripts can access resources from other applications as long as the table-level access settings allow it. Any configured network policies (`networkPolicies`, `wildcardPolicy`, and `performancePolicy`) are created but not enforced. • tracking: The system runs the tracked operation as long as the table-level access settings allow it. Cross-scope privileges and any configured network policies (`networkPolicies`, `wildcardPolicy`, and `performancePolicy`) are created and enforced. • enforcing: The system blocks the tracked operation from running until an administrator manually changes the status to **Allowed** and the table-level access settings allow it. Cross-scope privileges and any configured network policies (`networkPolicies`, `wildcardPolicy`, and `performancePolicy`) are created and enforced. Default: none
networkPolicies	Array	A list of network access policy [sys_arp_network_policy] definitions.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Supported now.config.json parameters (continued)

Parameter	Type	Description
		• policyType: Required. The type of network policy. If the policy type is now_inbound_global, the `scheme` property is required. Valid values: csp_script_src, csp_connect_src, now_inbound_scoped, now_inbound_global, now_outbound. • status: Required. The enforcement status of the policy. Valid values: requested, allowed, denied • $id: A unique ID for the metadata object in the format `Now.ID['String' or Number]`. • active: Flag that indicates whether the policy is active. • host: Host address including the scheme, hostname, and optional port but not including the path. • scheme: The protocol scheme. Required if the policy type is now_inbound_global and must be empty for all other types. Valid values: http, https, ws, wss • path: A list of paths. Each path must start with a slash (`/`). Use the `/*` suffix to allow for sub-paths (`/api/*`). If the policy type is now_inbound_scoped, provide either a path or resource. • resource: A resource identifier for cross-scope access if the policy type is now_inbound_scoped. For example, `processor.subprocessor.context`. Provide either a path or resource. • shortDescription: A short description of what the policy does. `"networkPolicies": [ { "policyType": "String", "status": "String", "$id": "String" or Number, "active": Boolean, "host": "String", "scheme": "String", "path": [Array], "resource": "String", "shortDescription": "String" } ]`
wildcardPolicy	Object	A wildcard, or exemption, policy [sys_arp_segment_policy] definition.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Supported now.config.json parameters (continued)

Parameter	Type	Description
		• $id: A unique ID for the metadata object in the format `Now.ID['String' or Number]`. • active: Flag that indicates whether the policy is active. Default: false • network: The network pillar configuration that includes the network wildcard selections. Valid values for `networkWildcard`: now_outbound, now_inbound_global, now_inbound_scoped, csp_connect_src, csp_script_src • scripting: The scripting pillar configuration that includes the scripting wildcard selections. Valid values for `scriptingWildcard`: sys_script_include, scriptable • arl: The application resource limit (ARL) pillar configuration that includes the network wildcard selections. Valid values for `arlWildcard`: scheduled_job_limit, event_handler, api_transaction_limit, interactive_transaction_limit • record: Flag that indicates whether to allow record pillar access. • shortDescription: A short description of what the policy does. `"wildcardPolicy": { "$id": "String" or Number, "active": Boolean, "network": { "active": Boolean, "networkWildcard": [Array] }, "scripting": { "active": Boolean, "scriptingWildcard": [Array] }, "arl": { "active": Boolean, "arlWildcard": [Array] }, "record": Boolean, "shortDescription": "String" }`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Supported now.config.json parameters (continued)

Parameter	Type	Description
performancePolicy	Object	A performance quota template [sys_app_resource_limit_template] definition. • name: Required. A name for the performance quota template. • $id: A unique ID for the metadata object in the format `Now.ID['String' or Number]`. • apiTransactionLimit: The API transaction quota percentage. • eventHandlerLimit: The event handler quota percentage. • interactiveTransactionLimit: The interactive transaction quota percentage. • mode: The enforcement mode for quota thresholds. If this parameter is not set, the value is determined by the value of the `applicationRuntimePolicy` parameter. Valid values: disabled, enforced, logOnly • scheduledJobLimit: The scheduled job quota percentage. `"performancePolicy": { "name": "String", "$id": "String" or Number, "apiTransactionLimit": Number, "eventHandlerLimit": Number, "interactiveTransactionLimit": Number, "mode": "String", "scheduledJobLimit": Number }`

Example: Application configuration in source code

```
{
  "scope": "x_snc_example_app",
  "scopeId": "2f8400eb07426110f736e28f69d3017a",
  "name": "ExampleApp",
  "dependencies": {
   "global": {
    "tables": ["incident"],
    "roles": ["admin"],
   },
   "x_custom": {
    "tables": ["custom_table"]
   }
  },
  "metadataDir": "metadata",
  "fluentDir": "src/fluent",
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
  "generatedDir": "generated",
  "serverModulesDir": "src/server",
  "clientDir": "src/client",
  "appOutputDir": "dist/app",
  "staticContentDir": "dist/static",
  "packOutputDir": "target",
  "modulePaths": {
        "src/server/*.ts": "dist/server/*.js",
      },
  "staticContentPaths": {
        "src/client/*.html": "dist/static/*.html",
      },
  "ignoreTransformTableList":
 ["ua_table_licensing_config", "sys_embedded_help_role"],
  "taxonomy": {
    "mapping": {
      "sys_script": "scripts/server/rules",
      "custom_table": "my-custom-folder/my-nested-folder"
    },
    "fallbackFolderName": "unclassified"
  }
}
```

Related topics

Application access settings

ServiceNow IDE

Create and develop applications in source code in an integrated development environment (IDE) on the ServiceNow AI Platform to improve collaboration across development teams and accelerate application development.

https://player.vimeo.com/video/1129991964?h=6929aca67a&badge=0&autopause=0&player_id=0&app_id=58479

Get started

Explore	Configure	Create
		
Learn about developing applications in the ServiceNow IDE.	Install and configure the ServiceNow IDE for developers to use.	Create workspaces and add applications.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Integrate	Develop	Reference
		
Integrate with source control providers.	Develop applications in source code.	Get details about ServiceNow Fluent APIs and more.

Note: You can use Build Agent to help you create and edit applications in the ServiceNow IDE. For more information, see Build Agent.

Which builder should I use to create an app?

Are you a developer who wants to use industry-standard development tools and processes?

The ServiceNow IDE and ServiceNow SDK support developing applications in source code with ServiceNow Fluent, creating JavaScript modules, and using third-party libraries. ServiceNow Fluent is a domain-specific programming language for creating application metadata in code.

The ServiceNow IDE is an implementation of Visual Studio Code for the Web on the ServiceNow AI Platform. The ServiceNow SDK uses Visual Studio Code Desktop locally. For more information, see Building applications in source code.

Are you a developer who wants more control in a centralized user interface?

Build apps smarter and deliver them faster with ServiceNow Studio. ServiceNow Studio empowers platform developers with a modern, unified environment for building on the ServiceNow AI Platform. ServiceNow Studio features streamlined navigation to applications and metadata, integrated low-code tools, efficient tracking and packaging of development work that accelerates development processes and enhances productivity. For more information, see Exploring ServiceNow Studio.

Need a more general app but still want low-code options?

App Engine Studio lets you build a broader range of apps than Creator Studio without being a programming pro. For more information, see Exploring App Engine Studio.

Want to build an app easily, without code?

Creator Studio specializes in helping you craft request-fulfillment applications without writing code. For example, an application to request office supplies by filling out a form, and someone approves or denies your request. For more information, see Exploring Creator Studio.

Troubleshoot and get help

- To learn more about what's new in the Brazil release, see the .
- Introduction to the ServiceNow IDE on ServiceNow University
- ServiceNow IDE Demo video
- ServiceNow IDE and Fluent Creator Toolbox video
- ServiceNow SDK examples GitHub repository

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- ServiceNow IDE, SDK, and Fluent forum in the ServiceNow Community
- ServiceNow IDE, SDK, and Fluent articles in the ServiceNow Community
- Search the Known Error Portal for known error articles
- Contact Customer Service and Support

Related topics

ServiceNow Fluent

JavaScript modules and third-party libraries

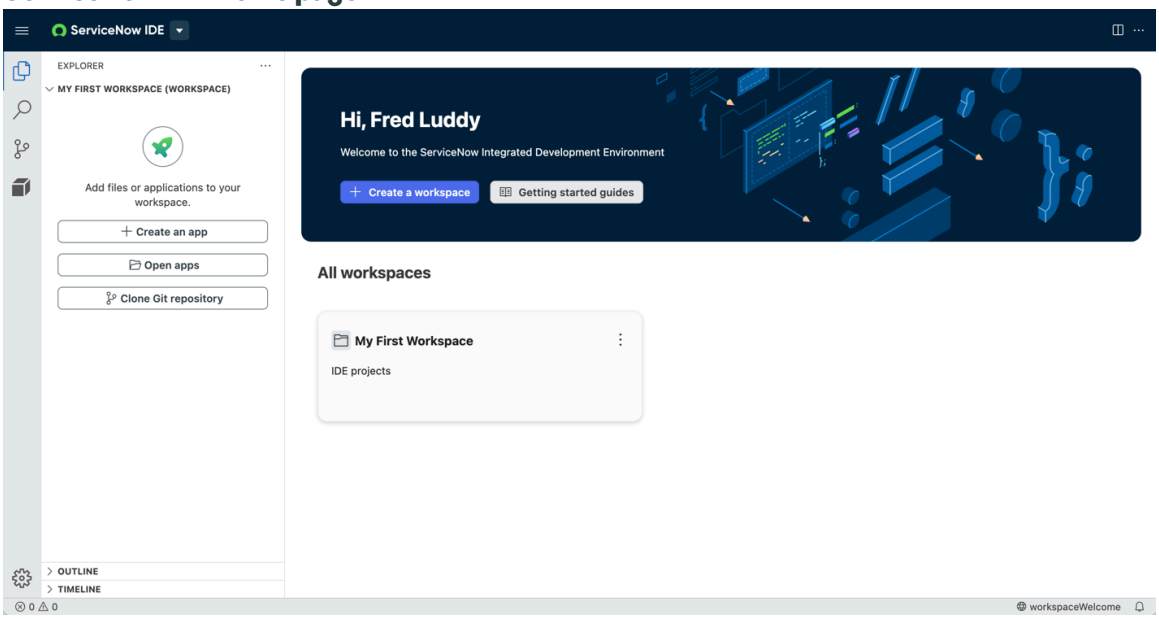
Exploring the ServiceNow IDE

Learn about developing applications in source code in the ServiceNow IDE.

ServiceNow IDE overview

The ServiceNow IDE is an implementation of Visual Studio Code for the Web on the ServiceNow AI Platform. With the ServiceNow IDE, you can get started building applications quickly using familiar tools and industry-standard development practices. The ServiceNow IDE has many of the same features as Visual Studio Code, including type safety, IntelliSense, dependency enforcement, code search, and source control integration.

ServiceNow IDE home page



In the ServiceNow IDE, you can create applications in source code using ServiceNow Fluent to define application metadata [sys_metadata]. To see a side-by-side visual representation of your changes in real time, you can open the application metadata in other ServiceNow AI Platform user interfaces embedded within the ServiceNow IDE.

Optionally, you can create JavaScript modules and use third-party libraries to organize and reuse code within an application scope. With ServiceNow Otto for Code, you can use the Code autocomplete skill to generate code suggestions for scripts in an application in the ServiceNow IDE. For more information, see Generate code with autocomplete.

In the background, this functionality is enabled by the ServiceNow SDK application packaging service, which builds applications in the ServiceNow IDE.

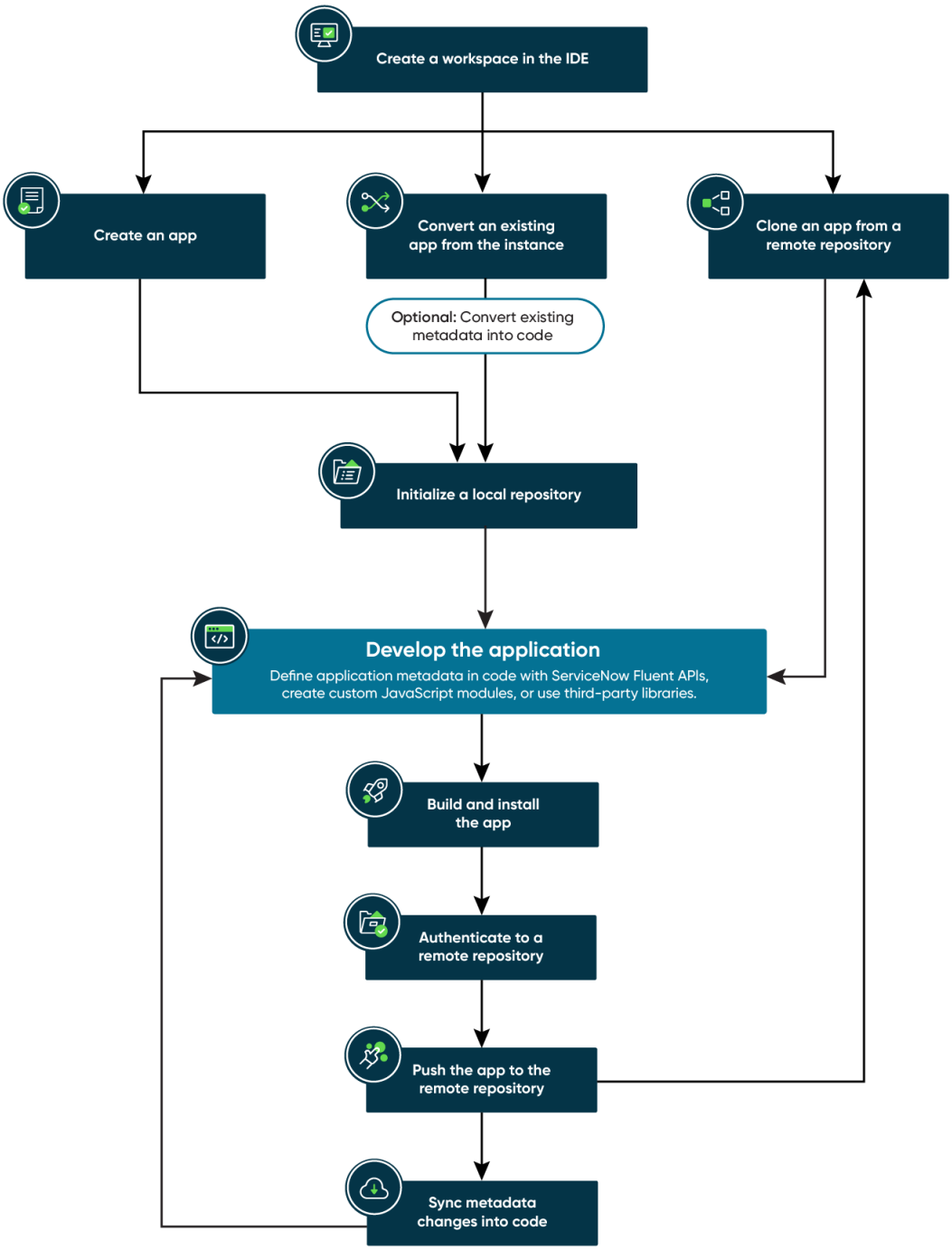
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: You can use Build Agent to help you create and edit applications in the ServiceNow IDE. For more information, see Build Agent.

ServiceNow IDE workflow

The following infographic shows the workflow for developers to get started developing applications with the ServiceNow IDE.

Developing applications with the ServiceNow IDE



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

1. Depending on the authentication method, an administrator or developer configures the credentials needed to connect to their Git provider.
   - To use OAuth 2.0 authentication, an administrator configures an OAuth 2.0 application registry in their Git provider and on the instance.
   - To use basic authentication, a developer generates a personal access token from their Git provider.
2. From their Git provider, a developer creates a dedicated Git repository for an application.
3. From the ServiceNow IDE, the developer configures their Git credentials.
4. The developer creates a ServiceNow IDE workspace to organize applications.
5. The developer creates an application or converts an existing application from the ServiceNow IDE.
6. The developer initializes a local Git repository for the application and pushes it to the remote repository they created to manage the application in source control.
7. The developer can define application metadata in source code using ServiceNow Fluent, create custom JavaScript modules, or use third-party libraries.
8. When metadata changes are detected, the developer synchronizes the application to download and transform changes to metadata from other interfaces into source code.
9. After saving their changes, they build the application, which compiles the source code and transforms it into application metadata.
10. When the build is complete, the developer can view their changes reflected in other embedded ServiceNow AI Platform user interfaces without leaving the ServiceNow IDE.
11. The developer stages, commits, and pushes their changes to the Git repository.
12. From the ServiceNow IDE, other developers can clone the repository, create branches, and begin collaborating on the application.

Other users can modify the application metadata at the same time as developers modify the source code. Developers can reuse module code in other modules or scripts within the application.

ServiceNow IDE benefits

Benefit	Feature	Users
Develop applications in an IDE based on Visual Studio Code on the ServiceNow AI Platform	ServiceNow IDE user interface	Developers
Write source code to define the metadata that makes up ServiceNow applications	ServiceNow Fluent Define application metadata in code with ServiceNow Fluent in the ServiceNow IDE	Developers
Organize and reuse code within an application scope with custom JavaScript modules and third-party JavaScript utilities	JavaScript modules and third-party libraries Create and use JavaScript modules	Developers

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Benefit	Feature	Users
	in applications in the ServiceNow IDE Use third-party libraries in applications in the ServiceNow IDE	
Collaborate on applications with users of different skill sets	Build and install an application in the ServiceNow IDE	Developers
Integrate source control with your Git provider of choice	Integrating source control with the ServiceNow IDE	Developers, Administrators

ServiceNow IDE user interface

The ServiceNow IDE user interface contains many of the same features as Visual Studio Code for the Web and some unique functionality for creating ServiceNow applications.

Overview of the ServiceNow IDE user interface

The ServiceNow IDE is an implementation of Visual Studio Code for the Web on the ServiceNow AI Platform, which includes the following features:

- Basic editing and keyboard shortcuts
- IntelliSense
- Code navigation
- Code and file search
- Command palette
- Source control integration

Visual Studio Code for the Web is similar to Visual Studio Code Desktop but doesn't support all of the features that are available in the desktop interface. For information about browser-based implementations of Visual Studio Code, see Visual Studio Code for the Web. The following Visual Studio Code features aren't supported in the ServiceNow IDE:

- Integrated terminal
- Debugger
- Marketplace extensions
- Integrated unit testing
- Remote development

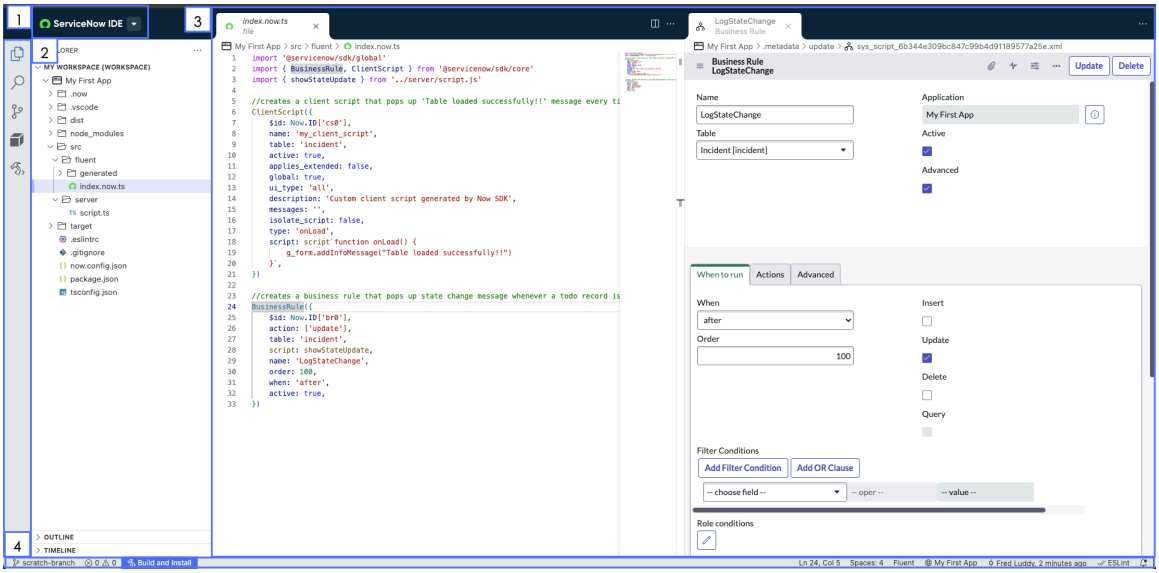
For general information about common features in Visual Studio Code and the ServiceNow IDE, see the Visual Studio Code documentation website.

Features in the ServiceNow IDE

Learn about features of the ServiceNow IDE user interface.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ServiceNow IDE user interface

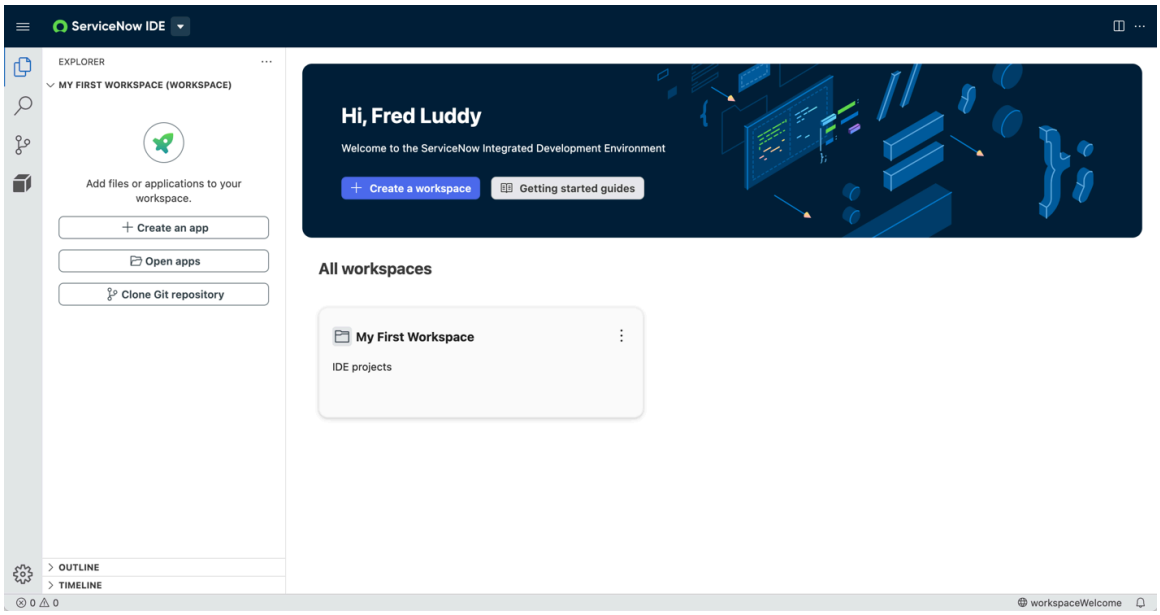


1. ServiceNow IDE home page: Get started creating workspaces and applications and view existing workspaces. Switch between development environments using the experience switcher.
2. Activity Bar
   - File Explorer view: Navigate through applications in your workspace and their code.
   - Search view: Find content within files.
   - Source Control view: Integrate with a Git repository for version control of your code.
   - File Categories view: Create, update, and view application metadata.
   - Now SDK view: Synchronize changes to application metadata or build and install applications.
   - Manage menu: Access and configure various aspects of the user interface.
3. Editor: Write source code and view application metadata through embedded ServiceNow AI Platform user interfaces.
4. Status bar: View status information about your applications and files and build applications.

In addition to these features, you can open the command palette to run a variety of commands to develop applications, manage files, use source control, and more. For more information, see Command palette.

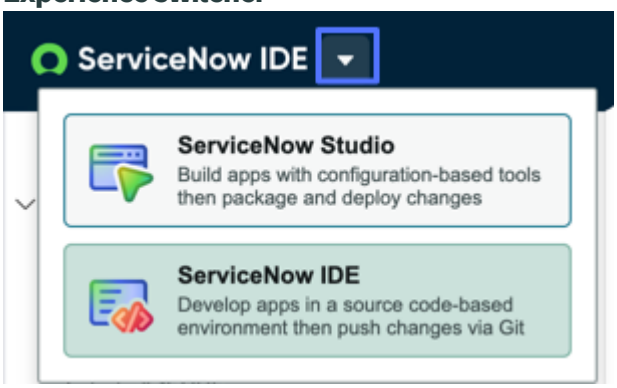
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ServiceNow IDE home page



Create workspaces, access existing workspaces, and create or open applications. Clone a Git repository to add an existing application from a remote repository. To get started creating workspaces and applications, see Developing applications with the ServiceNow IDE. To clone a Git repository, see Clone a Git repository with the ServiceNow IDE.

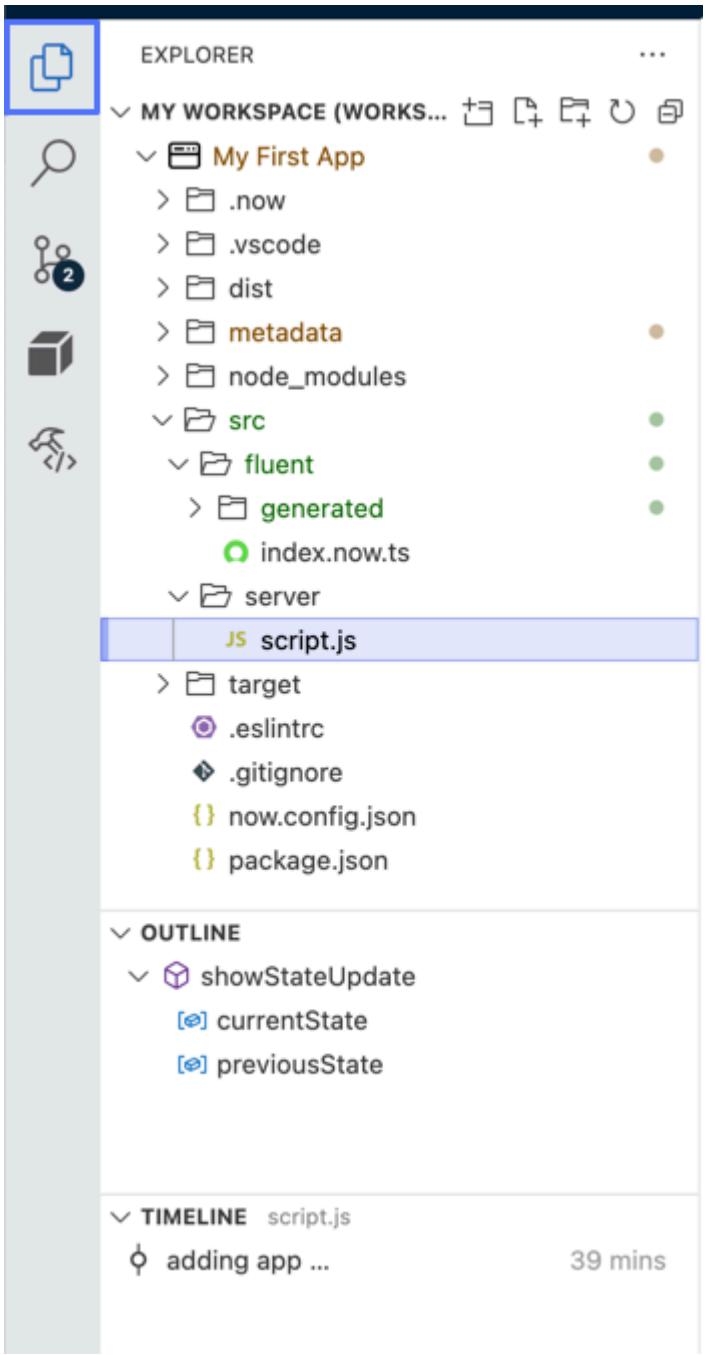
Experience switcher



Work in the right environment for your task by using the experience switcher to switch between developing in ServiceNow IDE, ServiceNow Studio, and Creator Studio. For information about configuring the experience switcher, see Configure non-default access to the experience switcher.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

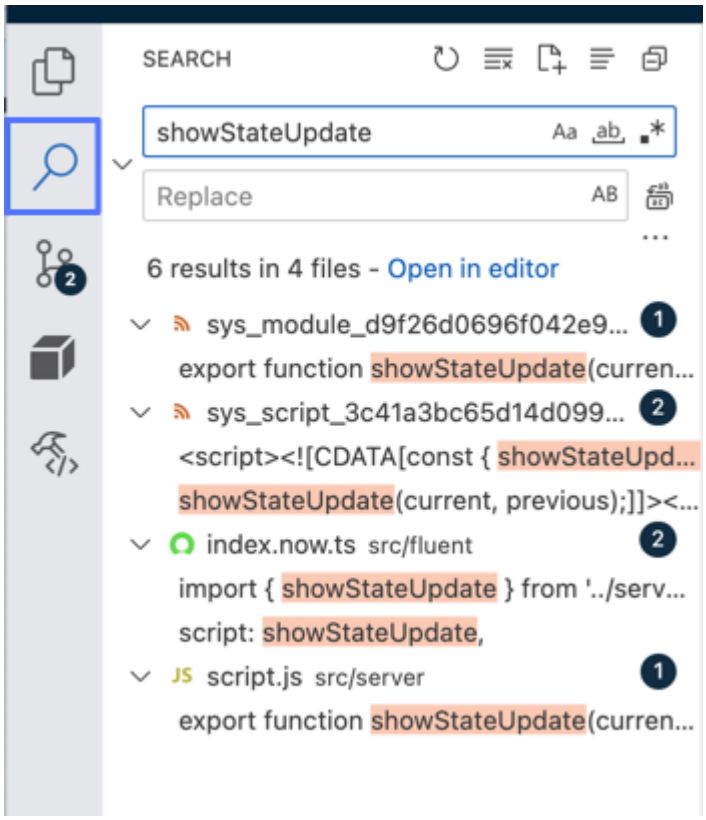
File Explorer view



Navigate through applications in a workspace and add files or folders. In the Outline section, view the outline of the active file in the editor. In the Timeline section, view a history of changes to the active file. For information about the default application file structure, see the Application structure section of the Building applications in source code topic.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Search view



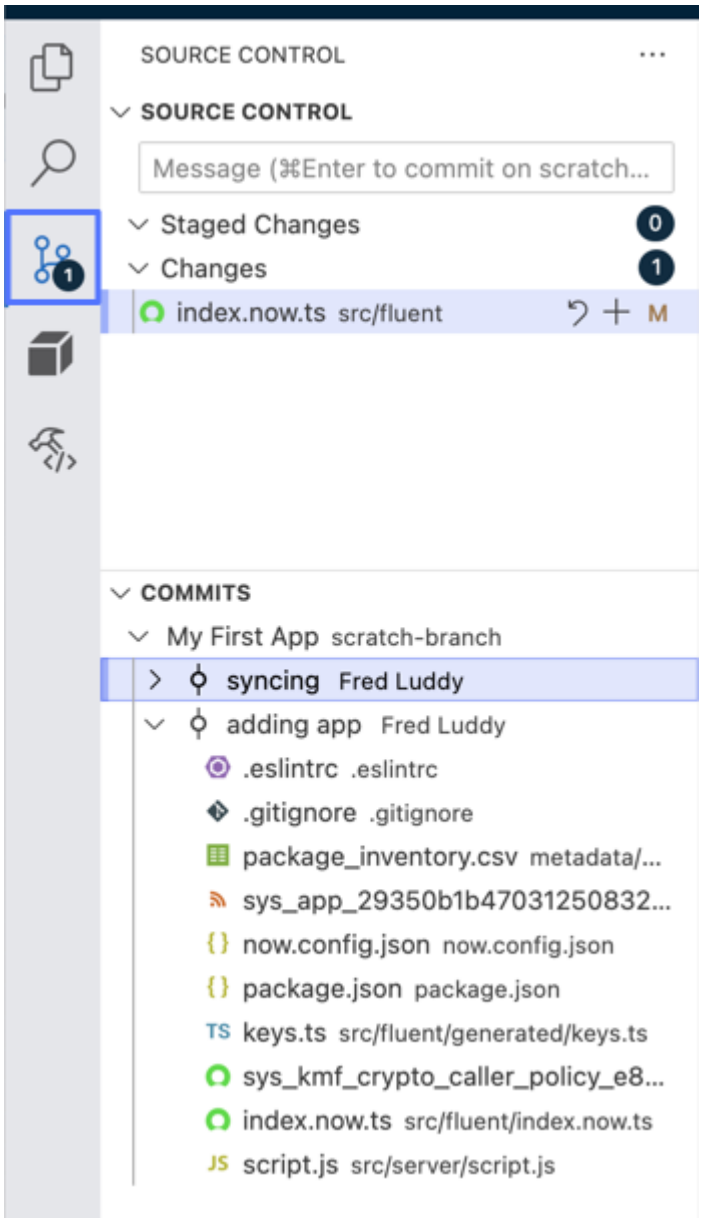
Search across files in the active workspace. The Search view supports using regular expressions, find and replace, and advanced search options.

To search for files by name, open the search box with the following keyboard shortcuts:

- Windows: Ctrl-P
- Mac: Cmd-P

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

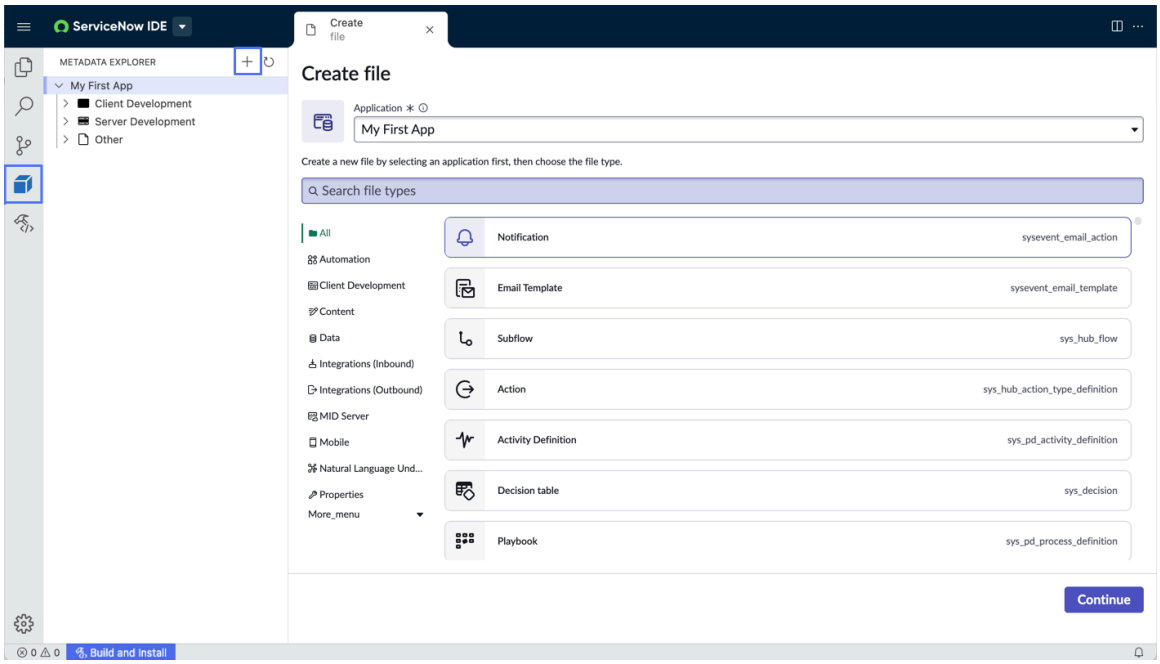
Source Control view



See changes to files grouped by application and the type of change. Stage, commit, and push changes to a remote repository and pull changes into your local repository. Create and checkout branches for your applications. In the Commits section, see the commit history for branches in the repository. To get started integrating with source control providers, see Integrating source control with the ServiceNow IDE.

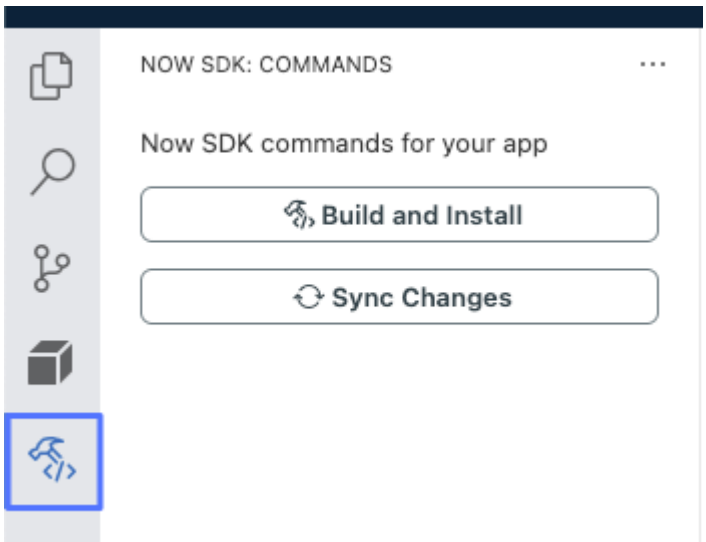
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

File Categories view



Create, update, and view application metadata in the most applicable embedded ServiceNow AI Platform user interface in the editor. If you need to create some application metadata outside of source code, you can create application files from the File Categories view. For more information, see Create an application file from the File Categories view.

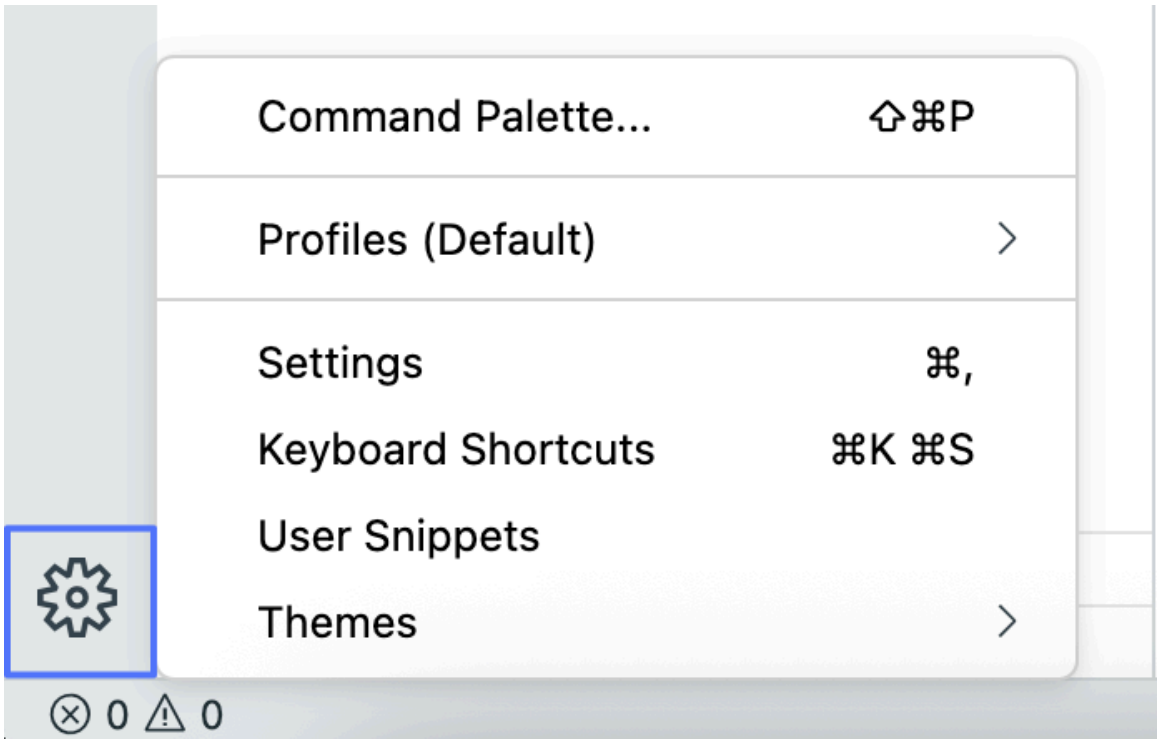
Now SDK view



Run commands from the ServiceNow SDK application packaging service to synchronize changes to application metadata or build and install an application. For more information, see Synchronizing applications in the ServiceNow IDE and Build and install an application in the ServiceNow IDE.

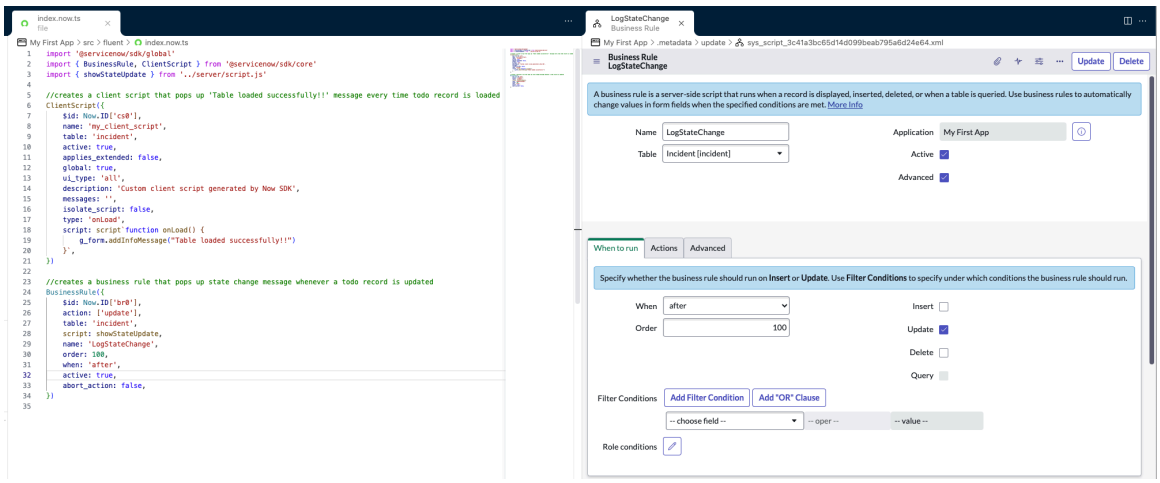
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Manage menu



Access and configure various aspects of the ServiceNow IDE user interface from the Manage menu. You can manage your settings, select a theme preference, view or edit keyboard shortcuts, and more.

Editor



Write and edit code using IntelliSense, custom language processing and validation for ServiceNow Fluent, side-by-side editing, and more.

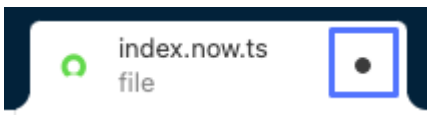
After building an application, review changes to ServiceNow Fluent source code as application metadata in embedded ServiceNow AI Platform user interfaces without leaving the ServiceNow IDE. Application metadata opens in the most applicable embedded ServiceNow AI Platform user interface. From links in the editor, you can open the File Categories view for metadata defined with a dedicated ServiceNow Fluent API.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
import '@servicenow/sdk/global'
import { DateColumn, StringColumn, Table } from '@servicenow/sdk/core'

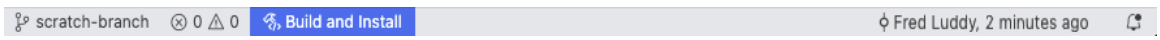
//creates todo table, with three columns (deadline, status and task)
Open Table
export const x_snc_hello_world_to_do = Table({
    name: 'x_snc_hello_world_to_do',
    schema: {
        deadline: DateColumn({ label: 'Deadline' }),
        state: StringColumn({
            label: 'State',
            choices: {
                ready: { label: 'Ready' },
                completed: { label: 'Completed' },
                inProgress: { label: 'In Progress' },
            },
        }),
        task: StringColumn({ label: 'Task', maxLength: 120 }),
    },
})
```

Switch between tabs and split the editor into groups. If a file has unsaved changes, the file tab includes a dot icon.



To get started developing applications in source code, see Developing applications with the ServiceNow IDE.

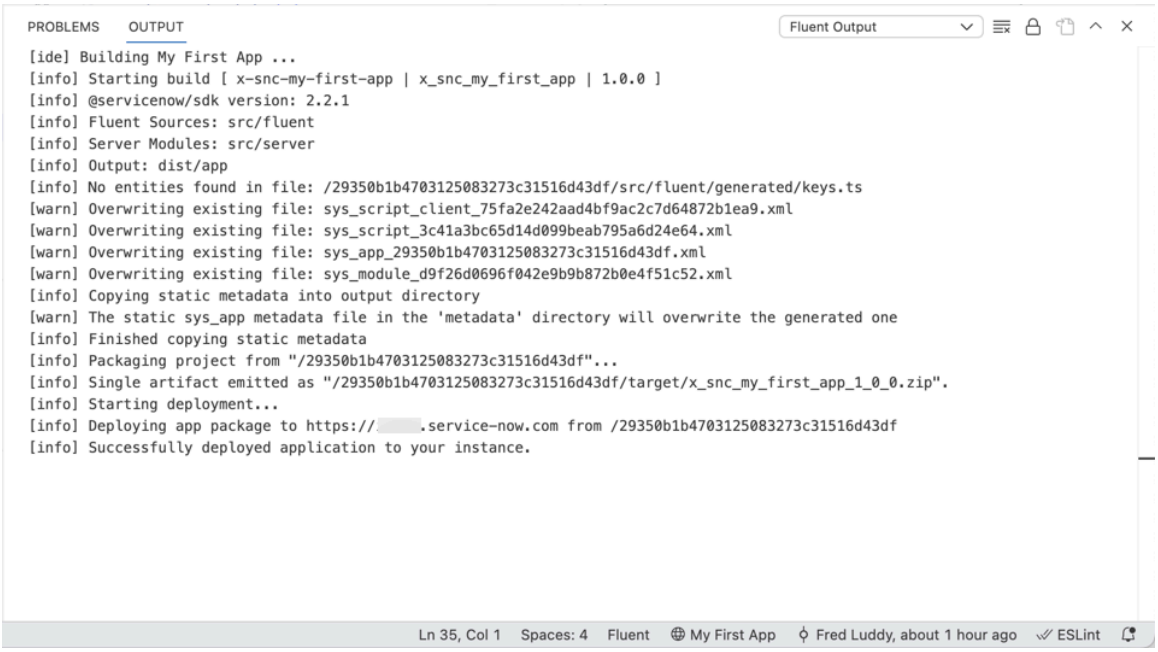
Status bar



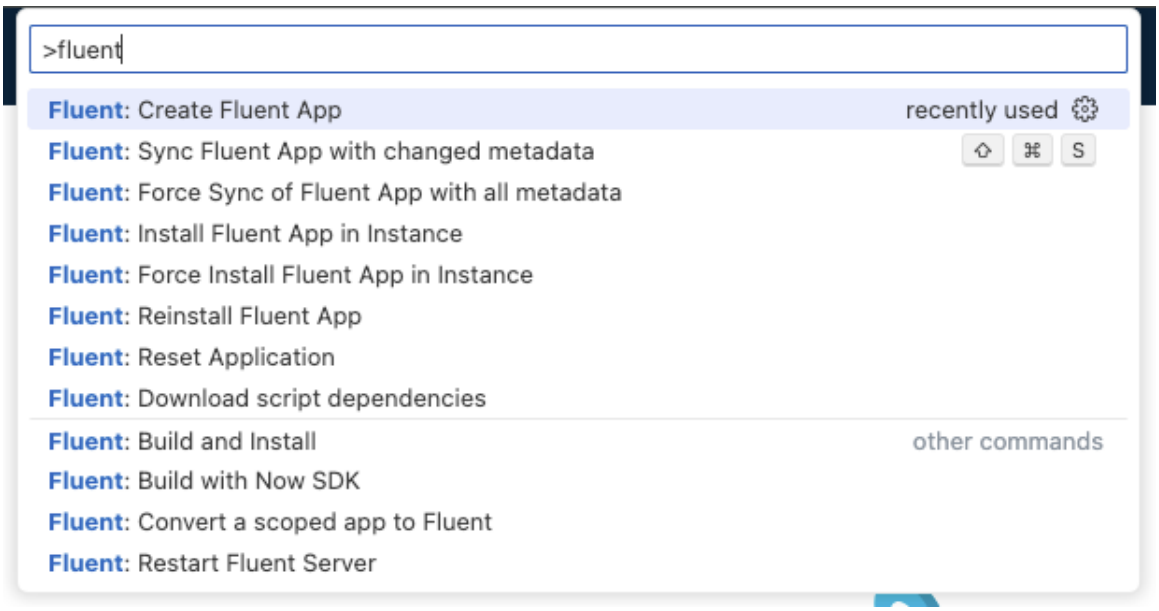
Get information about the state of your application and build applications. View the progress when creating, building, and syncing applications. In addition, the status bar displays the active branch, problems in your code, the active application scope, the active update set, and more.

From the status bar, the panel opens and displays more information about problems in your code and logging related to creating, building, and syncing applications.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Command palette



Run commands to configure the user interface, develop applications, manage files, use source control, access a list of keyboard shortcuts, and more. The command palette includes commands unique to the ServiceNow IDE and Visual Studio Code commands. For a list of commands specific to the ServiceNow IDE, see ServiceNow IDE commands.

To open the command palette, use one of the following keyboard shortcuts:

- Windows: Ctrl-Shift-P
- Mac: Cmd-Shift-P

To close the command palette, press the Escape key.

Getting started: Create your first application in the ServiceNow IDE

Get started in the ServiceNow IDE by creating an application that you can develop in source code, pushing it to a remote repository, and cloning it on another instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The ServiceNow IDE is an implementation of Visual Studio Code for the Web on the ServiceNow AI Platform. With the ServiceNow IDE, you can get started building applications quickly using familiar tools and industry-standard development practices. For an overview of the ServiceNow IDE user interface, see ServiceNow IDE user interface.

Tutorial overview

In this tutorial, you complete the following tasks to familiarize yourself with developing a new application in source code from the ServiceNow IDE:

- Create an application that supports development in source code.
- Configure basic authentication and connect your application in the ServiceNow IDE to a remote Git repository and then push your application to the repository.
- Create a simple table for a to-do list using ServiceNow Fluent APIs.
- Install a third-party library and use its code in a JavaScript module.
- Clone the application from the remote Git repository onto another instance.

Afterward, you should be ready to take the basic application you create and continue learning to develop it into a more complex application that meets your requirements.

Before you begin

You should have access to the following items to complete this tutorial:

- A non-production instance on the Xanadu release or later with ServiceNow IDE version 2.0.4 or later installed. To complete the last part of the tutorial by cloning the application to another instance, you need access to a second non-production instance. Personal developer instances (PDIs) are supported and can be obtained from developer.servicenow.com. For more information, see Install or update the ServiceNow IDE.
- A user with the admin role on both instances.
- A Git provider such as GitHub, GitLab, Bitbucket, or Azure Repos. This tutorial demonstrates using GitHub.

Tutorial part 1: Create an application in the ServiceNow IDE

Create an application that you can develop in source code in the ServiceNow IDE.

Before you begin

Role required: admin

About this task

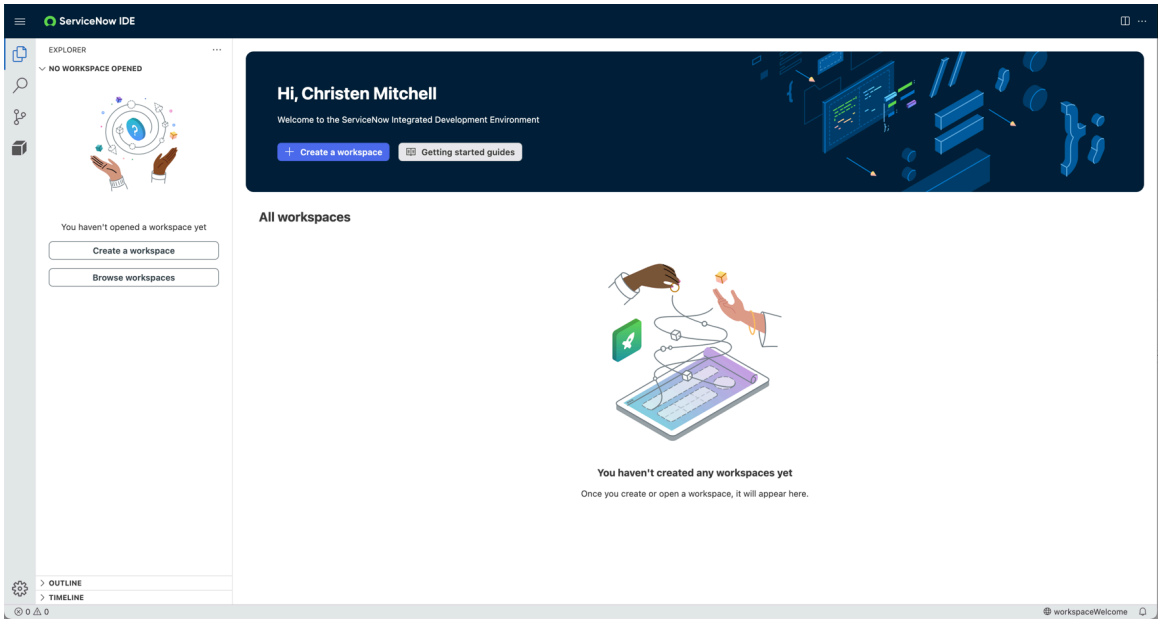
In the ServiceNow IDE, you can develop applications in source code using ServiceNow Fluent to define application metadata [sys_metadata]. You can also create JavaScript modules and use third-party libraries to organize and reuse code within an application scope. For an application to support development in source code, you must create an application or convert an existing application using the ServiceNow IDE or ServiceNow SDK. In this tutorial, you create a scoped application from the ServiceNow IDE.

Procedure

1. Navigate to **All** > **App Development** > **ServiceNow IDE**.

   Before you can create an application, you must create a workspace in which you can add applications and navigate through them from one place.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



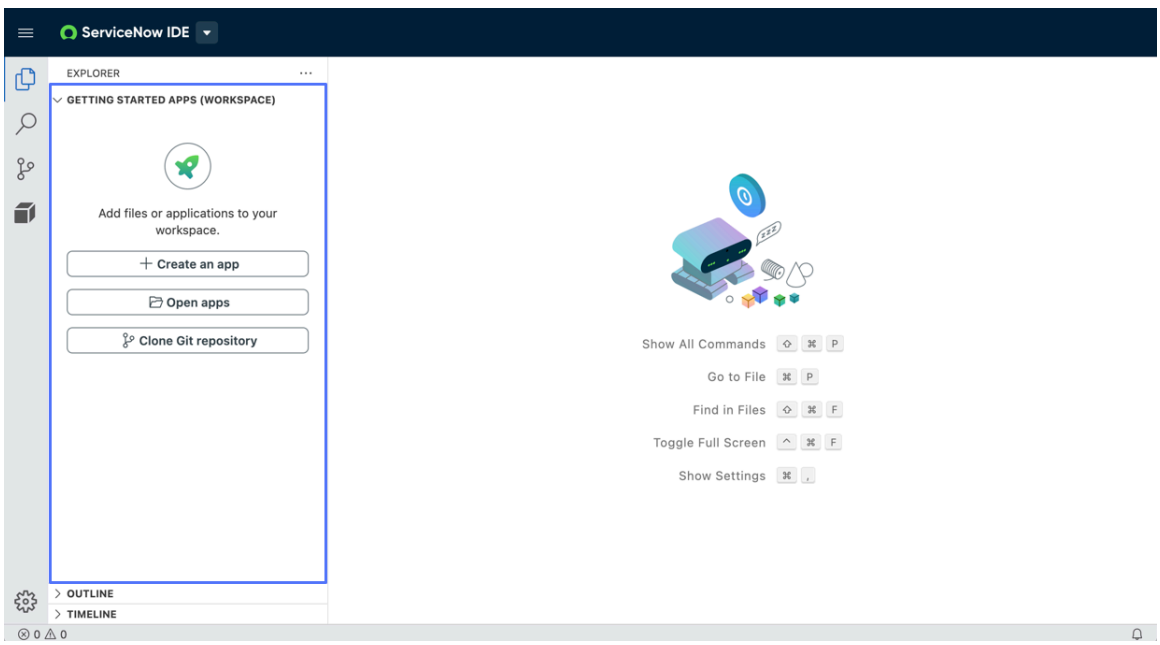
Workspaces are specific to a user, and you can create multiple workspaces to group different sets of applications. Applications can be added or removed from a workspace at any time. Workspaces in the ServiceNow IDE are based on workspaces in Visual Studio Code.

2. Create a workspace.

   a. Select **Create a workspace**.

   b. Enter a name for the workspace and press Enter.

   c. Enter a description for the workspace and press Enter.
   The workspace becomes the active workspace but doesn't contain any applications yet.



Tip: To switch the active workspace, you can browse and select other workspaces from the ServiceNow IDE home page or from the command palette with the `Workspaces: Browse Workspaces` command.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

3. Create an application in the workspace.

   a. Select **Create an app**.

   b. Enter a name, such as `Hello World`, and press Enter.

   c. Enter a description for the application and press Enter.

   d. Enter a scope, such as `x_snc_hello_world`, and press Enter.

   > **Important:** The scope name must be unique on the instance, begin with x_<prefix>, and be 18 characters or fewer. For more information, see Namespace identifier.

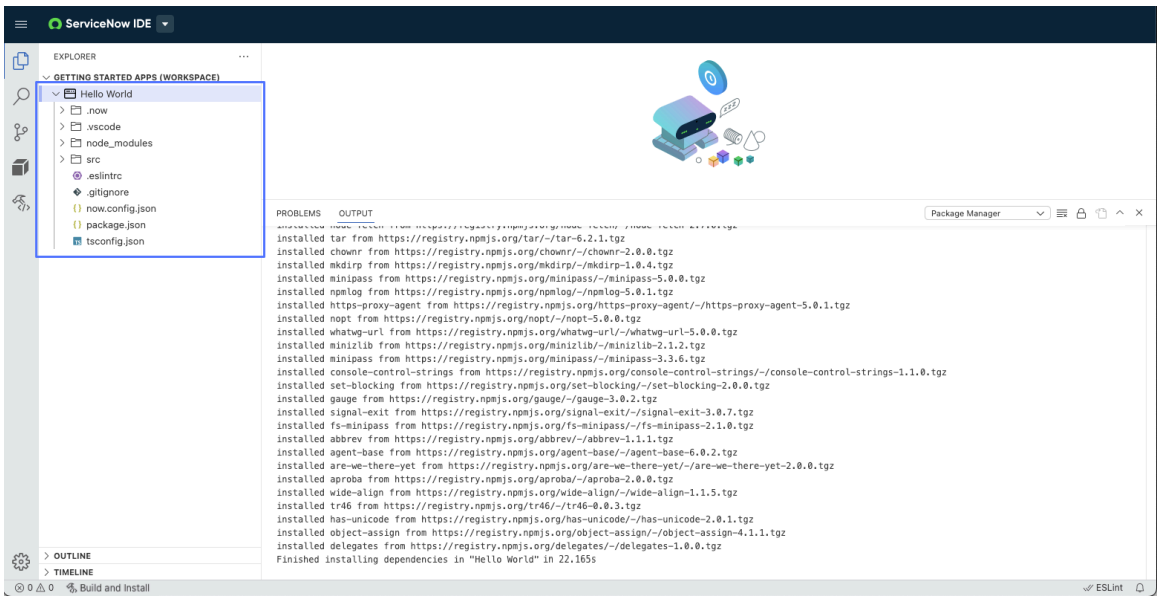
   e. Enter a package name, such as `x-snc-hello-world`, and press Enter.
   The package name must adhere to Node Package Manager (npm) package naming standards.

   f. Select the **A basic application using NowSDK and TypeScript** template to use TypeScript to create JavaScript modules.

   The application templates define the default application structure.

   > **Note:** TypeScript uses static typing and type annotations to support developers catching errors earlier while writing code. If you'd prefer to get started with JavaScript instead, select **JavaScript**.

   Your application is added to your workspace. From the File Explorer view ( ), you can navigate through the application files.

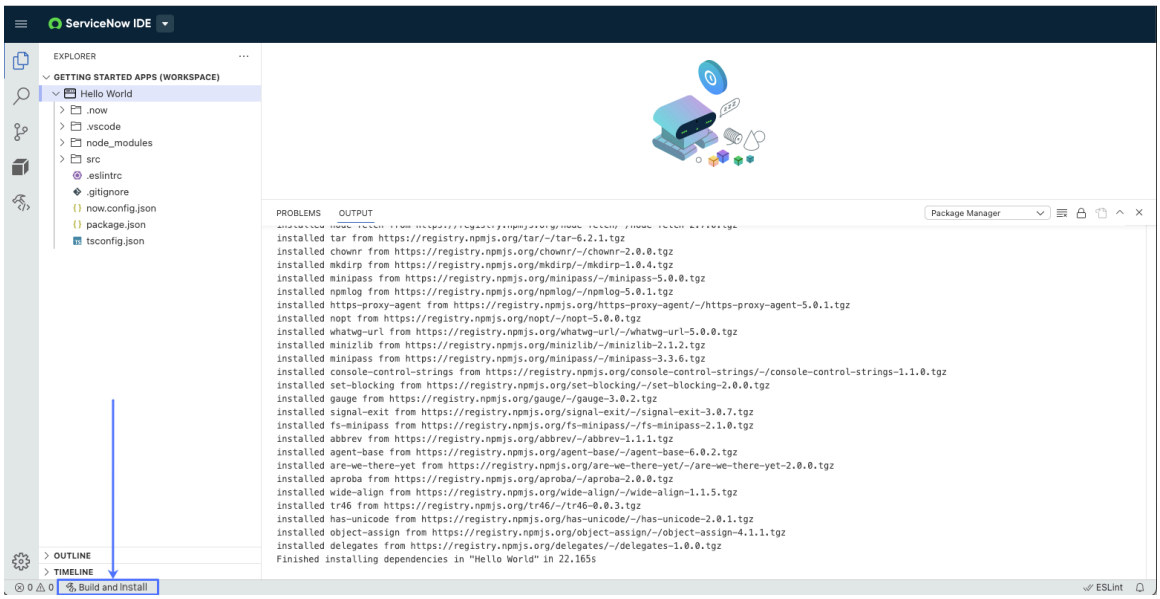


The `package.json` and `now.config.json` files define the application structure, which is similar to that of Node.js applications or Node Package Manager (npm) packages. A sample ServiceNow Fluent file (`index.now.ts`) and JavaScript module (`script.js` or `script.ts`) were created in the `src` directory, and dependencies were installed into the `node_modules` directory. The application is also added to the Custom Application [sys_app] table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

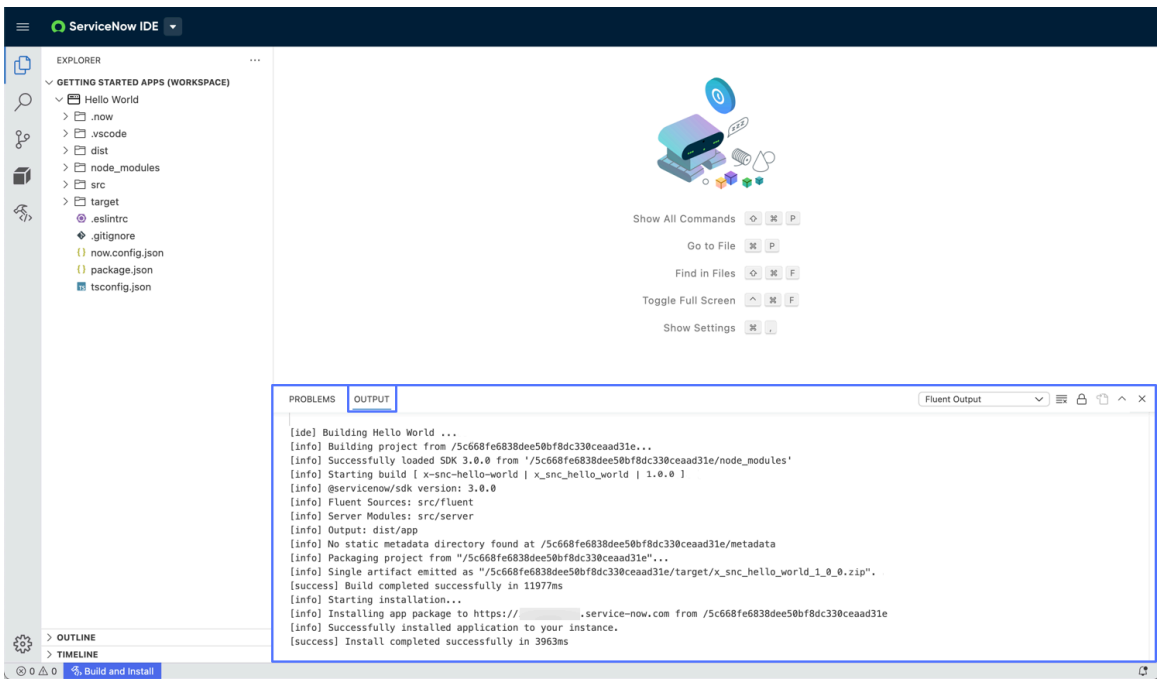
Next, build and install the application to make it available for development across the instance.

g. From the Status Bar at the bottom of the ServiceNow IDE, select **Build and Install**.



Building and installing the application compiles its ServiceNow Fluent code and JavaScript or TypeScript module code into Application Files [sys_metadata] and EcmaScript Modules [sys_module] on the instance, respectively. Build artifacts in the `dist/app` directory are packaged into an installable `.zip` file in the `target` directory.

Logs in the Output panel indicate the status of the build and if the application is installed successfully. If either process fails, review the logs to identify any issues.



For information about the application structure, see the Application structure section of the Building applications in source code topic.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

What to do next

Continue to Tutorial part 2: Initialize a repository for your application.

Tutorial part 2: Initialize a repository for your application

Initialize a local Git repository for an application in the ServiceNow IDE and push it to a remote Git repository in GitHub to manage the application in source control.

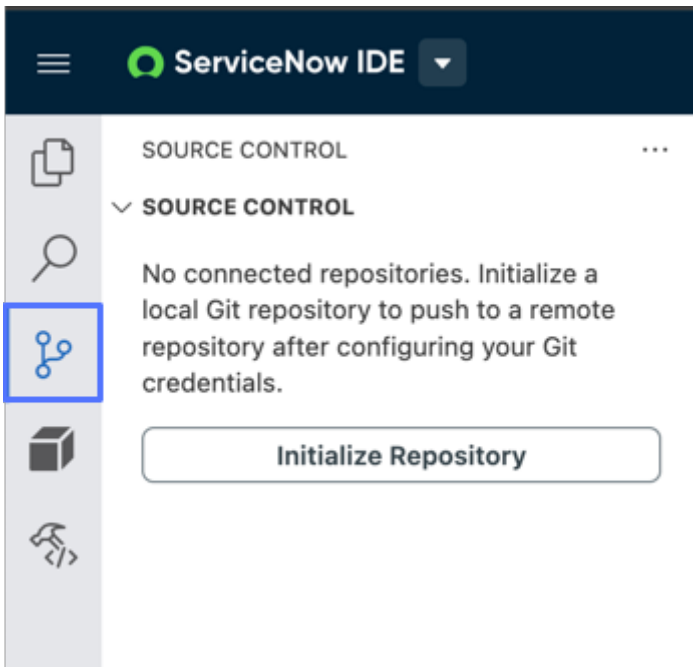
Before you begin

Complete Tutorial part 1: Create an application in the ServiceNow IDE.

Role required: admin

Procedure

1. From the Activity Bar, select the Source Control view ( ).



2. Initialize a Git repository that's local to the ServiceNow IDE.
   a. Select **Initialize Repository**.
   b. Press Enter to use **main** as the default branch name.

   A local repository is initialized for your application, and the application files that are tracked in source control are in the Untracked Changes list. Application files and directories specified in the `.gitignore` file aren't tracked in source control.

   Tip: If you select a file in the list, you can view a comparison of the current and previous version of the file in the Diff editor.

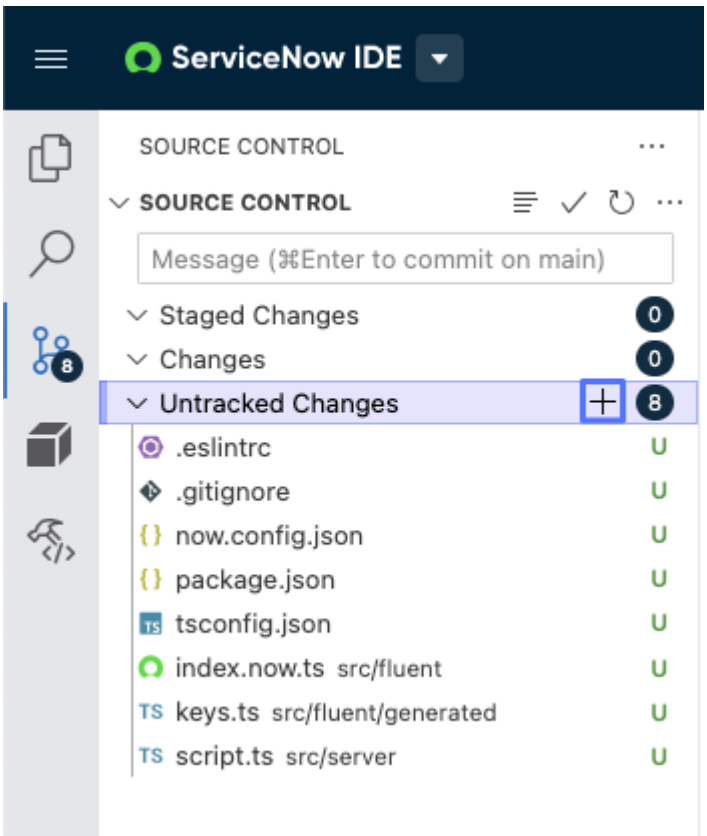
   Before you can push your application to a remote Git repository, you must stage and commit your changes in the local Git repository.

3. Stage and commit your changes.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. From the Untracked Changes list, select the Stage All Untracked Changes icon (+).

Example



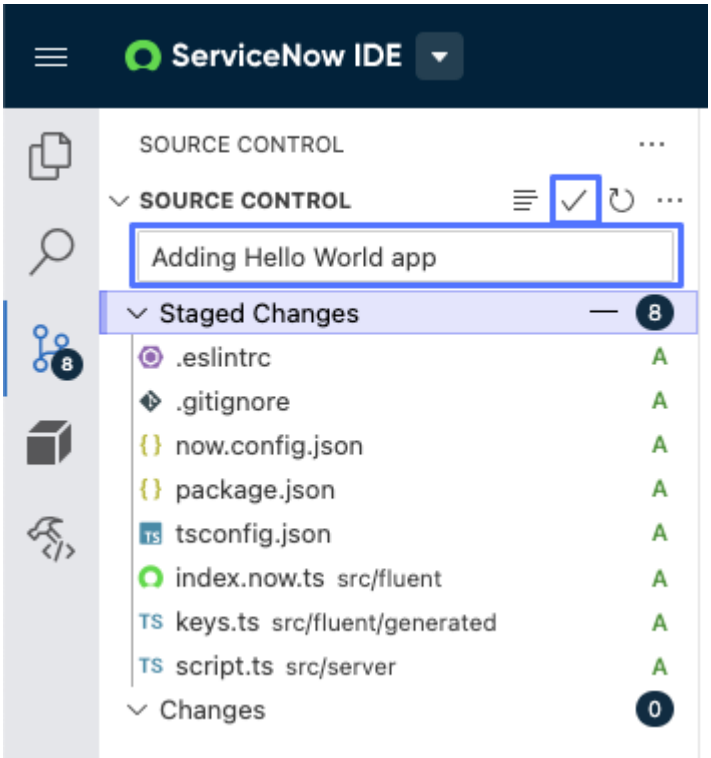
Your changes move to the Staged Changes list.

b. In the message box, enter a commit message.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

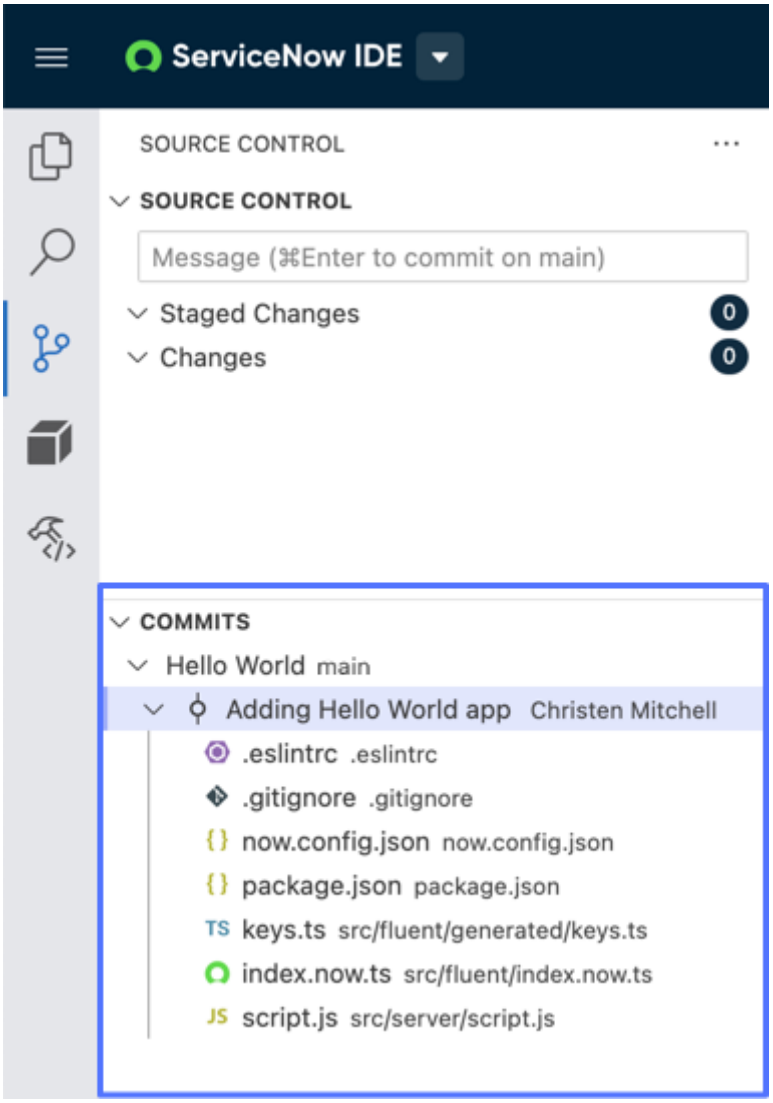
c. Select the Commit icon ( ).

Example



Now that you have committed changes, you can expand the Commits section to see your commit history and details about each commit.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



To push your changes to a remote repository that other developers can access, you need to create a repository with a Git provider like GitHub and generate a personal access token.

Note: The following steps demonstrate using GitHub and basic authentication as an example, but you could use another Git provider of your choosing or OAuth 2.0 authentication. For more information, see Integrating source control with the ServiceNow IDE.

4. Create a remote repository.

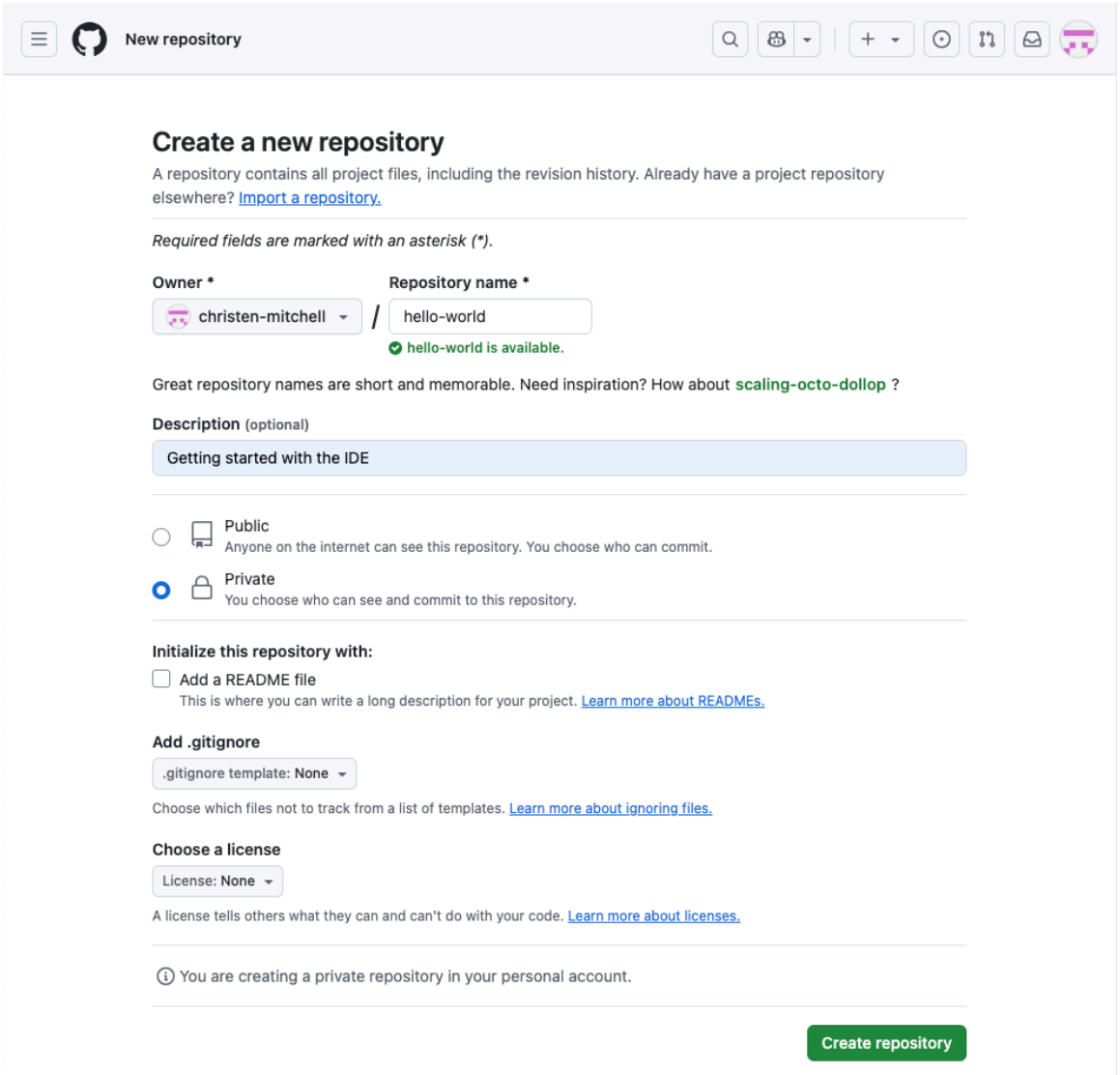
   a. Log in to your GitHub account.

   b. Select the Create new menu (+) and select **New repository**.

   c. On the New repository form, fill in the fields.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

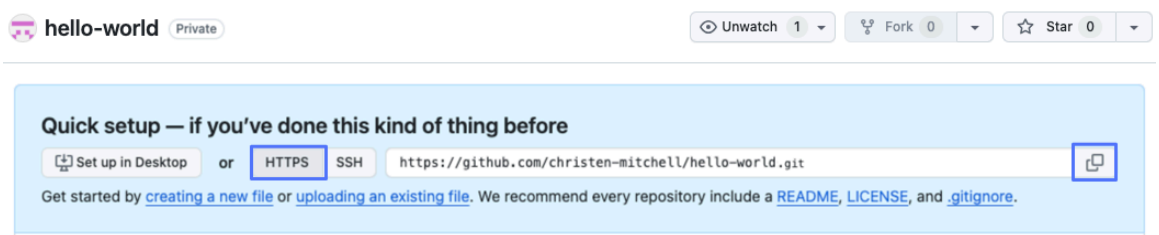
Example



d. Select **Create repository**.

e. From your repository, copy the HTTPS URL and paste it somewhere that you can easily access it again for use in the following steps.

Example



5. Create a personal access token.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

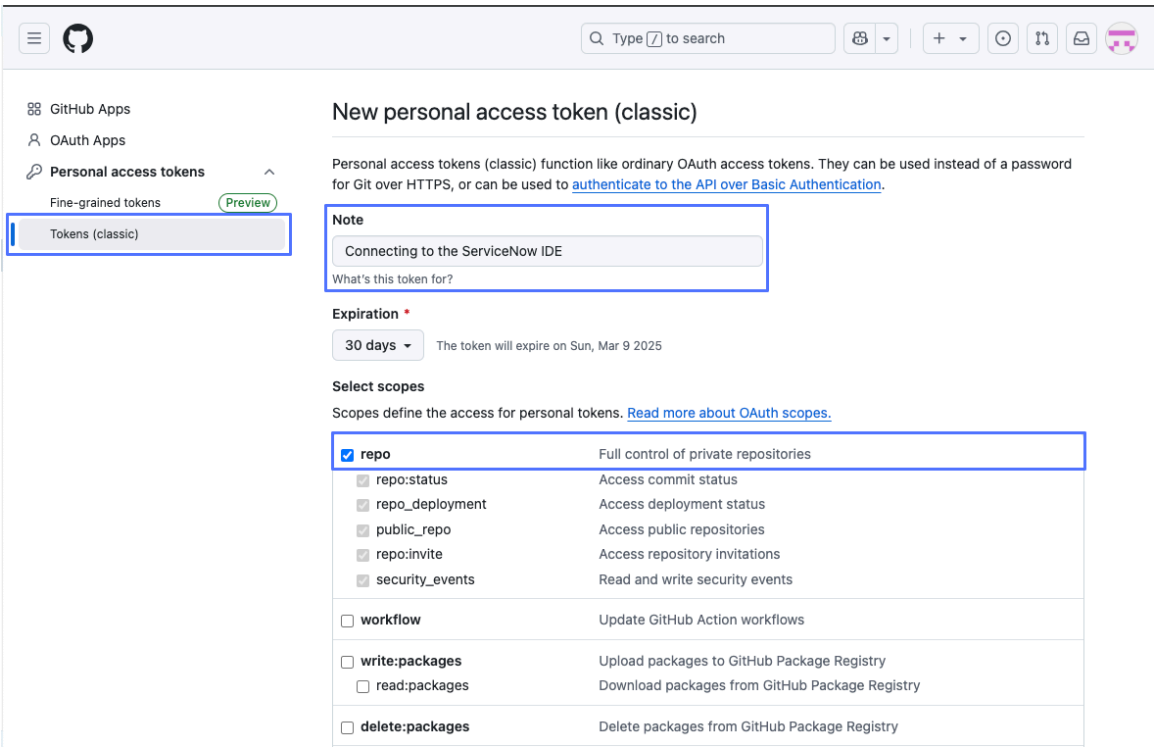
a. From your user profile in GitHub, select **Settings**.

b. From the settings categories, select **Developer Settings**.

c. From the Personal access tokens menu, select **Tokens (classic)**.

d. From the Generate new token menu, select **Generate new token (classic)**.

e. On the New personal access token (classic) form, select the **repo** scope and fill in the remaining required fields.
The **repo** scope is required to connect to the ServiceNow IDE.

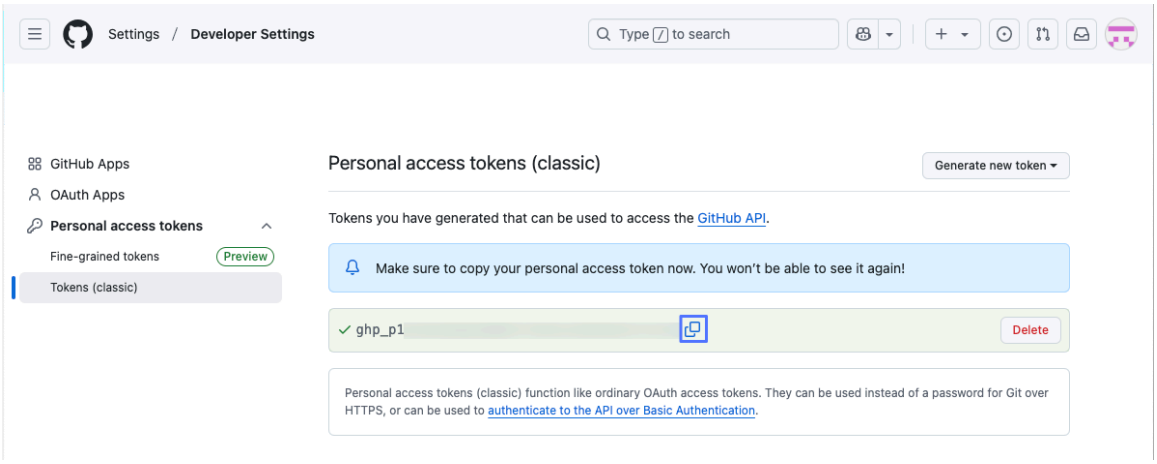


f. Select **Generate token**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

g. Copy the token and paste it somewhere that you can easily access it again for use in the following steps so you don't need to regenerate it.

Example



6. Configure basic authentication in the ServiceNow IDE to connect to the remote repository.

a. Navigate to the ServiceNow IDE.

b. Use one of the following keyboard shortcuts to open the command palette:

- Windows: Ctrl-Shift-P
- Mac: Cmd-Shift-P

c. Enter `Git: Set IDE Git credentials` and press Enter.

d. From the New Git credential form, select **Basic auth**.

e. On the form, fill in the fields.

New Git credential form

Field	Description
Git repository URL	The HTTPS URL of the Git repository you created in GitHub. For example: `https://github.com/<owner>/hello-world.git`.
Git username	Your GitHub user name.
Personal access token	The personal access token that you generated from GitHub.

f. Select **Submit**.

Now that your authentication credentials are configured, you can push your application to the remote repository.

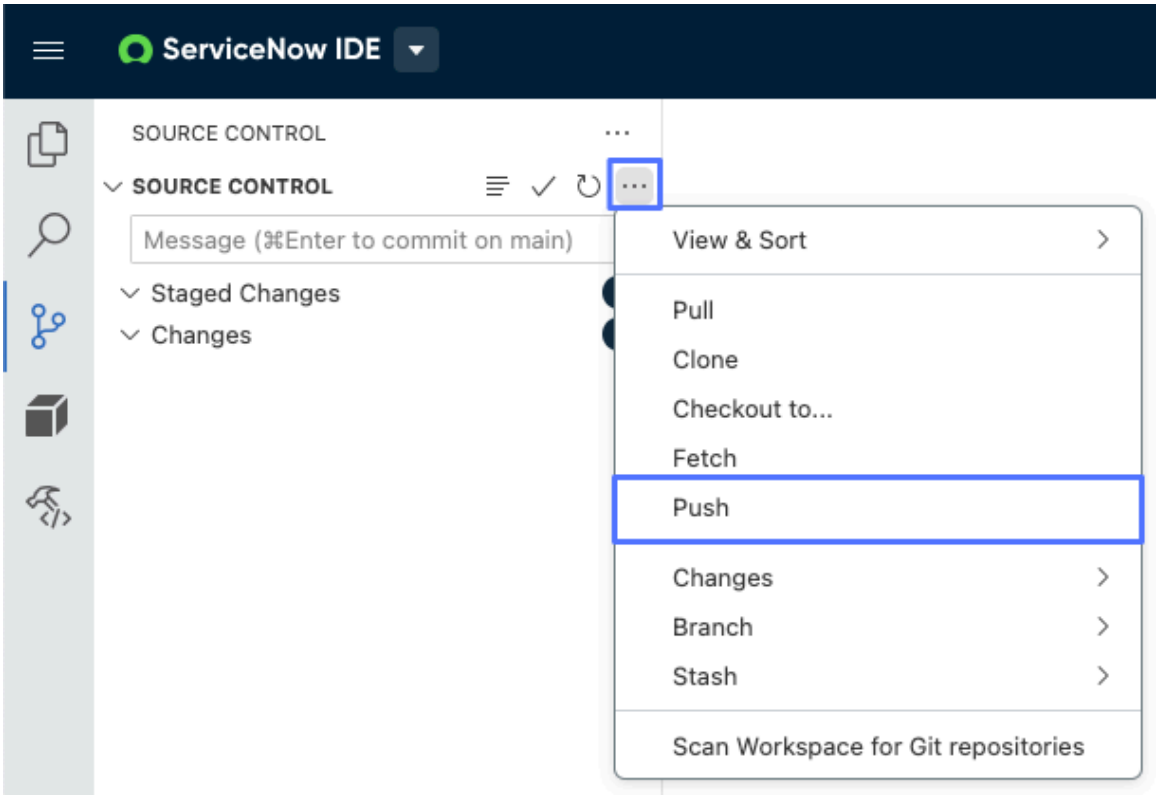
Tip: If you need to manage existing Git credentials, use the `Git: Manage Git credentials` command from the command palette.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

7. Push your changes to the remote repository.

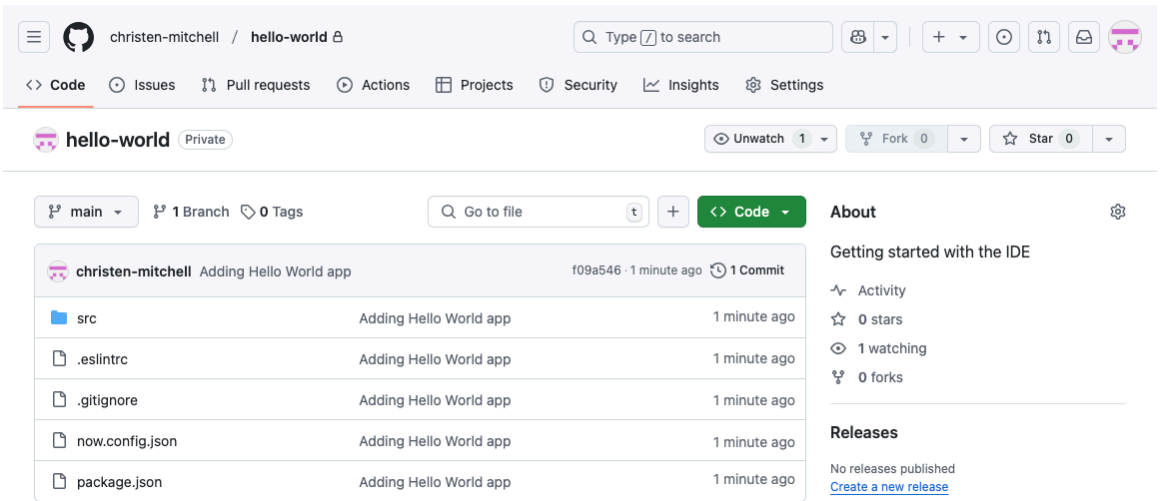
a. Select the More actions menu icon (···) and select **Push**.

Example



b. Enter the HTTPS URL of the remote Git repository and press Enter.

The application is added to the remote repository in GitHub, and changes to the application across users are tracked remotely.



Result

You can check out or create branches in the repository and push changes to the remote repository. For more information, see Using source control in the ServiceNow IDE.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: An application on an instance can be connected to only one repository at a time. To update the remote repository an application is connected to, you can use the `Git: Update remote origin` command from the command palette and enter a different remote repository URL.

What to do next
Continue to Tutorial part 3: Define a table in ServiceNow Fluent code.

Tutorial part 3: Define a table in ServiceNow Fluent code

Create a table and reference it in sample script definitions using the ServiceNow Fluent APIs.

Before you begin
Complete Tutorial part 2: Initialize a repository for your application.

Role required: admin

About this task
ServiceNow Fluent is a domain-specific language (DSL) based on TypeScript for defining the metadata files [sys_metadata] that make up applications and includes APIs for the different types of metadata, such as tables, roles, ACLs, business rules, and Automated Test Framework tests. You use objects in the ServiceNow Fluent APIs to define metadata in files with the `.now.ts` extension. The ServiceNow IDE has language processing and validation for ServiceNow Fluent APIs and applications by default. For more information about ServiceNow Fluent, see ServiceNow Fluent.

In this example, you create a simple table for a to-do list using objects in the ServiceNow Fluent Table API. Then, you update the sample code for business rule and client script definitions to reference the new table. Lastly, you review your changes from the File Categories view. For more information about the Table API, see Table API - ServiceNow Fluent.

Procedure

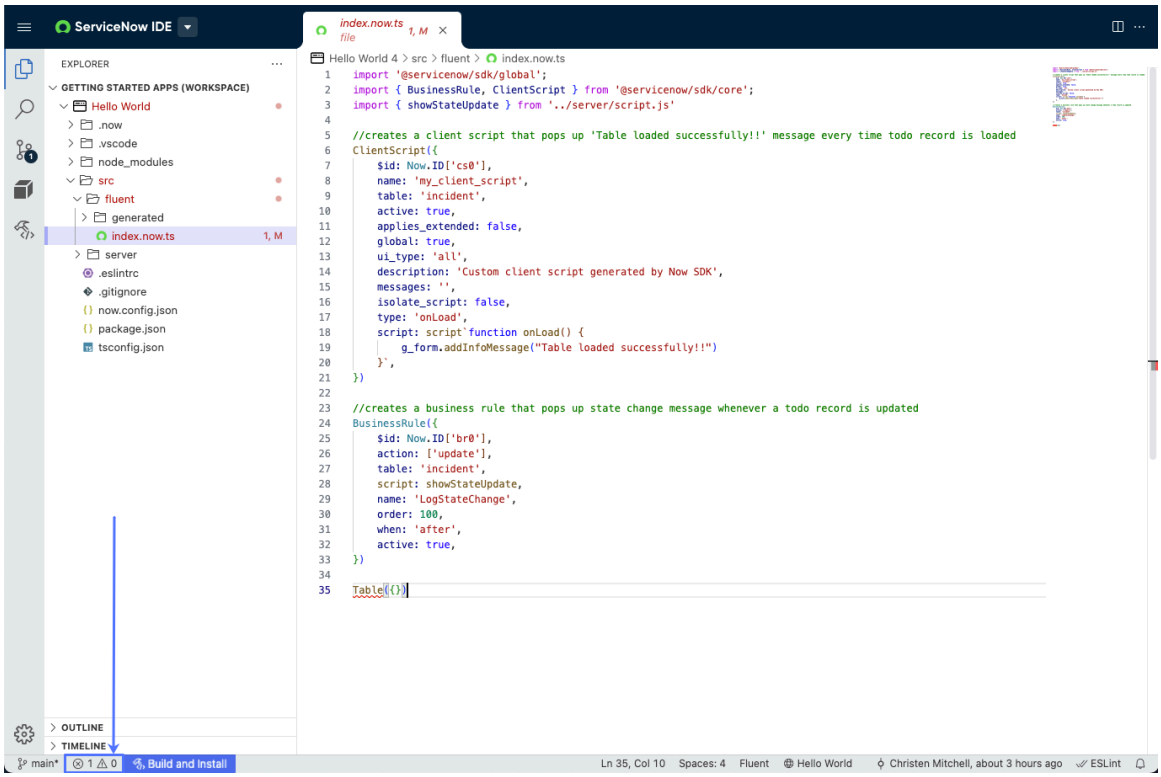
1. From the Activity Bar, select the File Explorer view ( ).
2. Navigate to the `src/fluent` directory in your application.
3. Open the `index.now.ts` sample file.

   Tip: You can write ServiceNow Fluent code in a single file or as many `.now.ts` files as you want and organize files in directories within the `fluent` directory.

4. On a new line at the end of the file, enter `Table({})` to add the Table object.
5. From the Status Bar, select the diagnostics icon (⊗ 1 ⚠ 0) to open the Problems panel and check for issues in the code.

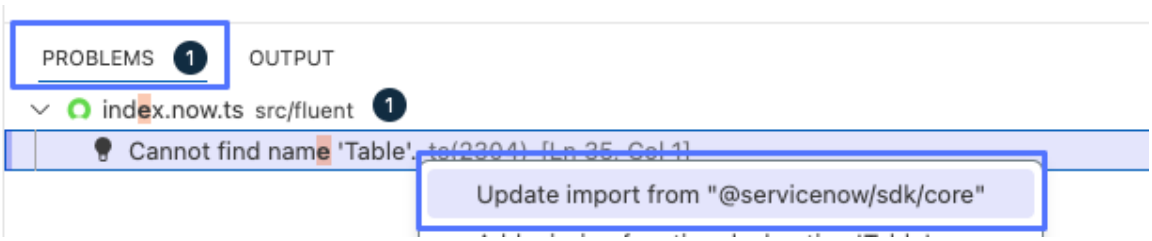
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example



6. Right-click the error that appears and select the **Update import from "@servicenow/sdk/core"** quick fix.

Example



In line 2, the *Table* object is added to the list of imports from `@servicenow/sdk/core`:

```
import { BusinessRule, ClientScript, Table }
 from '@servicenow/sdk/core'
```

Tip: After fixing this issue, you can close the Problems panel while you proceed through the following steps. In a later step, you return to it if any issues remain.

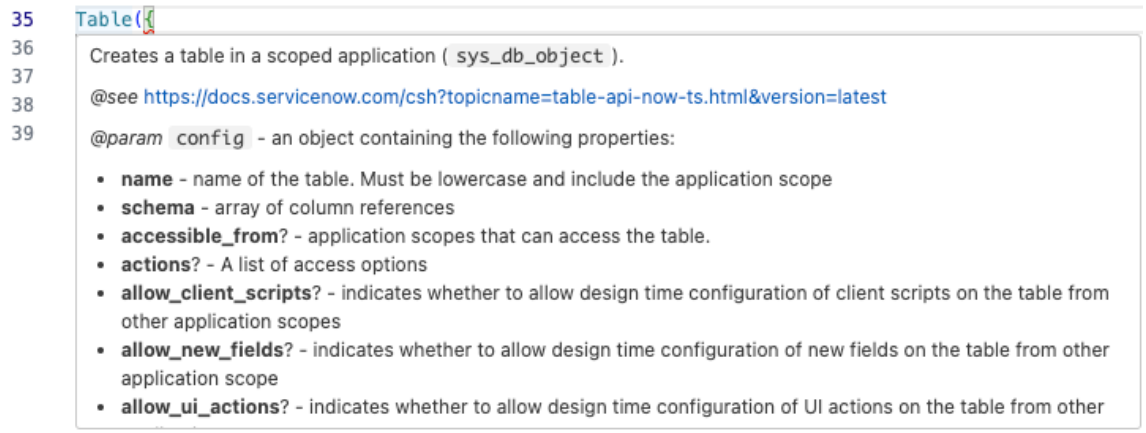
7. In the *Table* object, add the following properties.

   - name: The table name must begin with the application scope and use all lowercase letters in the following format: `<scope>_<name>`. You can find the scope in the `now.config.json` file for the application.
   - label: The label should be unique and appears for the table in list and form views.
   - extends: The name of another table on which the table is based.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
Table({
    name: 'x_snc_hello_world_to_do', //ensure that the name begins with the correct scope (<scope>_<name>)
    label: 'To-do Items',
    extends: 'task',
})
```

Tip: Hover over an object to see its in-product documentation.



8. For type-ahead support when defining columns in the table, before the *Table* object, add an exported variable with the same name as the *name* property.

```
export const x_snc_hello_world_to_do = Table({
    name: 'x_snc_hello_world_to_do',
    label: 'To-do Items',
    extends: 'task',
})
```

9. In the *Table* object, add the *schema* property to define columns in the table.

```
export const x_snc_hello_world_to_do = Table({
    name: 'x_snc_hello_world_to_do',
    label: 'To-do Items',
    extends: 'task',
    schema: {
        //define columns here
    }
})
```

The schema property is an array of *Column* objects. There are many types of columns based on the field type. Column objects use the format *<Type>Column* where *<Type>* is the field type.

Use the following details to define three columns in the table: Deadline, Matrix, and Task. Refer to the Column object documentation to help you configure each column.

Column name	Details
deadline	○ Label: Deadline ○ Type: Date/Time

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Column name	Details
matrix	○ Label: Matrix ○ Type: String ○ Choices: ▪ Label: Urgent and Important ▪ Label: Important but Not Urgent ▪ Label: Urgent but Not Important ▪ Label: Neither Urgent nor Important
task	○ Label: Task ○ Type: String ○ Max length: 120

Example
With these details, the *schema* property should look similar to this example. The keys that you use for the choices can be any string.

```
export const x_snc_hello_world_to_do = Table({
    name: 'x_snc_hello_world_to_do',
    label: 'To-do Items',
    extends: 'task',
    schema: {
        deadline: DateColumn({ label: 'Deadline' }),
        matrix: StringColumn({
            label: 'Matrix',
            choices: {
                do: { label: 'Urgent and Important' },
                decide: { label: 'Important but Not Urgent' },
                delegate: { label: 'Urgent but Not Important' },
                delete: { label: 'Neither Urgent nor Important' },
            },
        }),
        task: StringColumn({ label: 'Task', maxLength: 120 }),
    },
})
```

10. In line 2, add imports for the *DateColumn* and *StringColumn* objects.

```
import { BusinessRule, ClientScript, Table, DateColumn,
 StringColumn } from '@servicenow/sdk/core'
```

11. Update the existing business rule and client script definitions to reference the table you created.
    a. In the *ClientScript* object, change the value of the *table* property to the table name (`x_snc_hello_world_to_do`).
    b. Repeat the previous step for the *BusinessRule* object.
12. If the diagnostics icon (⊗ 1 ⚠ 0) shows any errors or warnings, select it to open the Problems panel and review the diagnostic messages and quick fixes to resolve them.
13. Save your changes using one of the following keyboard shortcuts.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Windows: Ctrl-S
- Mac: Cmd-S

Note: If you have unsaved changes in a file, a dot icon appears on the file tab.

14. From the Status Bar, select **Build and Install**.

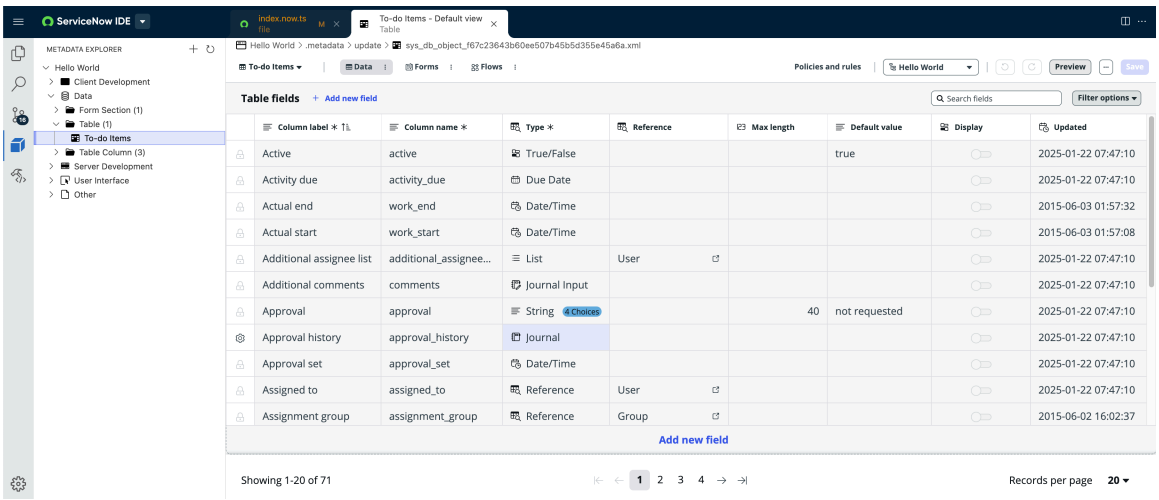
    If the installation completes successfully, the updated ServiceNow Fluent source code is compiled into Application Files [sys_metadata] on the instance.

15. Review your changes as metadata.

    a. From the Activity Bar, select the File Categories view or Apps view.

    b. Select your application to expand it.

    c. Navigate to **Data** > **Table** and select **To-do Items**.
    The table opens in Table Builder.



16. **Optional:** Edit the metadata and synchronize your changes into the source code.
    From the File Categories view, you can simulate another user editing the metadata outside of the source code to see changes transformed back into the code you added.

    a. In Table Builder, search for the Task field and change the Column label from "Task" to "To Do".

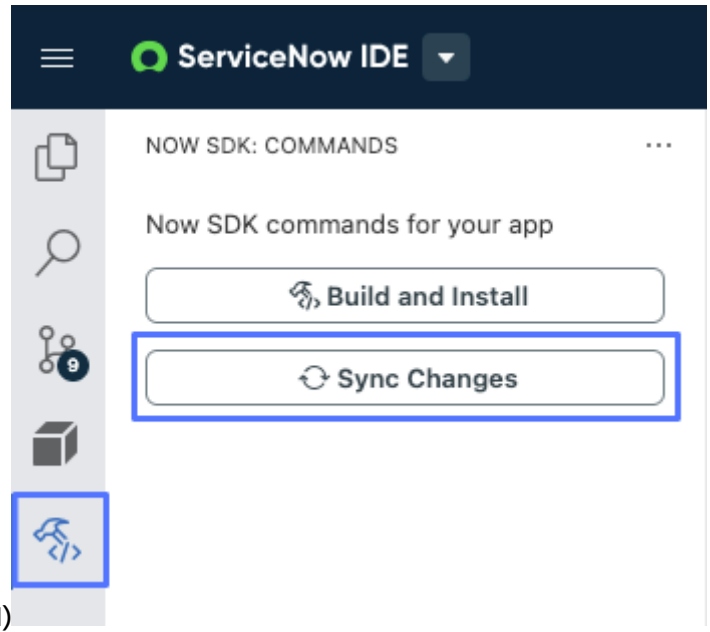
    b. Select **Save**.

    c. From the Activity Bar, select the Now SDK view ( ).

    d. Select **Sync Changes**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example



(Optional)

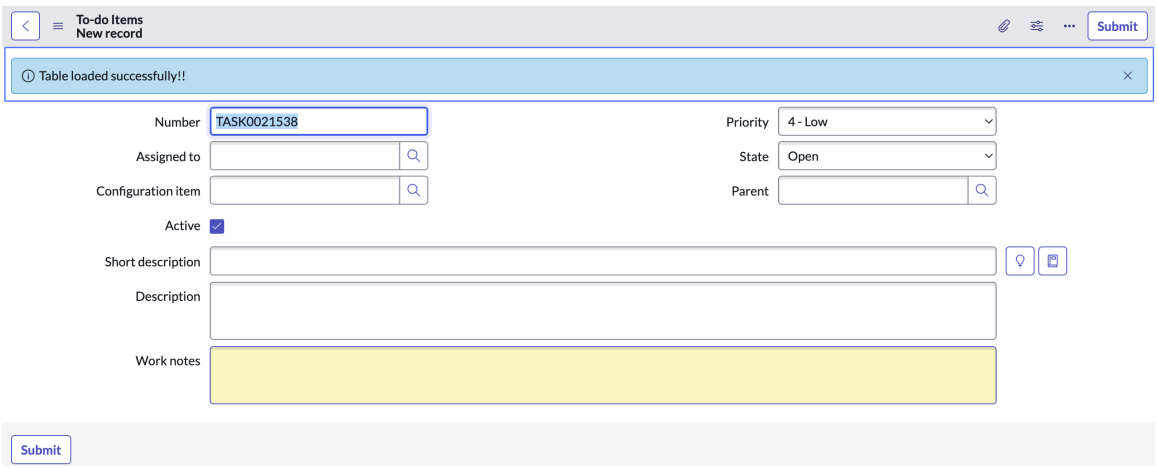
In the `index.now.ts` file, you should see the *`label`* property of the task column changed to `'To Do'`.

```
task: StringColumn({ label: 'To Do', maxLength: 120 }),
```

Result

You have created your first application metadata using ServiceNow Fluent APIs. The To-do Items [x_snc_hello_world_to_do] table can be modified in source code by other ServiceNow IDE users or from other ServiceNow AI Platform user interfaces.

From the ServiceNow AI Platform, you can navigate to the list view of the table by entering `x_snc_hello_world_to_do.list` in the navigation filter. Because you updated the client script definition to run on the To-do Items [x_snc_hello_world_to_do] table, if you select **New** to add a record to the table, the message from the client script appears when the record loads.



What to do next

Continue to Tutorial part 4: Install and use a third-party library.

Tutorial part 4: Install and use a third-party library

Install a third-party library from Node Package Manager (npm) and use it in a JavaScript module.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

Complete Tutorial part 3: Define a table in ServiceNow Fluent code.

Role required: admin

About this task

Installing third-party libraries allows you to use existing open-source functionality in JavaScript modules to accelerate application development. Then you can refer to JavaScript modules that call third-party code from server-side script definitions in your source code, such as the business rule in the `index.now.ts` file.

In this example, you install a Lodash library to get common JavaScript utilities and methods. You use one of these methods, *`snakeCase`*, in the *`showStateUpdate`* function in a sample JavaScript module to display a message string in snake case, which separates words with underscores instead of spaces. In the `index.now.ts` file, the sample business rule is configured to use the *`showStateUpdate`* function for its script and to run after a record is updated in the To-do Items [x_snc_hello_world_to_do] table.

Procedure

1. Install the *`snakeCase`* method from the Lodash library in your application.

   a. From the Activity Bar, select the File Explorer view ( ).

   b. Open the `package.json` file for the application.

   c. After the `devDependencies` field, add the `dependencies` field with the package name and version of the library.

   Example

   ```
   },
        "dependencies": {
             "lodash.snakecase": "4.1.1"
        }
   }
   ```

   d. For applications that use TypeScript in JavaScript modules, add the lodash.snakecase types to the `devDependencies` field to get the type annotations for the library.

   ```
   "devDependencies": {
             "@types/lodash.snakecase": "4.1.1",
             "typescript": "5.5.4",
             "@servicenow/sdk": "2.2.4",
             "@servicenow/glide": "26.0.1",
             "eslint": "8.50.0",
             "@servicenow/eslint-plugin-sdk-app-plugin": "2.2.4"
        }
   ```

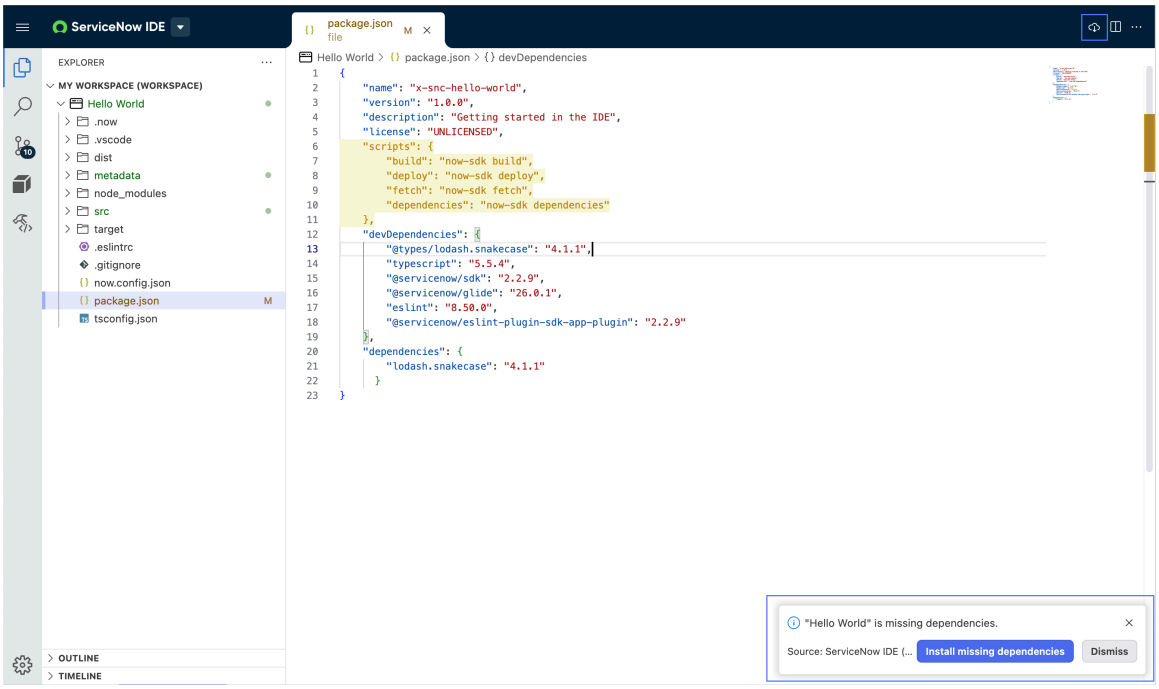
   e. Save your changes.

   f. When prompted, select **Install missing dependencies**.

   Tip: You can also select the Install Dependencies icon ( ) or use the `Package Manager: Install Dependencies` command from the command palette.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example



Libraries are installed as modules in the `node_modules` directory.

2. Use code from the Lodash library in a JavaScript module in your application.

   a. Navigate to the `src/server` directory in your application.

   b. Open the `script.ts` sample module.

   c. In the *`addInfoMessage`* method, wrap the message string in the *`snakeCase`* method from Lodash to convert it to snake case.

   ```
   gs.addInfoMessage(snakeCase(`state updated from "${previousState}"
    to "${currentState}"`))
   ```

   d. In line 2, add an import for the *`snakeCase`* method in the *`lodash`* module.

   Example

   ```
   import snakeCase from 'lodash.snakecase'
   ```

   Note: The global Glide APIs are also imported so that you can use methods like *`addInfoMessage`* in your module code.

   ```
   import { gs } from '@servicenow/glide'
   ```

   e. Save your changes.

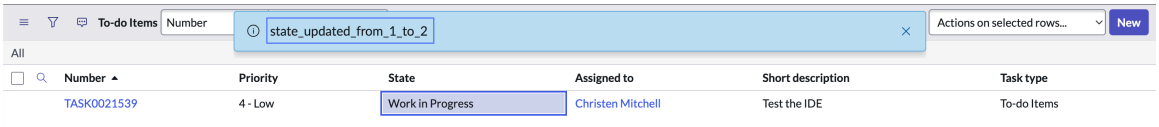
3. From the Status Bar, select **Build and Install**.

   If the installation completes successfully, the Lodash libraries are added to the EcmaScript Module [sys_module] table, and the `script.ts` module is updated in the EcmaScript Module [sys_module] table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Result

After you update any field on a record in the To-do Items [x_snc_hello_world_to_do] table, the sample business rule runs and displays the message in snake case, with the words separated by underscores instead of spaces.



What to do next

Continue to Tutorial part 5: Clone the application on a different instance.

Tutorial part 5: Clone the application on a different instance

Clone the application from the remote repository to develop it on another instance.

Before you begin

Complete Tutorial part 4: Install and use a third-party library.

Role required: admin

About this task

If you have access to another instance, you can install the application from the remote repository onto the instance to support development across instances.

Note: The instance and your user on the instance must meet the same requirements mentioned at the beginning of this tutorial.

Cloning is intended for developing an application on multiple non-production instances and managing it in a single repository. To publish an application and deploy it to a production instance, use the Application Repository. For more information, see ServiceNow application repository.

Procedure

1. Push your changes to the remote repository.

   a. From the Activity Bar, select the Source Control view ( ).

   b. From the Changes list, select the Stage All Tracked Changes icon ( ).

   c. From the Untracked Changes list, select the Stage All Untracked Changes icon ( ).
   All of your changes should be in the Staged Changes list.

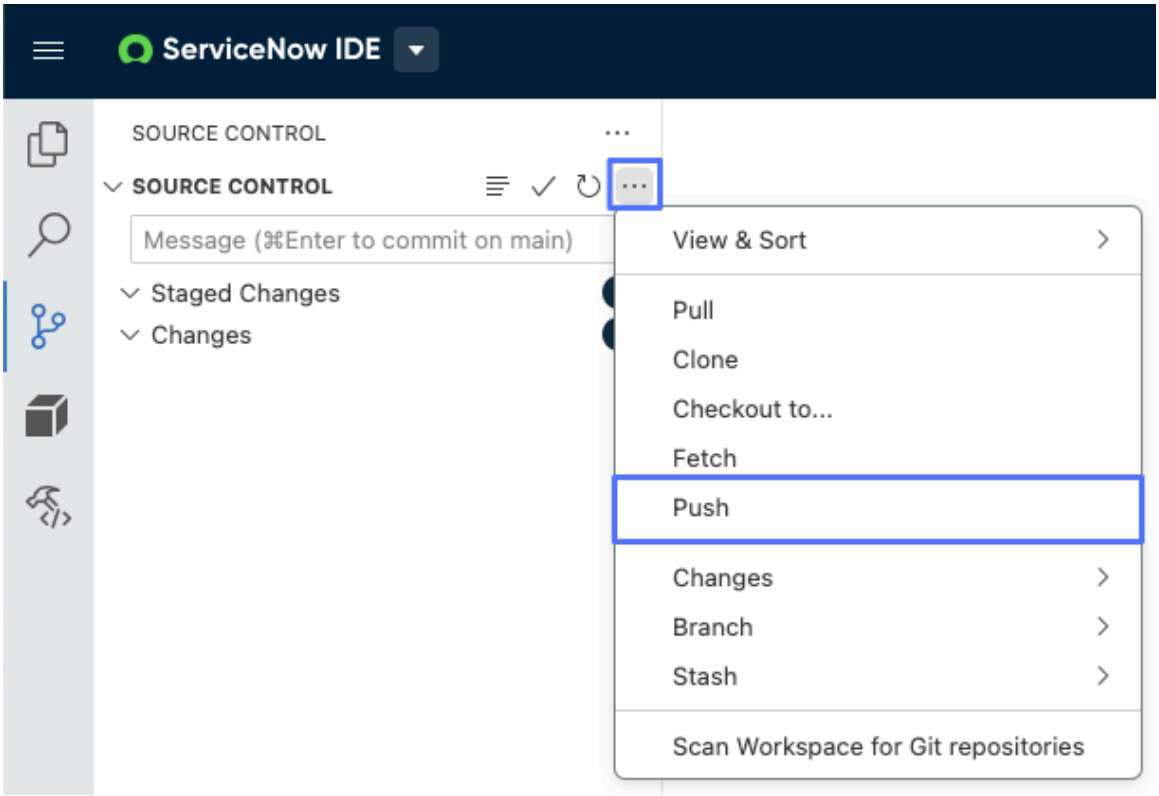
   d. In the message box, enter a commit message.

   e. Select the Commit icon ( ).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

f. Select the More actions menu icon (···) and select **Push**.

Example



All of your changes are available in the remote repository. You or other developers can clone the application onto another instance.

2. Log in to another ServiceNow instance.
3. Navigate to **All** > **App Development** > **ServiceNow IDE**.
4. Create a workspace or open an existing one.
5. Configure basic authentication in the ServiceNow IDE to connect to the remote repository.

   a. Use one of the following keyboard shortcuts to open the command palette:
   - Windows: Ctrl-Shift-P
   - Mac: Cmd-Shift-P

   b. Enter `Git: Set IDE Git credentials` and press Enter.

   c. From the New Git credential form, select **Basic auth**.

   d. On the form, fill in the fields.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

New Git credential form

Field	Description
Git repository URL	The HTTPS URL to the Git repository you created in GitHub. For example: `https://github.com/<owner>/hello-world.git`.
Git username	Your GitHub user name.
Personal access token	The personal access token that you generated from GitHub.

e. Select **Submit**.

6. Clone the application.

a. Use one of the following keyboard shortcuts to open the command palette:

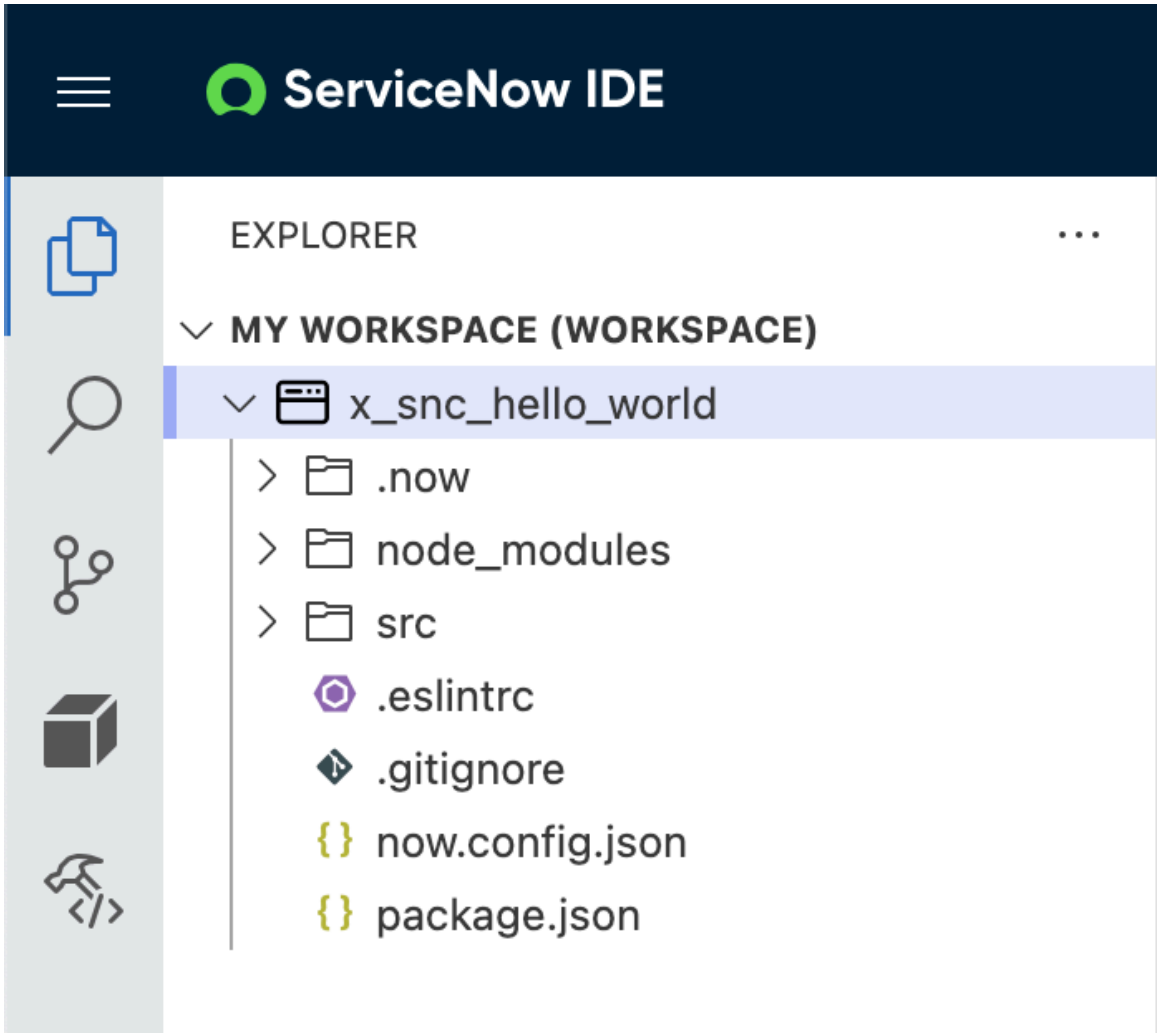
- Windows: Ctrl-Shift-P
- Mac: Cmd-Shift-P

b. Enter `Git: Clone` and press Enter.

c. Enter the HTTPS URL to the remote Git repository and press Enter.

Result
The application is added to the instance with the files from the remote repository.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



What to do next
Continue to Tutorial part 6: Learn more about the ServiceNow IDE.

Tutorial part 6: Learn more about the ServiceNow IDE

Learn more about the ServiceNow IDE to develop more complex applications.

Congratulations! You have created your first application in the ServiceNow IDE, learned how to write ServiceNow Fluent code to define application metadata, installed a third-party library and used its code in a JavaScript module, and managed all of these changes in source control with GitHub.

For more information about development in the ServiceNow IDE, refer to the following resources:

- ServiceNow IDE documentation
- ServiceNow Fluent API reference
- Introduction to the ServiceNow IDE on ServiceNow University
- ServiceNow IDE Demo video
- ServiceNow IDE and Fluent Creator Toolbox video
- ServiceNow SDK examples GitHub repository
- ServiceNow IDE, SDK, and Fluent forum in the ServiceNow Community
- ServiceNow IDE, SDK, and Fluent articles in the ServiceNow Community

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Search the Known Error Portal for known error articles
- Contact Customer Service and Support

Configuring the ServiceNow IDE

Install and set up the ServiceNow IDE for developers to use on a ServiceNow instance.

Install or update the ServiceNow IDE

Install or update the ServiceNow IDE application (sn_glider) from the ServiceNow® Store.

Before you begin

Review the ServiceNow IDE application listing in the ServiceNow Store for information on dependencies, licensing or subscription requirements, and release compatibility.

Role required: admin

About this task

ServiceNow IDE is active by default on instances on the Brazil release. Update to the latest version of the ServiceNow IDE to use the latest features.

Procedure

1. Navigate to **Admin** > **Application Manager**.
2. Search for the ServiceNow IDE application (sn_glider).

   You can search for the application by its name or ID. If you can't find the application, you might have to request it from the ServiceNow Store.
3. Select the ServiceNow IDE application (sn_glider).
4. Select **Install** or **Proceed to Update**.

   In the Review Installation Details dialog box, you can select the version to install, review any dependencies that are installed along with your application, and select whether to install the application now or later.
5. In the dialog box, select **Install**.

What to do next

Create a workspace for your applications. For more information, see Create a workspace in the ServiceNow IDE.

ServiceNow IDE uses the public npm registry (`https://registry.npmjs.org`) as its default package source. If your network blocks access to this registry, you must have access to an alternate registry to download packages and build applications in the ServiceNow IDE. If access to the public npm registry is blocked on your system, you must configure a private npm registry in your Package Manager user settings in the ServiceNow IDE. For more information, see Install an npm package from a private registry with the ServiceNow IDE.

Update the ServiceNow SDK version for an application in the ServiceNow IDE

Configure which version of the ServiceNow SDK an application uses from the ServiceNow IDE.

Before you begin

Role required: admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

About this task

From the ServiceNow IDE, you can configure which version of the ServiceNow SDK, including the ServiceNow Fluent APIs, an application uses beginning with ServiceNow SDK version 3.0.

Procedure

1. Navigate to **All** > **App Development** > **ServiceNow IDE**.
2. Open a workspace with an application.
3. From the Activity Bar, select the File Explorer view ( ).
4. Open the `package.json` file for the application.
5. In the `devDependencies` field, update the versions of the `@servicenow/sdk` and `@servicenow/eslint-plugin-sdk-app-plugin` packages to 3.0.0 or later.

   Example

   For example:

   ```
   "devDependencies": {
               "@servicenow/sdk": "3.0.0",
               "@servicenow/glide": "26.0.1",
               "eslint": "8.50.0",
               "@servicenow/eslint-plugin-sdk-app-plugin": "3.0.0"
   }
   ```

6. Save your changes.
7. Use one of the following keyboard shortcuts to open the command palette:
   - Windows: Ctrl-Shift-P
   - Mac: Cmd-Shift-P
8. Enter `Package Manager: Install Dependencies` and press Enter.
   The dependencies are updated in the `node_modules` directory.

Result

You can use the latest version of the ServiceNow SDK application packaging service and ServiceNow Fluent APIs in your application.

Adding applications in ServiceNow IDE

Add applications in the ServiceNow IDE to get started with development in source code.

1. Create a workspace in the ServiceNow IDE

   Create a workspace in which you can add applications in the ServiceNow IDE.

2. Get started developing an application in source code by creating an application, converting an existing application, or cloning an existing application from a Git repository:

   - Create an application with the ServiceNow IDE

     Create an application that supports development in source code.

   - Convert an application with the ServiceNow IDE

     Convert an existing application to support development in source code.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Clone a Git repository with the ServiceNow IDE

  Clone an existing application that supports development in source code from a remote Git repository.

Create a workspace in the ServiceNow IDE

Create a workspace to view and organize the applications you're working on in the ServiceNow IDE.

Before you begin

Role required: admin

About this task

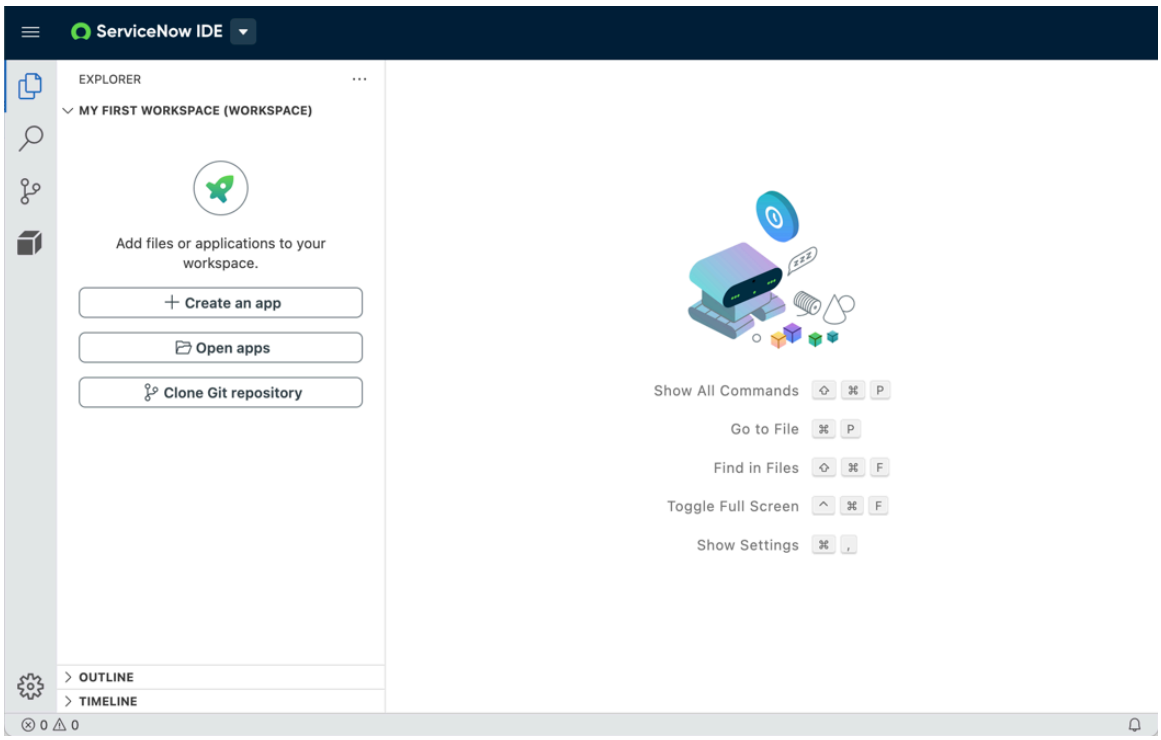
In a workspace in the ServiceNow IDE, you can add any applications that you're working on to navigate through multiple applications from one place. You can create multiple workspaces to group different sets of applications. Workspaces are specific to a user, and applications can be added or removed from a workspace at any time. When you open the ServiceNow IDE, the workspace in which you were most recently working opens. Workspaces in the ServiceNow IDE are based on workspaces in Visual Studio Code. For general information about workspaces, see the Visual Studio Code documentation website.

Procedure

1. Navigate to **All** > **App Development** > **ServiceNow IDE**.
2. From the ServiceNow IDE home page, select **Create a workspace**.

   Tip: You can also use the `Workspaces: Create a Workspace` command from the command palette.

3. Enter a name for the workspace and press Enter.
4. Enter a description for the workspace and press Enter.
   The workspace becomes the active workspace.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

5. **Optional:** Add existing applications to the workspace.

   a. Select **Open apps**.

   b. Select an application.
   Only applications created or converted with the ServiceNow IDE or ServiceNow SDK can be added. You can add additional applications to a workspace from the main menu or the command palette with the `Workspaces: Add Application to Workspace` command.

What to do next

Create, convert, or clone an application and add it to your workspace. For more information, see Adding applications in ServiceNow IDE.

To switch the active workspace, you can browse and select other workspaces from the ServiceNow IDE home page or from the command palette with the `Workspaces: Browse Workspaces` command.

Create an application with the ServiceNow IDE

Create an application to develop in source code with the ServiceNow IDE.

Before you begin

Create a workspace for your applications. For more information, see Create a workspace in the ServiceNow IDE.

Note: You can use Build Agent to help you create and edit applications in the ServiceNow IDE. For more information, see Build Agent.

Role required: admin

Procedure

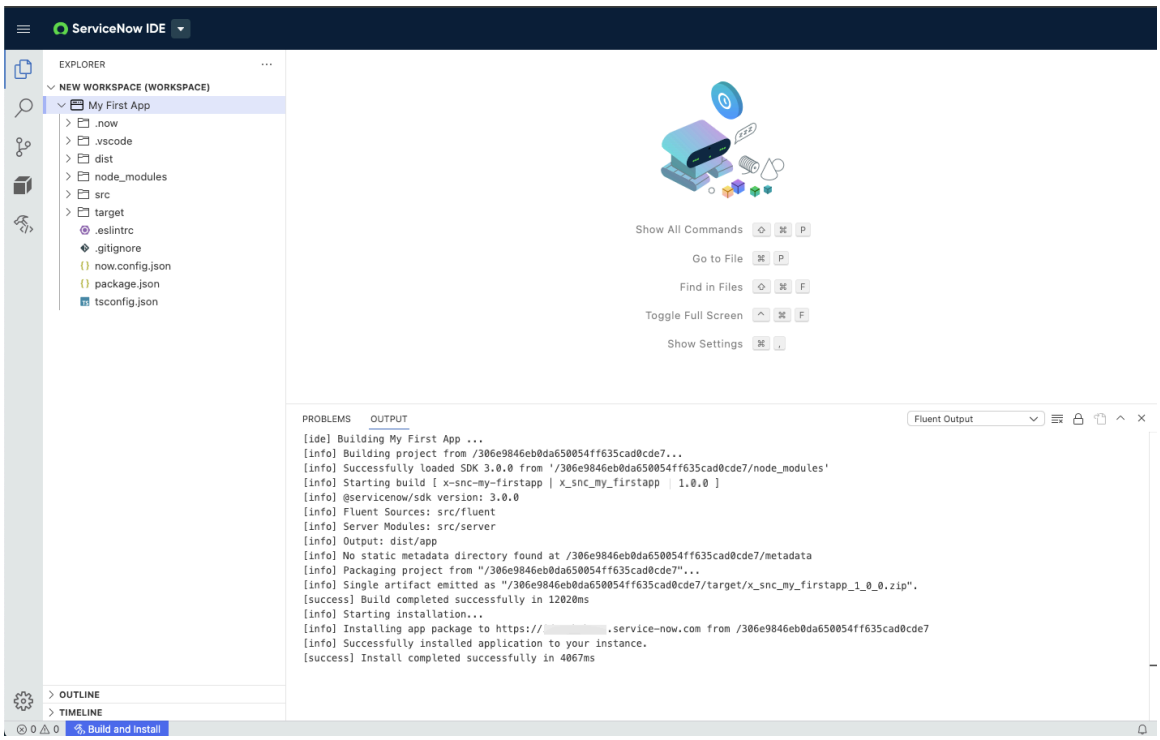
1. Navigate to **All** > **App Development** > **ServiceNow IDE**.
2. Open a workspace.
3. Use one of the following keyboard shortcuts to open the command palette:
   - Windows: Ctrl-Shift-P
   - Mac: Cmd-Shift-P
4. Enter `Fluent: Create Fluent App` and press Enter.
5. Select the type of application.
   - Scoped: Create a scoped application that is protected by identifying and restricting access to application files and data.
   - Global: Create an application in the global scope to allow it to be accessible to other global applications.
6. Enter a name for the application and press Enter.
7. Enter a description for the application and press Enter.
8. For scoped applications, enter a scope name and press Enter.
   The scope name must be unique on the instance, begin with x_<prefix>, and be 18 characters or fewer. For more information, see Namespace identifier.
9. Enter a package name for the application and press Enter.
   The package name must adhere to Node Package Manager (npm) package naming standards.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

10. Select a template that defines the default application structure.
11. **Optional:** Build and install your application to compile source code into application metadata and make your changes available across the instance. For more information, see Build and install an application in the ServiceNow IDE.

Result

An application with the default application structure is added to the instance and open in your workspace. For information about the application structure, see the Application structure section of the Building applications in source code topic.



Trouble?

In the status bar, a message confirms whether the application was created. If the application creation fails, review the output logs in the panel.

Note: For the ServiceNow IDE to install the required dependencies in an application, the public npm registry must respond with the HTTP `Access-Control-Allow-Origin` header.

What to do next

From your Git provider, create a dedicated Git repository for the application. Initialize a local Git repository for your application and push it to the remote repository. For more information, see Initialize a Git repository with the ServiceNow IDE.

In the ServiceNow IDE, start developing your application in source code with ServiceNow Fluent, writing custom JavaScript modules, or adding third-party libraries.

Convert an application with the ServiceNow IDE

Convert an existing application to support development in source code with the ServiceNow IDE.

Before you begin

Create a workspace for your applications. For more information, see Create a workspace in the ServiceNow IDE.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Role required: admin

Note: With ServiceNow IDE version 3.1.4 and earlier, the sn_glider.ide_fluent_admin role is also required to convert applications.

About this task

Existing applications that weren't created with the ServiceNow IDE or ServiceNow SDK must be converted to support development in source code. Converting an application adds the necessary files and directories for developing it in source code. You can choose whether to convert existing application metadata into ServiceNow Fluent code.

Procedure

1. Navigate to **All** > **App Development** > **ServiceNow IDE**.
2. Open a workspace.
3. Use one of the following keyboard shortcuts to open the command palette:
   - Windows: Ctrl-Shift-P
   - Mac: Cmd-Shift-P
4. Enter `Fluent: Convert an app to Fluent` and press Enter.
5. Select an application from the list.

   Applications that haven't been converted are listed as `Is Fluent app: False`.

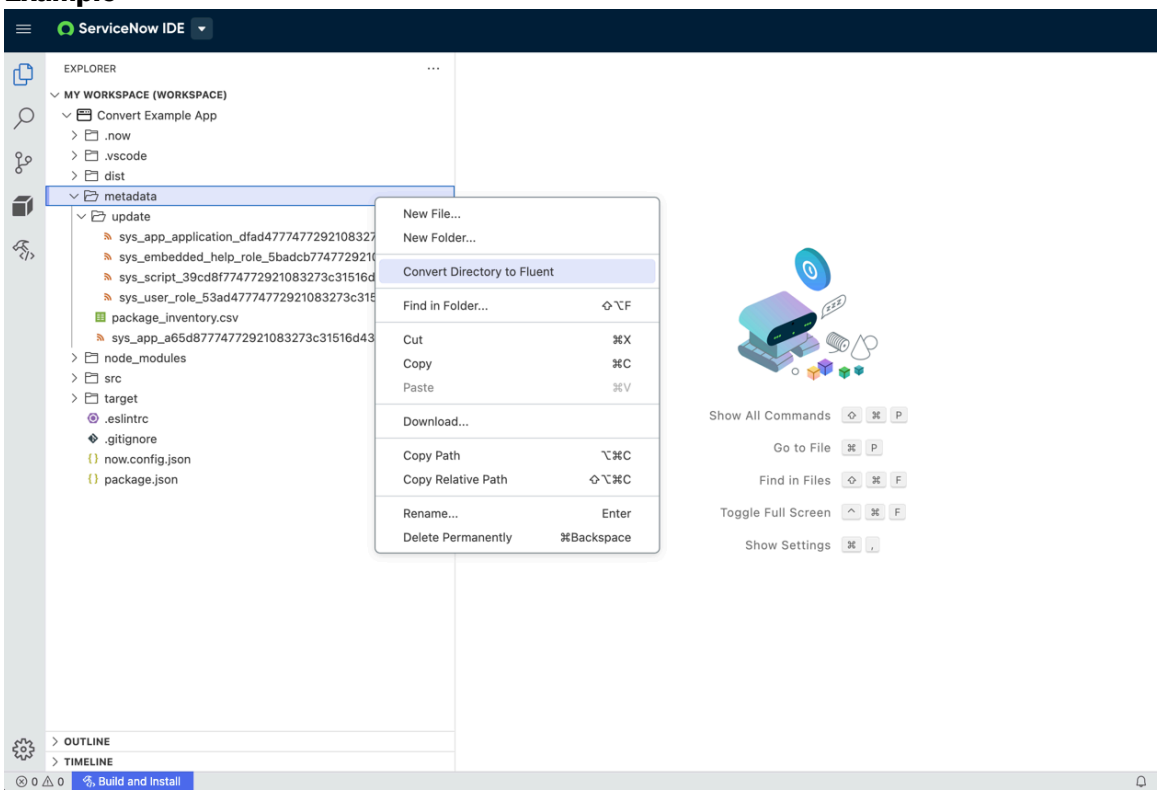
   The application is added to your workspace with the default application structure but the application metadata isn’t converted into ServiceNow Fluent code. For information about the application structure, see the Application structure section of the Building applications in source code topic.
6. **Optional:** To convert existing metadata into ServiceNow Fluent code, complete the following steps.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

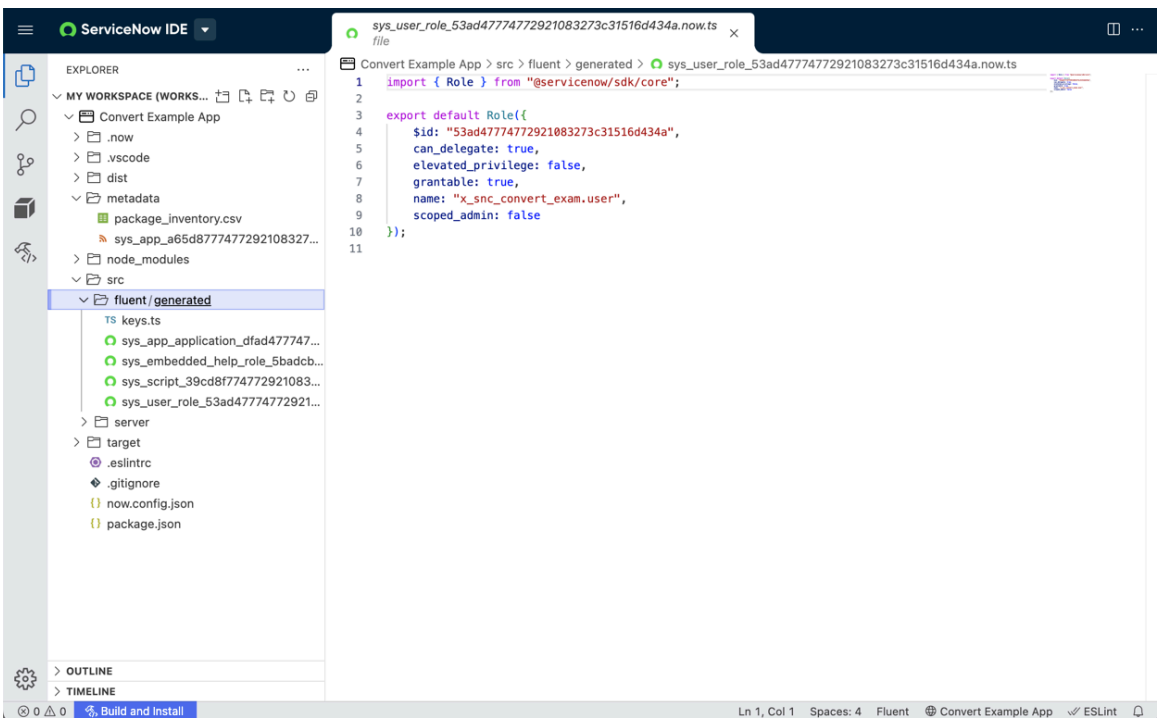
a. From the Activity Bar, select the File Explorer view ( ).

b. Right-click the `metadata` directory for the application and select **Convert Directory to Fluent**.

Example



Application metadata is defined in ServiceNow Fluent code in the `fluent/generated` directory and removed from the `metadata` directory and its sub-directories.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: A limited number of metadata types, such as Metadata Snapshots [sys_metadata_link] and UX Assets [sys_ux_lib_asset], can't be represented as ServiceNow Fluent code and aren't transformed. These metadata types remain as metadata XML files in the `metadata` directory of your application.

7. **Optional:** Build and install your application to compile source code into application metadata and make your changes available across the instance. For more information, see Build and install an application in the ServiceNow IDE.

Result

The converted application is added to your workspace with the necessary files and directories to support development in source code. After installing a converted application, the **Package JSON** field of the custom application record [sys_app] contains the path to the `package.json` file for the application.

Trouble?

Note: For the ServiceNow IDE to install the required dependencies in an application, the public npm registry must respond with the HTTP `Access-Control-Allow-Origin` header.

What to do next

From your Git provider, create a dedicated Git repository for the application. Initialize a local Git repository for your application and push it to the remote repository. For more information, see Initialize a Git repository with the ServiceNow IDE.

In the ServiceNow IDE, start developing your application in source code with ServiceNow Fluent, writing custom JavaScript modules, or adding third-party libraries.

Clone a Git repository with the ServiceNow IDE

Clone a remote Git repository to collaborate on applications in source control with the ServiceNow IDE.

Before you begin

- Create a workspace for your applications. For more information, see Create a workspace in the ServiceNow IDE.
- Set your basic or OAuth 2.0 credentials for the ServiceNow IDE to connect to your Git repository. For more information, see Connect to a Git provider using basic authentication with the ServiceNow IDE or Configure OAuth 2.0 credentials to connect to a Git provider with the ServiceNow IDE.

Role required: admin

About this task

You can clone a remote Git repository that contains applications created or converted with the ServiceNow IDE or ServiceNow SDK. The repository must contain one or more applications with `now.config.json` and `package.json` files.

Note: ServiceNow IDE supports cloning from Git servers on Git version 2.3.2 or later.

Cloning is intended for developing an application on multiple non-production instances and managing it in a single repository. To publish an application and deploy it to a production instance, use the Application Repository. For more information, see ServiceNow application repository.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **App Development** > **ServiceNow IDE**.
2. Open a workspace to which you want to add the application.
3. Use one of the following keyboard shortcuts to open the command palette:
   - Windows: Ctrl-Shift-P
   - Mac: Cmd-Shift-P
4. Enter `Git: Clone` and press Enter.
5. Enter a remote Git repository URL and press Enter.

Result

The application is added to the instance and your workspace with the files from the remote repository. Only the default branch is cloned initially. To check out another branch, you must fetch the other branches from the remote repository using the `Git: Fetch` command from the command palette.

What to do next

You can check out or create branches in the repository and push changes to the remote repository. For more information, see Using source control in the ServiceNow IDE.

Integrating source control with the ServiceNow IDE

Integrate with remote Git repositories to manage applications in source control with the ServiceNow IDE.

You can connect to a Git provider using basic or OAuth 2.0 authentication. Then, initialize a repository for an application in the ServiceNow IDE and push it to a remote repository or clone a remote repository that contains an application. After setting up authentication and connecting to a repository, you can use common Git commands to manage applications in source control.

Learn how to get started using source control in the ServiceNow IDE in the following topics.

Instance requirements

If an instance includes modified configurations for saving attachments, the following requirements must be met to perform Git operations:

- If the *glide.attachment.extensions* system property is configured, it must contain `txt,gitdata`.
- If the *glide.security.attachment_type.use_blacklist* system property is set to true, the *glide.attachment.blacklisted.extensions* system property must not contain `txt,gitdata` and the *glide.attachment.blacklisted.types* system property must not contain `text/plain,application/octet-stream`.

If you want to use custom extensions for attachments, set the following properties to custom values: *sn_glider.git.attachment.extension.text* and *sn_glider.git.attachment.extension.binary*. For more information, see ServiceNow IDE properties.

Connect to a Git provider using basic authentication with the ServiceNow IDE

Connect to a Git domain using basic authentication credentials to manage applications in source control from the ServiceNow IDE.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

- From your Git provider, generate a personal access token to use for basic authentication to the Git domain . You must provide your Git user name and personal access token when configuring your credentials for the ServiceNow IDE.
- Create a dedicated Git repository for an application in a Git provider such as GitHub, GitLab, Bitbucket, or Azure Repos.

Role required: admin

Procedure

1. Navigate to **All** > **App Development** > **ServiceNow IDE**.
2. Use one of the following keyboard shortcuts to open the command palette:
   - Windows: Ctrl-Shift-P
   - Mac: Cmd-Shift-P
3. Enter `Git: Set IDE Git credentials` and press Enter.
4. From the New Git credential form, select **Basic auth**.
5. On the form, fill in the fields.

New Git credential form

Field	Description
Git repository URL	The HTTPS URL of a Git repository associated with your Git credentials.
Git username	Your Git user name.
Personal access token	A personal access token that you generate from your Git provider.

6. Select **Submit**.

Result

Your Git credentials are associated with your user on the instance and used for all repositories in the domain from the Git repository URL. If you add different credentials for a repository in the same domain, the new credentials are used and the previous credentials are set to inactive.

What to do next

After initializing or cloning a repository, you can begin using source control. For more information, see Using source control in the ServiceNow IDE.

To manage existing Git credentials, use the `Git: Manage Git credentials` command from the command palette.

Connect to a Git provider using OAuth 2.0 with the ServiceNow IDE

Set up an OAuth 2.0 application registry and credentials to connect to your Git provider from the ServiceNow IDE.

Configure an OAuth 2.0 application registry for the ServiceNow IDE

Configure how the client ID and secret are sent to the OAuth 2.0 provider associated with your Git provider.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

Create an OAuth application with your Git provider and configure it to redirect to your instance. In this OAuth application, use your instance URL as the homepage URL and `https://<instance>/oauth_redirect.do` for the authorization callback URL. GitHub, GitLab, Bitbucket, and Azure Repos are supported by default.

Important: For Azure Repos, the maximum length of the **Client Secret** field must be updated to 2048 before you add the secret.

1. Navigate to **All** > **System Definition** > **Tables**.
2. Filter the table by entering `oauth_entity` for the **Name** field.
3. Select the Application Registries [oauth_entity] table.
4. In the Columns related list, locate the Client Secret column and enter `2048` as the value of its **Max length** attribute.
5. Select **Update**.

Role required: admin

About this task

To use OAuth 2.0 authentication with the ServiceNow IDE, you must register the OAuth application you created from your Git provider. Follow this procedure to configure an OAuth application registry [oauth_entity] on your instance.

Procedure

1. Navigate to **All** > **System OAuth** > **Application Registry** and then select **New**.
2. On the interceptor page, select **Connect to a third-party OAuth provider**.
3. On the form, fill in the fields.

   For additional information about fields on the form, see Connect to a third-party OAuth provider.

Application Registries form

Field	Description
Name	A unique name for the third-party OAuth connection.
Client ID	The client ID of the OAuth application in your Git provider.
Client Secret	The client secret of the OAuth application in your Git provider.
OAuth API Script	The script used to customize request and response to the external OAuth provider. Select one of the following depending on your Git provider: ○ GitHub: OauthAPIScriptForGitHub ○ GitLab: OauthAPIScriptForGitLab ○ Bitbucket: OauthAPIScriptForBitbucket ○ Azure Repos: OauthAPIScriptForAzureRepos

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
	Note: To use other Git providers, you can use these scripts as examples to create your own OAuth API script for your OAuth provider. The name of the script must begin with "Oauth".
Default Grant type	The default grant type used to establish the token. Select **Authorization code**. An authorization code is granted to the client to obtain an access token, which is then used to obtain access to the resource.
Refresh Token Lifespan	The time, in seconds, that the refresh token is valid.
Application	The application scope that contains this record.
Accessible from	An option to make the application registry accessible from all application scopes or from this scope only.
Active	An option for turning the OAuth application on or off.
Authorization URL	The OAuth authorization code endpoint for your Git domain. For example: ○ GitHub: `https://github.com/login/oauth/authorize` ○ GitLab: `https://gitlab.com/oauth/authorize` ○ Bitbucket: `https://bitbucket.org/site/oauth2/authorize` ○ Azure Repos: `https://app.vssps.visualstudio.com/oauth2/authorize`
Token URL	The OAuth server token endpoint for your Git domain. For example: ○ GitHub: `https://github.com/login/oauth/access_token` ○ GitLab: `https://gitlab.com/oauth/token` ○ Bitbucket: `https://bitbucket.org/site/oauth2/access_token` ○ Azure Repos: `https://app.vssps.visualstudio.com/oauth2/token`
Redirect URL	The OAuth callback endpoint. If empty, the instance auto-generates a value of `https://<instance>/oauth_redirect.do`.
Use mutual authentication	An option to use mutual authentication. Leave this option cleared to turn off using mutual authentication for token request and revocation.
Send Credentials	The way in which the OAuth client populates the client credentials in the request.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
	○ For GitHub, GitLab, or Bitbucket, select **As Basic Authorization header**. ○ For Azure Repos, select **In Request Body (Form URL-Encoded)**.

4. Select **Submit**.

What to do next

Developers using the ServiceNow IDE must configure their own OAuth 2.0 credentials.

Configure OAuth 2.0 credentials to connect to a Git provider with the ServiceNow IDE

Connect to a Git domain using OAuth 2.0 credentials to manage applications in source control from the ServiceNow IDE.

Before you begin

- An administrator must configure how the client ID and secret are sent to the OAuth 2.0 provider associated with your Git provider. For more information, see Configure an OAuth 2.0 application registry for the ServiceNow IDE.
- Create a dedicated Git repository for an application in a Git provider such as GitHub, GitLab, Bitbucket, or Azure Repos.

Role required: admin

Procedure

1. Navigate to **All** > **App Development** > **ServiceNow IDE**.
2. Use one of the following keyboard shortcuts to open the command palette:
   - Windows: Ctrl-Shift-P
   - Mac: Cmd-Shift-P
3. Enter `Git: Set IDE Git credentials` and press Enter.
4. From the New Git credential form, select **OAuth**.
5. On the form, fill in the fields.

New Git credential form

Field	Description
Git repository URL	The HTTPS URL of a Git repository associated with your Git credentials.
Git username	Your Git user name.
Select an OAuth profile	The OAuth 2.0 credentials for your Git provider. The OAuth profile is created with the application registry, which must be configured by an administrator.

6. Select **Submit**.

   The first time you use a Git command, you're prompted to authorize your user from the Git provider.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Result
Your Git credentials are associated with your user on the instance and used for all repositories in the domain from the Git repository URL. If you add different credentials for a repository in the same domain, the new credentials are used and the previous credentials are set to inactive.

What to do next
After initializing or cloning a repository, you can begin using source control. For more information, see Using source control in the ServiceNow IDE.

To manage existing Git credentials, use the `Git: Manage Git credentials` command from the command palette.

Configure a MID Server to use source control with the ServiceNow IDE

Configure a MID Server to use source control with the ServiceNow IDE if your Git provider is behind a firewall.

Before you begin
Install a MID Server with a REST capability. For more information, see Installing the MID Server and Configure MID Server capabilities.

Role required: admin

Procedure
1. Change the application scope of the instance to ServiceNow IDE.
   a. In the Unified Navigation, select the globe icon ().
   b. Select **Application scope**.
   c. Select **ServiceNow IDE**.
2. Navigate to **All** > **MID Server** > **Applications**.
3. Select **New**.
4. On the form, fill in the fields.

MID Server Application form

Field	Description
Name	A name for the MID Server Application.
Default MID Server	The MID Server configured with a REST capability.
Application	The ServiceNow IDE.
Included in application ALL	An option to include this MID Server Application in the definition of ALL for a MID Server.

5. Select **Submit**.

What to do next
If you haven't already, configure basic or OAuth 2.0 authentication to connect to a Git domain or repository. For more information, see Connect to a Git provider using basic authentication with the ServiceNow IDE or Connect to a Git provider using OAuth 2.0 with the ServiceNow IDE.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: The MID Server user must have the sn_glider.ide_git_user role or admin role to perform Git operations in the ServiceNow IDE. For more information, see Create the MID Server user and grant the role and ServiceNow IDE MID Server User [sn_glider.ide_git_user].

Initialize a Git repository with the ServiceNow IDE

Initialize a local Git repository for an application and push it to a remote Git repository to manage an application in source control.

Before you begin

- Create or convert an application with the ServiceNow IDE. For more information, see Create an application with the ServiceNow IDE or Convert an application with the ServiceNow IDE.
- Create a dedicated Git repository for the application from your Git provider.
- Set your basic or OAuth 2.0 credentials for the ServiceNow IDE to connect to your Git repository. For more information, see Connect to a Git provider using basic authentication with the ServiceNow IDE or Configure OAuth 2.0 credentials to connect to a Git provider with the ServiceNow IDE.

Role required: admin

About this task

An application on an instance can be connected to only one repository at a time. To clone an application that exists in a remote Git repository, see Clone a Git repository with the ServiceNow IDE.

Procedure

1. Navigate to **All** > **App Development** > **ServiceNow IDE**.
2. Open a workspace with an application that isn't connected to a Git repository.
3. From the Activity Bar, select the Source Control view ( ).
4. Select **Initialize Repository** or use the `Git: Initialize Repository` command from the command palette.
5. Select the application for which you want to initialize a Git repository and press Enter.
6. Select **main** as the default branch name or enter another name and press Enter.
7. Select the Stage All Untracked Changes icon ( ).
8. Enter a commit message and select the Commit icon ( ).
9. Select the More actions menu icon ( ) and select **Push**.
10. Enter a remote repository URL and press Enter.

Result

The application is available in the remote repository.

Trouble?

If your Git credentials aren't configured or are inactive, the application isn't pushed to the remote repository. When prompted to configure your credentials, select **Configure** to configure your Git credentials and then try again.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

What to do next

You can check out or create branches in the repository and push changes to the remote repository. For more information, see Using source control in the ServiceNow IDE.

Using source control in the ServiceNow IDE

Use Git commands and other source control features in the ServiceNow IDE to manage changes to an application across a development team.

Role required: admin

Git commands

After initializing a local Git repository in the ServiceNow IDE, you can perform several Git commands from the Source Control view ( ) or command palette, including but not limited to the following commands.

Git commands in the ServiceNow IDE

Command	Description
Checkout to (> **Checkout to...**) Command palette: **Git: Checkout to...**	Check out another branch from the repository. Select the branch from the list or create a branch and push it to the remote repository. Only one branch per repository can be checked out at a time on an instance (or developer sandbox).
Clone (> **Clone**) Command palette: **Git: Clone**	Clone a remote repository to add an application to your workspace. For more information, see Clone a Git repository with the ServiceNow IDE.
Commit ( ) Command palette: **Git: Commit**	Commit your staged changes to the local repository. Enter a commit message to describe your changes.
Create branch (> **Branch** > **Create branch...**) Command palette: **Git: Create branch...**	Add a branch to a repository and check it out.
Discard ( ) Command palette: **Git: Discard Changes**	Discard changes to undo modifications to an application.
Fetch (> **Fetch**) Command palette: **Git: Fetch**	Fetch to sync commits and branches from the remote repository into the local repository.
Pull (> **Pull**) Command palette: **Git: Pull**	Pull to merge the latest changes from the remote repository into the local repository.
Push (> **Push**)	Push your committed changes to the remote repository.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Git commands in the ServiceNow IDE (continued)

Command	Description
Command palette: **Git: Push**	
Stage (+) Command palette: **Git: Stage Changes**	Stage the changes in your working directory that you want to commit. When you stage changes, files move from the Changes list to the Staged Changes list.
Stash (··· > **Stash**) Command palette: **Git: Stash**, **Git: Pop**, **Git: Drop**, **Git: List**, **Git: Apply**, **Git: Clear**	Stash all uncommitted changes to save them in your working directory locally and come back to them later. The pop, drop, list, apply, and clear subcommands are supported when stashing. Select **Stash** > **Apply** to reapply your changes in your working directory.

Merge conflicts

If there's a conflict between local and remote changes in a file, the file is listed under Merge conflicts in the Source Control view ( ). You can review conflicts in the editor and resolve them by accepting the current or incoming change, both, or manually editing the file.

Commit history

The commit history for a branch is listed in the Commits section of the Source Control view ( ) and includes details about the commits. To see the commit history for a file, navigate to the File Explorer view ( ), select the file, and expand the Timeline section. When you select a commit, a Diff editor comparing the changes opens.

Repository changes

To update the remote repository an application is connected to, you can use the `Git: Update remote origin` command from the command palette and enter a different remote repository URL.

Developing applications with the ServiceNow IDE

Create and develop applications in source code with the ServiceNow IDE. Build applications to make your changes available across an instance and collaborate with users of any skill set.

In the ServiceNow IDE, you create applications or convert existing applications and develop them in source code using ServiceNow Fluent to define application metadata [sys_metadata]. Optionally, you can create JavaScript modules and use third-party libraries to organize and reuse code within application scopes.

When your changes are saved and ready, build an application to compile the source code to application metadata and make your changes available to users across the instance. As other users modify the application metadata, you can synchronize the changes into the source code.

Learn how to get started developing applications with the ServiceNow IDE in the following topics.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

System requirements

ServiceNow IDE uses the public npm registry (`https://registry.npmjs.org`) as its default package source. If your network blocks access to this registry, you must have access to an alternate registry to download packages and build applications in the ServiceNow IDE. If access to the public npm registry is blocked on your system, you must configure a private npm registry in your Package Manager user settings in the ServiceNow IDE. For more information, see Install an npm package from a private registry with the ServiceNow IDE.

Related topics

- Integrating source control with the ServiceNow IDE
- ServiceNow Fluent API reference

Synchronizing applications in the ServiceNow IDE

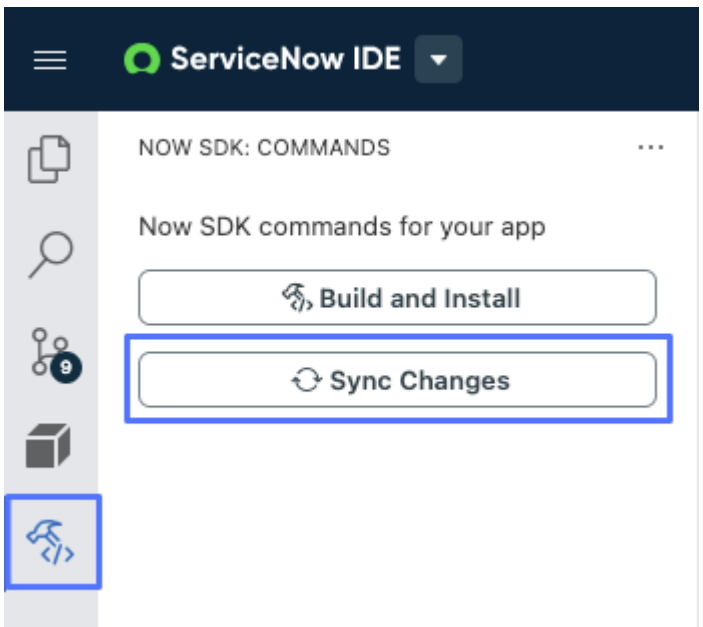
Synchronizing an application in the ServiceNow IDE downloads and transforms application metadata into ServiceNow Fluent code.

ServiceNow Fluent supports two-way synchronization, which allows changes to metadata to be synced from other ServiceNow AI Platform user interfaces into source code. When you build and install an application, changes to source code are compiled into metadata across the instance. This process enables collaborating with users of different skill sets because developers can modify the ServiceNow Fluent code while others modify the application metadata across the platform.

In the ServiceNow IDE, when metadata in an application is changed, you're prompted to sync the application to get the latest changes. If metadata changes are detected in application, syncing is required before you can install the application across the instance.

Note: If you ignore the notification to sync an application, you aren't prompted to do so again for the remainder of the session. If you try to install the application or refresh the browser, you are prompted to sync again.

You can sync an application on demand from the Now SDK view ( ) or from the command palette with the `Fluent: Sync Fluent App with changed metadata` command.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Synchronization process in the ServiceNow IDE

The following list explains the stages of the synchronization process for applications in the ServiceNow IDE:

1. Before the first time an application is synced:
   - You should initialize a repository.
   - You should stage any changes to the application to retain them when the application is synced.
2. The first time that an application is synced:
   - New metadata that isn't defined in source code is transformed into source code in the `src/fluent/generated` directory.
   - Changes to metadata that is defined in source code are downloaded into source code in the `src/fluent` and `src/fluent/generated` directories.
   - Changes to existing metadata XML that hasn't been converted into source code are downloaded into the `metadata` directory.
3. Subsequent times that an application is synced:
   - Only the changes made since the last time the application was synced are downloaded and transformed.
   - Optionally, you can synchronize all metadata in the application with the `Fluent: Force Sync of Fluent App with all metadata` command from the command palette. Before running this command, you should stage any changes to retain them.

Related topics

ServiceNow Fluent

Define application metadata in code with ServiceNow Fluent in the ServiceNow IDE

Define application metadata in code with ServiceNow Fluent in the ServiceNow IDE.

Before you begin

Create, convert, or clone an application and add it to your workspace. For more information, see Adding applications in ServiceNow IDE.

Role required: admin

About this task

ServiceNow Fluent is a domain-specific language (DSL) based on TypeScript for defining the metadata files [sys_metadata] that make up applications and includes APIs for the different types of metadata, such as tables, roles, ACLs, business rules, and Automated Test Framework tests. For more information about ServiceNow Fluent APIs and examples, see ServiceNow Fluent API reference. The ServiceNow IDE has language processing and validation for ServiceNow Fluent APIs and applications by default.

Procedure

1. Navigate to **All** > **App Development** > **ServiceNow IDE**.
2. Open a workspace with an application.
3. From the Activity Bar, select the File Explorer view ( ).
4. In the `src/fluent` directory, add a file with the `.now.ts` extension.
   You can define application metadata in a single file or as many `.now.ts` files as you want and organize files in directories within the `fluent` directory.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Tip: In the `fluent` directory, you can refer to an example file named `index.now.ts`.

5. In the `.now.ts` file, use objects in the ServiceNow Fluent APIs to define metadata in the application.

   Example

   For example, to create a table [sys_db_object] in the application, use the Table API. The following example includes a simple table for to-do lists using a *`Table`* object with the necessary properties and values.

```
export const x_snc_example_to_do = Table({
    name: 'x_snc_example_to_do',
    schema: {
        deadline: DateColumn({ label: 'Deadline' }),
        state: StringColumn({
            label: 'State',
            choices: {
                ready: { label: 'Ready' },
                completed: { label: 'Completed' },
                inProgress: { label: 'In Progress' },
            },
        }),
        task: StringColumn({ label: 'Task', maxLength: 120 }),
    },
})
```

   Note: You can use the Record API to define application metadata that doesn't have a dedicated API.

6. Add the required imports for the APIs from `@servicenow/sdk/core`.

   Example

   For example, to define a table with a date column and string column, import the corresponding objects to use from the Table API:

```
import { Table, DateColumn, StringColumn }
 from '@servicenow/sdk/core'
```

7. **Optional:** For ServiceNow Fluent APIs that use server-side scripts, such as business rules, import code from JavaScript modules to call them from an object's *`script`* property.

   Example
   This example imports the `showStateUpdate` function, which can then be referenced from the *`script`* property.

```
import { showStateUpdate } from '../server/script.js'
```

8. Save your changes.
9. **Optional:** Build and install your application to compile source code into application metadata and make your changes available across the instance. For more information, see Build and install an application in the ServiceNow IDE.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Trouble?

Tip: You can use the following directives in a code comment to help manage your code:

- `@fluent-ignore`: Suppresses ServiceNow Fluent diagnostic warnings and errors in the following line of code.
- `@fluent-disable-sync`: Turns off syncing changes to a ServiceNow Fluent object. Use before a call expression (for example, `Record({ ... })`) to turn off syncing for that object and its child objects. Only use this directive if you want to ignore changes made outside of the source code to the object and never update it when syncing.
- `@fluent-disable-sync-for-file`: Turns off syncing changes to a ServiceNow Fluent file (`.now.ts`). Use in the first line of the file to turn off syncing for all code in the file. Only use this directive if you want to ignore changes made outside of the source code to the file and never update it when syncing.

Example: Defining application metadata in source code with ServiceNow Fluent

In files with the `.now.ts` extension, use objects in the ServiceNow Fluent APIs to define metadata in the application. You must also include the required imports for the APIs from `@servicenow/sdk/core`. For objects with server-side scripts, such as the *`BusinessRule`* object, you can import and use code from JavaScript modules.

The following example includes the definitions of a table, client script, and business rule in the application. The client script uses a script from the `client-script.js` file. The business rule uses a function from the `script.js` JavaScript module.

```
import '@servicenow/sdk/global'
import { BusinessRule, ClientScript, DateColumn, StringColumn,
 Table } from '@servicenow/sdk/core'
import { showStateUpdate } from '../server/script.js'

//creates todo table, with three columns (deadline, status and task)
export const x_snc_example_to_do = Table({
    name: 'x_snc_example_to_do',
    schema: {
        deadline: DateColumn({ label: 'Deadline' }),
        state: StringColumn({
            label: 'State',
            choices: {
                ready: { label: 'Ready' },
                completed: { label: 'Completed' },
                inProgress: { label: 'In Progress' },
            },
        }),
        task: StringColumn({ label: 'Task', maxLength: 120 }),
    },
})

//creates a client script that pops up 'Table loaded successfully!!' message everytime todo
 record is loaded
ClientScript({
    $id: Now.ID['cs0'],
    name: 'my_client_script',
    table: 'x_snc_example_to_do',
    active: true,
    appliesExtended: false,
```

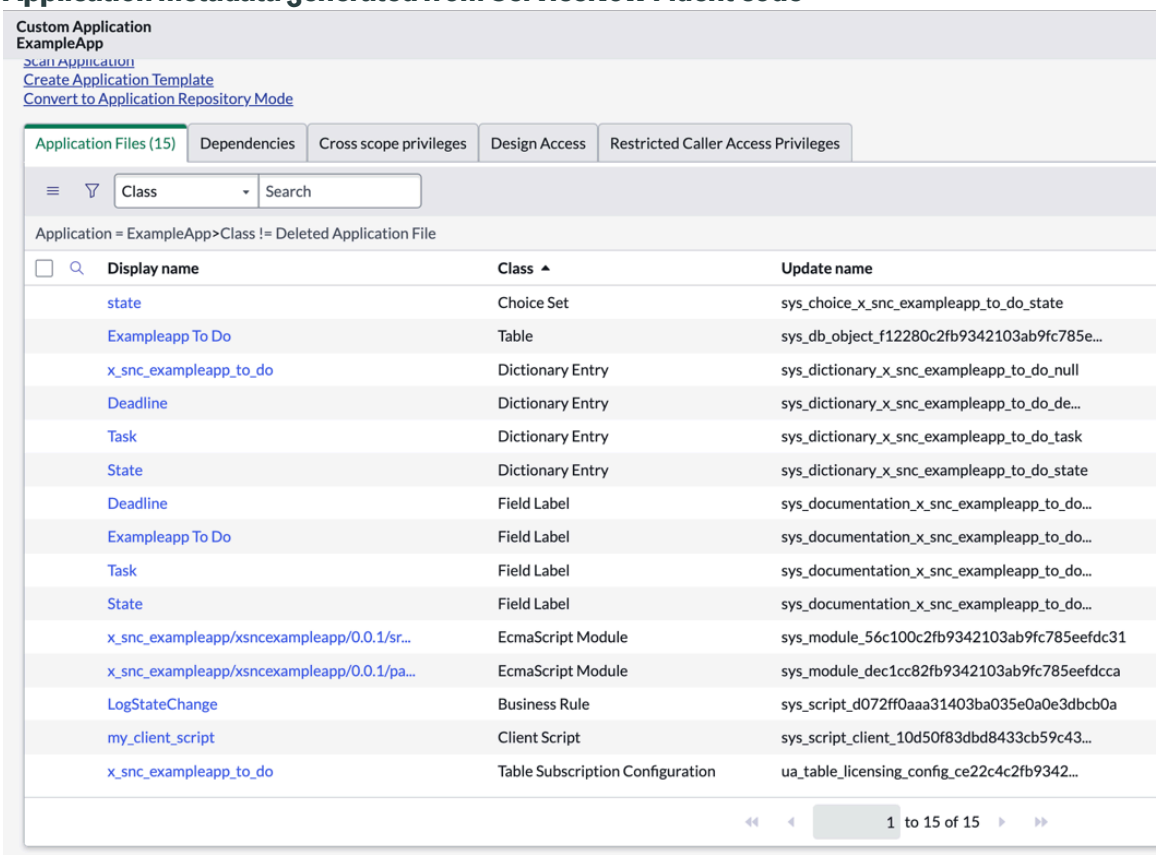
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    global: true,
    uiType: 'all',
    description: 'Custom client script generated by Now SDK',
    isolateScript: false,
    type: 'onLoad',
    script: Now.include('../client/client-script.js'),
})

//creates a business rule that pops up state change message whenever a todo record is
 updated
BusinessRule({
    $id: Now.ID['br0'],
    action: ['update'],
    table: 'x_snc_example_to_do',
    script: showStateUpdate,
    name: 'LogStateChange',
    order: 100,
    when: 'after',
    active: true,
})
```

After building the application, this source code generates the following application metadata files on the instance.

Application metadata generated from ServiceNow Fluent code



Related topics

ServiceNow Fluent

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Create and use JavaScript modules in applications in the ServiceNow IDE

Optimize your codebase by defining reusable code blocks with JavaScript modules in the ServiceNow IDE.

Before you begin

Create, convert, or clone an application and add it to your workspace. For more information, see Adding applications in ServiceNow IDE.

Note: To use TypeScript in modules, select TypeScript as the template type when creating an application.

Role required: admin

About this task

To learn about support for using JavaScript modules in scoped applications, including some limitations, see JavaScript modules and third-party libraries. For general information about the syntax used to create JavaScript modules, see the JavaScript modules page on the MDN Web Docs website.

Note: With ServiceNow Otto for Code, you can use the Code autocomplete skill to generate code suggestions for module scripts. For more information, see Generate code with autocomplete.

Procedure

1. Navigate to **All** > **App Development** > **ServiceNow IDE**.
2. Open a workspace with an application.
3. From the Activity Bar, select the File Explorer view ( ).
4. In the `src/server` directory of the application, create a JavaScript or TypeScript file to contain the module code you want to reuse.
5. **Optional:** Import server APIs or script includes to call them from your module.

   (Optional) Glide APIs can be imported from the `@servicenow/glide` package or their namespace in the package. Script includes can be imported from their application scope or the global scope in the `@servicenow/glide` package.

   For example:

   ```
   import { API } from "@servicenow/glide";
   import { API } from "@servicenow/glide/<namespace>";
   import { ScriptInclude } from "@servicenow/glide/<scope>";
   import { global } from "@servicenow/glide/global";
   ```

6. In the module, identify the code to export with *`export`* statements.

   You can use a named export or default export. Named exports can be variables, constants, functions, or classes whereas default exports can be functions or classes only.

   Example

   The following example is one way of adding a named export for multiple features (a function and a variable) in a module:

   ```
   export { myFunction, myVariable };
   ```

7. Use code from the exported module in other modules or server-side scripts.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

File	Steps
Module	**a.** From the `src/server` directory, create or open a JavaScript module. **b.** Import the module code with *`import`* statements. The following example is one way that you could import an exported feature in a module: `import { feature } from "path/to/module";` **Note:** To import code from one TypeScript file to another TypeScript file, you must include the `.ts` file extension. For example, `import { feature } from './module.ts'`. **c.** Call the module code from this module to reuse it.
Server-side script in source code	**a.** Create or open the definition of application metadata that includes a server-side script, such as a business rule, in source code (`.now.ts` file). **b.** In the *`script`* property, import and call the module code to reuse it. You can import a function or provide an inline script. ▪ Import an exported function, function expression, or default function. For example: `script: FunctionExport,` ▪ Inline scripts must use *`require`* statements to import the module code. For example: `script: ` const { process } = require('./dist/modules/server/handler.js') process(request, response)`,` For more information about server-side scripts in source code, see ServiceNow Fluent API reference.
Server-side script record	**a.** Open the record for a server-side script, such as a business rule. **b.** Import the module code with *`require`* statements. The following example is one way that you could import an exported feature in a script: `const { feature } = require("path/to/module");` **c.** Call the module code from this script to reuse it.

8. Save your changes.
9. From the Status Bar, select **Build and Install**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Build and Install

The active file that's open in the editor determines which application to build. If no files are open, select the application to build when prompted.

After building, the modules are added to the EcmaScript Module [sys_module] table.

What to do next

To use third-party libraries in a JavaScript module, see Use third-party libraries in applications in the ServiceNow IDE.

Related topics

JavaScript modules and third-party libraries

Use third-party libraries in applications in the ServiceNow IDE

Call third-party libraries in your application to use existing open-source functionality with the ServiceNow IDE.

Before you begin

Create a JavaScript module. For more information, see Create and use JavaScript modules in applications in the ServiceNow IDE.

To install and use packages from private registries, you must configure your Package Manager user settings in the ServiceNow IDE. For more information, see Install an npm package from a private registry with the ServiceNow IDE.

Role required: admin

About this task

Third-party libraries are added to applications as JavaScript modules. For general information about the syntax used to create JavaScript modules, see the JavaScript modules page on the MDN Web Docs website.

Important: You can't use third-party libraries that rely on unsupported functionality, such as unsupported APIs or ECMAScript features. For more information about unsupported functionality, see Third-party library support in Brazil and JavaScript modules and third-party libraries.

Procedure

1. Navigate to **All** > **App Development** > **ServiceNow IDE**.
2. Open a workspace with an application.
3. From the Activity Bar, select the File Explorer view ( ).
4. Open the `package.json` file for the application.
5. Add the `dependencies` field with the package name and version of any third-party libraries to use.

   Example

   ```
   "dependencies": {
         "<package name>": "<version>"
     }
   ```

6. Install the third-party library packages.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. Use one of the following keyboard shortcuts to open the command palette:
   - Windows: Ctrl-Shift-P
   - Mac: Cmd-Shift-P

b. Enter `Package Manager: Install Dependencies` and press Enter.

Packages are installed in the `node_modules` directory.

7. In a JavaScript module, import the library using an *`import`* statement.

Example

In this example, the module includes a namespace import for the lodash module.

```
import * as lodash from "lodash"
```

In this example, the module includes a named import for the camelCase function in the lodash module.

```
import camelCase from 'lodash'
```

8. Call code imported from the library in your module to reuse it.
9. From the Status Bar, select **Build and Install**.



The active file that's open in the editor determines which application to build. If no files are open, select the application to build when prompted.

After building, the modules are added to the EcmaScript Module [sys_module] table.

Related topics

JavaScript modules and third-party libraries

Install an npm package from a private registry with the ServiceNow IDE

Install Node Package Manager (npm) packages from a private registry as dependencies in your application to use them as third-party libraries.

Before you begin

Role required: admin

About this task

To install packages from private registries, you must configure your Package Manager user settings in the ServiceNow IDE. Then you can choose in which applications to install packages.

ServiceNow IDE uses the public npm registry (`https://registry.npmjs.org`) as its default package source. If your network blocks access to this registry, you must have access to an alternate registry to download packages and build applications in the ServiceNow IDE. If access to the public npm registry is blocked on your system, you must configure a private npm registry in your Package Manager user settings in the ServiceNow IDE.

Note: To install packages from a private registry, the registry must respond with the HTTP `Access-Control-Allow-Origin` header.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **App Development** > **ServiceNow IDE**.
2. Open a workspace.
3. Use one of the following keyboard shortcuts to open the command palette:
   - Windows: Ctrl-Shift-P
   - Mac: Cmd-Shift-P
4. Enter `Preferences: Open User Settings (JSON)` and press Enter.
5. Specify a private registry as the default registry or a scoped registry from which you can install packages.

Option	Example
Default registry	`"package-manager.defaultRegistry": "<private-registry-url>",`
Scoped registry	`"package-manager.scopedRegistries": [ { "scope": "@<scope>", "registry": "<private-registry-url>" }, ... ]`

Note: To install packages from public registries, you can specify a public registry as the default registry or a scoped registry in addition to any private registries.

6. Provide your credentials for accessing the private registry.

Option	Description
Basic authentication	Provide a user name and password to access the default registry that you specified. `"package-manager.basicAuth": [ { "registry": "<private-registry-url>", "user": "<user-name>", "pass": "<password>" }, ... ]`
Legacy authentication	Provide a legacy token to access the default registry that you specified. Legacy tokens are basic authentication credentials encoded in Base64 format. `"package-manager.legacyAuth": [ { "registry": "<private-registry-url>", "token": "<legacy-token>" },`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Description
	<pre>...]</pre>
Token authentication	Provide a bearer token to access the default registry that you specified. <pre>"package-manager.tokenAuth": [{ "registry": "<private-registry-url>", "token": "<token>" }, ...]</pre>

7. Save your changes to the `settings.json` file.
8. Install packages from the configured registries.
   a. Use one of the following keyboard shortcuts to open the command palette:
      - Windows: Ctrl-Shift-P
      - Mac: Cmd-Shift-P
   b. Enter `Package Manager: Install Dependencies` and press Enter.
   c. Select an application to install packages in as dependencies.

   Packages are installed in the `node_modules` directory.

Example: User settings for private registry access

In the following example, a user configured access to a private registry using basic authentication. The user also configured their settings to install packages from a public registry with the `@example` scope.

```
{
    "files.autoSave": "off",
    "package-manager.defaultRegistry": "<private-registry-url>",
    "package-manager.basicAuth": [
        {
        "registry": "<private-registry-url>",
            "user": "<user-name>",
            "pass": "<password>"
        }
    ],
    "package-manager.scopedRegistries": [
        {
            "scope": "@example",
            "registry": "<public-registry-url>"
        }
    ]
}
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

What to do next
Use the packages that you installed as third-party libraries in your application. For more information, see Use third-party libraries in applications in the ServiceNow IDE.

Create an application file from the File Categories view

Create an application file in an embedded ServiceNow AI Platform user interface from the File Categories view in the ServiceNow IDE.

Before you begin
Create, convert, or clone an application and add it to your workspace. For more information, see Adding applications in ServiceNow IDE.

Role required: admin

About this task
If you need to create some application metadata outside of source code, you can create application files from the File Categories view within the ServiceNow IDE. To create an application file from the ServiceNow IDE, an embedded ServiceNow AI Platform user interface opens in the editor.

Procedure
1. Navigate to **All** > **App Development** > **ServiceNow IDE**.
2. Open a workspace with an application.
3. From the Activity Bar, select the File Categories view or Apps view.
4. Select the Create New File icon (+).

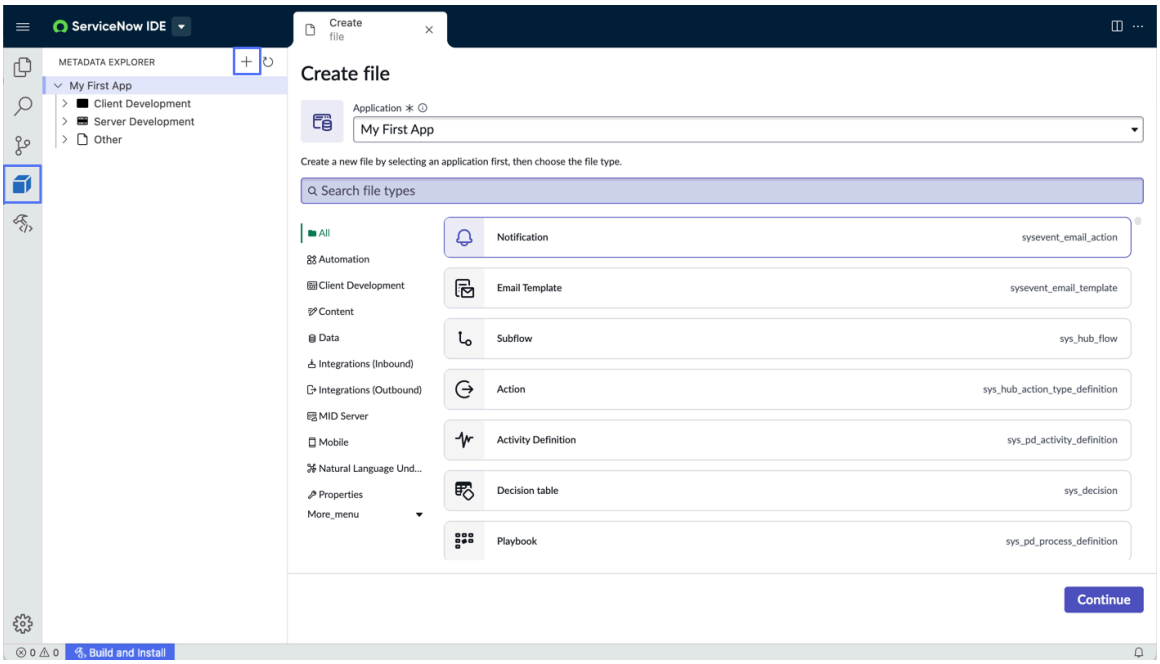
   Tip: Alternatively, you can right-click an existing metadata category or file type in an application and select **Create New File** to create an application file of that category or type.

5. From the Application menu, select an application from your workspace.
6. From the list of file types, select the type of file to create.

   Tip: Search for a file type or select a category to find the file type you need.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example



7. Select **Continue**.
8. In the embedded ServiceNow AI Platform user interface, fill in the fields to create the application file.
   The user interface and fields vary depending on the type of file that you created.
9. Select **Submit**.

Build and install an application in the ServiceNow IDE

Build an application to compile its source code and install application changes across an instance from the ServiceNow IDE.

Before you begin

Create, convert, or clone an application and add it to your workspace. For more information, see Adding applications in ServiceNow IDE.

Role required: admin

About this task

When you build an application, ServiceNow Fluent code is compiled into application metadata, and modules in the application are added to the EcmaScript Module [sys_module] table on the instance.

Procedure

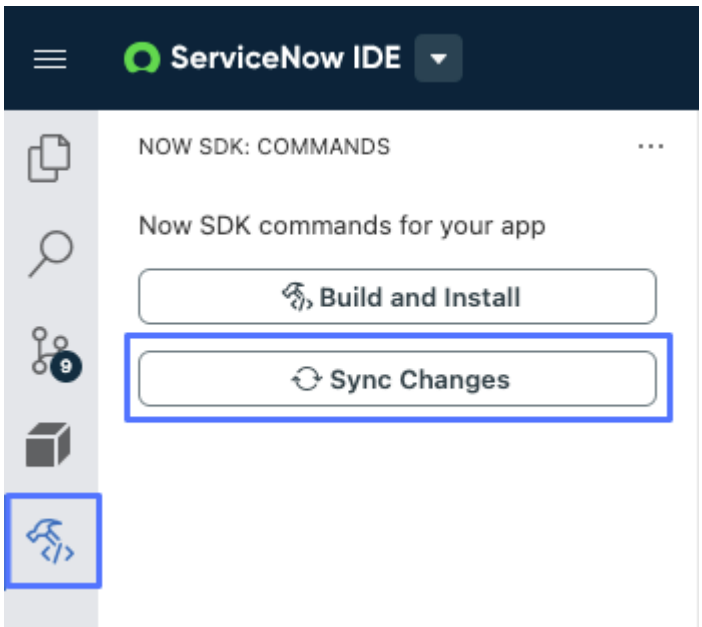
1. Navigate to **All** > **App Development** > **ServiceNow IDE**.
2. Open a workspace with an application.
3. Synchronize changes made to the application metadata from across the instance.
   Syncing an application confirms you have the latest changes before making additional changes. Only the changes made since the last time the application was synced are downloaded and transformed. For more information, see Synchronizing applications in the ServiceNow IDE.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. From the Activity Bar, select the Now SDK view ( ).

b. Select **Sync Changes**.
The active file that's open in the editor determines which application to synchronize. If no files are open, select the application to synchronize when prompted.

Example



4. From the Activity Bar, select the File Explorer view ( ).
5. Modify your application in one of the following ways and save your changes:
   - Define application metadata in code with ServiceNow Fluent in the ServiceNow IDE
   - Create and use JavaScript modules in applications in the ServiceNow IDE
   - Use third-party libraries in applications in the ServiceNow IDE
6. From the Status Bar, select **Build and Install**.

   The active file that's open in the editor determines which application to build. If no files are open, select the application to build when prompted.

   Trouble?
   In the status bar, you can see a message that confirms whether the build was successful or failed. If the build fails, review the output logs and any problems in your code from the panel.
7. **Optional:** Review your changes as application metadata.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. From the Activity Bar, select the File Categories view or Apps view.

b. Open the metadata files that you changed and review your changes from the embedded ServiceNow AI Platform user interfaces.

Tip: To see your changes in source code and metadata side by side, select the Split Editor icon () or right-click a tab and select one of the **Split** options.

Result

The application is updated across the instance to reflect your changes. Other users can modify the application metadata simultaneous with modifications to the source code. Also, other developers can reuse module code in other modules or scripts within an application.

Trouble?

If you need to ensure that the metadata on the instance matches the metadata in the installation package, you can uninstall and reinstall the application on the instance using the `Fluent: Reinstall Fluent App` command from the command palette. For more information, see ServiceNow IDE commands.

Warning: Changes to metadata on the instance that haven't been synced into your application in the ServiceNow IDE are removed.

What to do next

Use source control to stage, commit, and push your changes to a remote Git repository. For more information, see Using source control in the ServiceNow IDE.

ServiceNow IDE reference

Reference topics include information about ServiceNow Fluent APIs and ServiceNow IDE properties, roles, and more.

ServiceNow Fluent API reference

Use ServiceNow Fluent APIs to define the metadata that makes up scoped applications in source code with the ServiceNow IDE or ServiceNow SDK.

ServiceNow IDE commands

Use commands installed with ServiceNow IDE.

ServiceNow IDE properties

Use system properties installed with ServiceNow IDE.

ServiceNow IDE roles

Use roles installed with ServiceNow IDE.

ServiceNow IDE commands

Use commands from the command palette to perform many actions in the ServiceNow IDE.

The command palette includes commands unique to the ServiceNow IDE for creating and managing applications and commands from Visual Studio Code, such as Git commands for using source control. The following commands are unique to the ServiceNow IDE.

To open the command palette, use one of the following keyboard shortcuts:

- Windows: Ctrl-Shift-P
- Mac: Cmd-Shift-P

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

To close the command palette, press the Escape key.

ServiceNow IDE commands

Command	Description
Fluent: Apply Template for an existing Fluent App	Adds template files and directories for development in ServiceNow Fluent to an existing application.
Fluent: Build and Install	Build and install the application across the instance. Learn more: Build and install an application in the ServiceNow IDE
Fluent: Build with Now SDK	Build the application into an installable package.
Fluent: Convert an app to Fluent	Convert an existing application to support development in source code. **Note:** With ServiceNow IDE version 3.1.4 and earlier, the sn_glider.ide_fluent_admin role is required to use this command. Learn more: Convert an application with the ServiceNow IDE
Fluent: Create Fluent App	Create an application to develop in source code. Learn more: Create an application with the ServiceNow IDE
Fluent: Download script dependencies	Download script dependencies in your application to get type-ahead support for Glide APIs and script includes. Type definitions are downloaded into the `@types/servicenow` directory based on the scripts in your application. After downloading script dependencies, you can create a `tsconfig.json` file in the `src/fluent` directory that includes the type definitions. For example: `"include": [ "./**/*.client.js", "./**/*.server.js", "../../@types/servicenow/*.client.d.ts", "../../@types/servicenow/*.server.d.ts", ]` Learn more: Downloading dependencies with the ServiceNow SDK
Fluent: Force Install Fluent App in Instance	Install an existing build artifact, which might overwrite recent changes.
Fluent: Force Sync of Fluent App with all metadata	Synchronize all metadata in the application rather than only the metadata that changed since the last time that the application was synchronized. **Warning:** Force synchronizing all metadata in the application might overwrite any changes in the application that haven't been built and installed on the instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ServiceNow IDE commands (continued)

Command	Description
	Learn more: Synchronizing applications in the ServiceNow IDE
Fluent: Install Fluent App in Instance	Install the application across the instance without building it again.
Fluent: Reinstall Fluent App	Uninstall and reinstall the application on the instance to verify that the metadata on the instance matches the metadata in the installation package. The source files in the application aren't changed. **Warning:** Changes to metadata on the instance that haven't been synced into your application in the ServiceNow IDE are removed. If you have previous versions of modules in the EcmaScript Module [sys_module] table that aren't needed, re-installing an application removes previous versions of the application's modules from the table. **Note:** With ServiceNow IDE version 3.1.4 and earlier, the sn_glider.ide_fluent_admin role is required to use this command.
Fluent: Restart Fluent Server	Restart the ServiceNow Fluent language server.
Fluent: Sync Fluent App with changed metadata	Synchronize changes made to the application metadata since the last time that the application was synchronized. Learn more: Synchronizing applications in the ServiceNow IDE
Git: Manage Git credentials	Manage existing Git credentials for a Git domain and your user on the instance.
Git: Set IDE Git credentials	Configure basic or OAuth 2.0 credentials that are associated with your user to authenticate with a Git provider. Learn more: • Connect to a Git provider using basic authentication with the ServiceNow IDE • Connect to a Git provider using OAuth 2.0 with the ServiceNow IDE
File Categories: Focus on File Categories View	Navigate to the File Categories view. Learn more: Create an application file from the File Categories view
Package Manager: Install Dependencies	Install third-party dependencies into the `node_modules` directory. Learn more: Use third-party libraries in applications in the ServiceNow IDE

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ServiceNow IDE commands (continued)

Command	Description
Workspaces: Add Application to Workspace	Add an existing application to a workspace.
Workspaces: Add to Workspace	Create, open, or clone an application in a workspace.
Workspaces: Browse Workspaces	Browse through and open your ServiceNow IDE workspaces.
Workspaces: Create a Workspace	Create a ServiceNow IDE workspace. Learn more: Create a workspace in the ServiceNow IDE
Workspaces: Delete a Workspace	Delete a ServiceNow IDE workspace.
Workspaces: Rename a Workspace	Rename a ServiceNow IDE workspace.
Workspaces: Welcome	Navigate to the ServiceNow IDE home page, which lists all of your workspaces.

ServiceNow IDE properties

Manage how the ServiceNow IDE functions on an instance using the following system properties.

Note: To open the System Properties [sys_properties] table, enter `sys_properties.list` in the navigation filter. To add properties to the table, see Add a system property.

Properties for ServiceNow IDE

Property	Description
sn_glider.default_to_bundled_sdk	Specifies whether to use the bundled version or the latest version of the ServiceNow SDK when creating or converting applications in the ServiceNow IDE. If true, the version of the ServiceNow SDK that's bundled with a version of the ServiceNow IDE is used. If false, the latest version of the ServiceNow SDK is used. • Type: true | false • Default value: false • Location: Add the property to the System Property [sys_properties] table
sn_glider.enable_ide	Enables access to the ServiceNow IDE. If false, access to the ServiceNow IDE is turned off across the instance. • Type: true | false • Default value: true • Location: Add the property to the System Property [sys_properties] table

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties for ServiceNow IDE (continued)

Property	Description
sn_glider.fluent_convert_enabled	Enables converting existing applications that weren't created with the ServiceNow IDE or ServiceNow SDK to support development in source code from the ServiceNow IDE. • Type: true | false • Default value: true • Location: Add the property to the System Property [sys_properties] table • Learn more: Convert an application with the ServiceNow IDE
sn_glider.git.attachment.extension.binary	Defines a custom extension for attachment files with binary content types (for example, `application/octet-stream`) to override the default `gitdata` extension. The custom extension that you specify with this property must also be specified with the *`glide.attachment.extensions`* system property. **Note:** If you reuse an existing extension, confirm that the value you set passes MIME type validations for binary types. • Type: string • Default value: gitdata • Location: Add the property to the System Property [sys_properties] table • Learn more: ServiceNow IDE MID Server User [sn_glider.ide_git_user]
sn_glider.git.attachment.extension.text	Defines a custom extension for attachment files with text content types (for example, `text/plain`) to override the default `txt` extension. The custom extension that you specify with this property must also be specified with the *`glide.attachment.extensions`* system property. **Note:** If you reuse an existing extension, confirm that the value you set passes MIME type validations for text types. • Type: string • Default value: txt • Location: Add the property to the System Property [sys_properties] table • Learn more: ServiceNow IDE MID Server User [sn_glider.ide_git_user]
sn_glider.pinnedSdkVersion	Specifies the version of the ServiceNow SDK to use when creating or converting applications in the ServiceNow IDE.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties for ServiceNow IDE (continued)

Property	Description
	• Type: string • Default value: The latest version of the ServiceNow SDK • Location: Add the property to the System Property [sys_properties] table

ServiceNow IDE roles

The following roles are installed for use with ServiceNow IDE.

Note: Users with the Creator Studio Restricted User [sn_creatorstudio.restricted_user] role aren't able to access the ServiceNow IDE regardless of any other roles assigned.

ServiceNow IDE Administrator [sn_glider.admin]

Create and develop applications in the ServiceNow IDE.

For more information about granular admin roles, see Granular admin roles.

Contains Roles

List of roles contained within the role.

- sn_glider.create_app_admin
- sn_glider.read_admin
- sn_glider.write_admin

Groups

This role is assigned to no groups by default.

Special considerations

In addition to this role, the following roles are necessary for certain functionality in the ServiceNow IDE:

- credential_admin: Required to perform Git operations.
- sn_udc.admin: Required to modify application files from the File Categories view.
- sn_udc.basic_read: Used for read-only access to application files in the File Categories view.

ServiceNow IDE MID Server User [sn_glider.ide_git_user]

Assign administrator privileges to a MID server user to manage applications in source control from the ServiceNow IDE.

MID Server users must have the sn_glider.ide_git_user role or admin role to perform Git operations in the ServiceNow IDE. For more information, see Create the MID Server user and grant the role and Configure a MID Server to use source control with the ServiceNow IDE.

Contains Roles

None.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Groups

This role is assigned to no groups by default.

Special considerations

None.

ServiceNow SDK

Create and develop applications in source code locally in Visual Studio Code Desktop and upload changes to an instance using the ServiceNow software development kit (SDK).

Get started

Explore  Learn about developing applications with the ServiceNow SDK.	Configure  Install and configure the ServiceNow SDK locally.	Authenticate  Authenticate to ServiceNow instances.
Create  Create applications in source code.	Develop  Develop applications in source code.	Reference  Get details about the ServiceNow SDK CLI and ServiceNow Fluent APIs.

Which builder should I use to create an app?

Are you a developer who wants to use industry-standard development tools and processes?

The ServiceNow IDE and ServiceNow SDK support developing applications in source code with ServiceNow Fluent, creating JavaScript modules, and using third-party libraries. ServiceNow Fluent is a domain-specific programming language for creating application metadata in code.

The ServiceNow IDE is an implementation of Visual Studio Code for the Web on the ServiceNow AI Platform. The ServiceNow SDK uses Visual Studio Code Desktop locally. For more information, see Building applications in source code.

Are you a developer who wants more control in a centralized user interface?

Build apps smarter and deliver them faster with ServiceNow Studio. ServiceNow Studio empowers platform developers with a modern, unified environment for building on the ServiceNow AI Platform. ServiceNow Studio features streamlined navigation to applications

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

and metadata, integrated low-code tools, efficient tracking and packaging of development work that accelerates development processes and enhances productivity. For more information, see Exploring ServiceNow Studio.

Need a more general app but still want low-code options?

App Engine Studio lets you build a broader range of apps than Creator Studio without being a programming pro. For more information, see Exploring App Engine Studio.

Want to build an app easily, without code?

Creator Studio specializes in helping you craft request-fulfillment applications without writing code. For example, an application to request office supplies by filling out a form, and someone approves or denies your request. For more information, see Exploring Creator Studio.

Troubleshoot and get help

- To learn more about what's new in the Brazil release, see the .
- ServiceNow SDK and Fluent Creator Toolbox video
- ServiceNow SDK examples GitHub repository
- ServiceNow IDE, SDK, and Fluent forum in the ServiceNow Community
- ServiceNow IDE, SDK, and Fluent articles in the ServiceNow Community
- Search the Known Error Portal for known error articles
- Contact Customer Service and Support

Related topics

ServiceNow Fluent

JavaScript modules and third-party libraries

Exploring the ServiceNow SDK

Learn about developing applications in source code locally and installing changes on an instance with the ServiceNow SDK.

ServiceNow SDK overview

With the ServiceNow SDK, you can create and modify applications locally in Visual Studio Code Desktop and install those applications on a non-production instance beginning with the Washington DC release. By developing applications locally, you can work offline and use features that are available in Visual Studio Code but not in the ServiceNow IDE.

In Visual Studio Code, you can develop applications in source code using ServiceNow Fluent to define application metadata [sys_metadata], write custom JavaScript modules, or add third-party libraries.

The ServiceNow SDK includes a command-line interface (CLI) for managing changes between an instance and a local application. Using simple CLI commands, you can authenticate to an instance, create or convert an application, transform application metadata (XML) into ServiceNow Fluent code, build and install the application on the instance, and more.

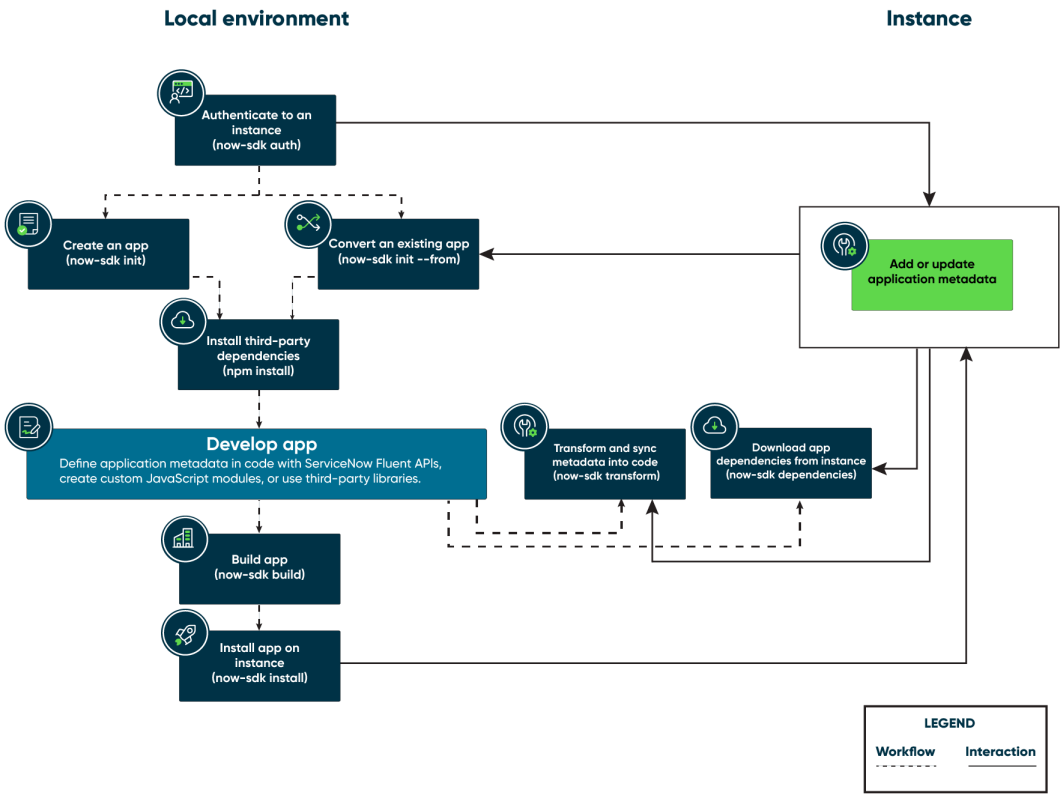
The ServiceNow SDK also serves as the application packaging service that builds applications in the ServiceNow IDE. For more information, see Exploring the ServiceNow IDE.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ServiceNow SDK workflow

The following infographic shows the workflow for developers to get started developing applications with the ServiceNow SDK.

Developing applications with the ServiceNow SDK



1. A developer installs the ServiceNow SDK locally. For more information, see Install the ServiceNow SDK in an application.
2. Using the CLI, the developer authenticates to a non-production instance to interface between their local environment and the instance with the `now-sdk auth` command. The developer must have the admin role on the instance.
3. The developer creates an application (`now-sdk init`) or converts an existing application from the instance (`now-sdk init --from`) for use with the ServiceNow SDK.
4. In Visual Studio Code, the developer can define application metadata in source code using ServiceNow Fluent, create custom JavaScript modules, or use third-party libraries.

   They can also download application and script dependencies from the instance to code against (`now-sdk dependencies`).
5. The developer builds the application, which compiles the source code and transforms it into application metadata for the instance (`now-sdk build`).
6. The developer installs the application on the instance (`now-sdk install`).
7. Other users can continue to modify the application metadata on the instance, and developers can reuse the code from modules in other modules or scripts within the application.
8. The developer downloads and transforms application metadata from the instance to get the latest updates to the application metadata locally (`now-sdk transform`), keeping it synchronized with the application on the instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

You can manage applications in source control with your preferred Git provider. To develop an application on another system with the ServiceNow SDK, other developers should clone the application from a remote GIt repository rather than downloading and transforming it from the instance.

ServiceNow SDK benefits

Benefit	Feature	Users
Develop applications offline, outside of an instance in Visual Studio Code and manage them in source control with your preferred Git provider.	Developing applications with the ServiceNow SDK Visual Studio Code documentation	Developers
Write source code to define the metadata that makes up ServiceNow applications	ServiceNow Fluent	Developers
Organize and reuse code within an application scope with custom JavaScript modules and third-party JavaScript utilities	JavaScript modules and third-party libraries	Developers

Configuring the ServiceNow SDK

Install or upgrade the ServiceNow SDK locally and get language processing and validation for ServiceNow Fluent in Visual Studio Code.

Install the ServiceNow SDK in an application

Install the ServiceNow software development kit (SDK) in a local application using Node Package Manager (npm).

Before you begin

Your system must meet the following minimum requirements to use the latest version of the ServiceNow SDK:

- Node.js v20.18.0
- npm v8.19.3

To install or upgrade Node.js or npm, see the installation instructions on the npm website.

Role required: admin

About this task

The ServiceNow SDK is available as an npm package from the public npm registry. The ServiceNow SDK supports integrating with ServiceNow instances beginning with the Washington DC release.

In the following procedure, you install the ServiceNow SDK within a new application using `npx` rather than globally. To install the ServiceNow SDK globally, use the `npm install --global @servicenow/sdk` command.

Note: If you use `npx` to install the ServiceNow SDK in application rather than globally, you must use `npx @servicenow/sdk [command]` rather than `now-sdk [command]` with the ServiceNow SDK CLI.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Open a command-line tool on your system.
2. Verify that your system meets the requirements.
   a. Enter `node -v` to check if you have Node.js installed and which version.
   b. Enter `npm -v` to check if you have npm installed and which version.
3. Create a local directory for your application.
4. Change directories into the directory for your application using the `cd` command.

   Example

   ```
   cd <path/to/directory>
   ```

5. Create an application with the ServiceNow SDK.

State	Steps
Authentication hasn't been configured	If you haven't added authentication credentials and an alias on your system yet, install the ServiceNow SDK with the `auth` command before creating an application with the `init` com mand. For example: `npx @servicenow/sdk auth --add <instance>` For more information, see Authenticating to a ServiceNow in stance with the ServiceNow SDK.
Authentication has been configured	If you have added authentication credentials, install the Servi ceNow SDK in new applications with the `init` command. For example: `npx @servicenow/sdk init` For more information, see Create an application with the Ser viceNow SDK or Convert an application with the ServiceNow SDK.

What to do next

Install the ServiceNow Fluent Language server to get language processing and validation for ServiceNow Fluent in Visual Studio Code. For more information, see Install the ServiceNow Fluent Language server in Visual Studio Code.

Upgrade the ServiceNow SDK

Upgrade to the latest version of the ServiceNow SDK for use with instances beginning with the Washington DC release.

Before you begin

Your system must meet the following minimum requirements to use the latest version of the ServiceNow SDK:

- Node.js v20.18.0
- npm v8.19.3

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

To install or upgrade Node.js or npm, see the installation instructions on the npm website.

Role required: admin

Procedure

Complete the following steps depending on whether you're updating a global installation of the ServiceNow SDK or the version used by an application.

Option	Steps
Application package	**a.** Open the `package.json` file for your application. **b.** In the `devDependencies` object, update the versions of the `@servicenow/sdk` and `@servicenow/eslint-plugin-sdk-app-plugin` packages. For example: `"devDependencies": { "@servicenow/sdk": "<version>", "@servicenow/glide": "26.0.1", "eslint": "8.50.0", "@servicenow/eslint-plugin-sdk-app-plugin": "<version>" }` **c.** Open a command-line tool on your system. **d.** Change directories into the directory for your application using the `cd` command: `cd <path/to/directory>` **e.** Enter the following command to verify the version of the ServiceNow SDK: `now-sdk -v`
Global package	**a.** Open a command-line tool on your system. **b.** Change directories into the directory for your application using the `cd` command: `cd <path/to/directory>` **c.** Enter the following command to upgrade to the latest version of the ServiceNow SDK: `npm update -g @servicenow/sdk` **d.** Enter the following command to verify the version of the ServiceNow SDK: `now-sdk -v`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

What to do next

Install the ServiceNow Fluent Language server to get language processing and validation for ServiceNow Fluent in Visual Studio Code. For more information, see Install the ServiceNow Fluent Language server in Visual Studio Code.

Use the ServiceNow SDK to authenticate to a ServiceNow instance. For more information, see Authenticating to a ServiceNow instance with the ServiceNow SDK.

Install the ServiceNow Fluent Language server in Visual Studio Code

Get language processing and validation for ServiceNow Fluent in Visual Studio Code with the ServiceNow Fluent Language server.

Before you begin

Role required: admin

About this task

The ServiceNow Fluent Language server provides code completion features and validation for ServiceNow Fluent language files (`.now.ts`) and for ServiceNow SDK configuration files (`now.config.json`).

Procedure

1. In Visual Studio Code, select the Extensions view from the Activity Bar.
2. Search for `ServiceNow® Fluent Language`.
3. Select **Install**.

What to do next

To use ServiceNow Fluent in applications with the ServiceNow SDK, see Define application metadata in code with ServiceNow Fluent and the ServiceNow SDK.

Related topics

ServiceNow Fluent

Authenticating to a ServiceNow instance with the ServiceNow SDK

Authenticate to a ServiceNow instance and store user credentials for accessing the instance on your system with the ServiceNow SDK.

You can use basic or OAuth 2.0 credentials to authenticate to an instance. To use OAuth 2.0 authentication, your instance must have ServiceNow IDE (version 1.1 or later) installed or have the required XML configuration imported. By default, instances on the Brazil release include ServiceNow IDE version 1.1.4 and support using OAuth 2.0 authentication with the ServiceNow SDK.

After authenticating to an instance with the ServiceNow SDK command-line interface (CLI), you can begin developing applications and installing them on your instance. A non-production instance should be used for application development.

Authenticate to a ServiceNow instance using basic authentication with the ServiceNow SDK

Use basic authentication to connect to a ServiceNow instance with the ServiceNow SDK.

Before you begin

Role required: admin

About this task

This procedure uses the ServiceNow SDK command-line interface (CLI). From a command-line tool, enter `now-sdk --help` to get information about the available commands and global

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

options. To get additional information about a command and its parameters, enter the command and `--help` or `-h`. For example, `now-sdk auth --help`. For more information about the CLI, see ServiceNow SDK CLI.

Note: To use non-interactive authentication in CI/CD pipelines, see the install command.

Procedure

1. Create a local directory for your application.
2. In Visual Studio Code, open the directory.
3. From the application directory, open an integrated Terminal window.
4. Specify the instance to authenticate to with the `auth` command.

```
npx @servicenow/sdk auth --add <instance>
```

Example
For example:

```
npx @servicenow/sdk auth --add
 https://myinstance.service-now.com
```

Note: Using the `npx` command installs the ServiceNow SDK in your application directory instead of globally.

5. Respond to the following series of prompts.

Prompt	Response
`Type of authentication to use`	Select `basic`.
`Alias for these credentials`	Enter an alias for your credentials and the instance. The alias can be used for authentication with the `init`, `transform`, `dependencies`, and `install` commands.
`The username to authenticate with the instance`	Enter your username for the instance. You must be assigned the admin role.
`The password to authenticate with the instance`	Enter your password.

Result
The alias and credentials are stored in the device keychain or credential manager on your system and are set as the default credentials.

What to do next
Add an application to your local system with the ServiceNow SDK. For more information, see Adding applications with the ServiceNow SDK.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Authenticate to a ServiceNow instance using OAuth 2.0 with the ServiceNow SDK

Use OAuth 2.0 authentication to connect to a ServiceNow instance with the ServiceNow SDK.

Before you begin

To use OAuth 2.0 authentication, your instance must have ServiceNow IDE (version 1.1 or later) installed or have the required XML configuration imported. By default, instances on the Brazil release include ServiceNow IDE version 1.1.4 and support using OAuth 2.0 authentication with the ServiceNow SDK.

Role required: admin

About this task

This procedure uses the ServiceNow SDK command-line interface (CLI). From a command-line tool, enter `now-sdk --help` to get information about the available commands and global options. To get additional information about a command and its parameters, enter the command and `--help` or `-h`. For example, `now-sdk auth --help`. For more information about the CLI, see ServiceNow SDK CLI.

Note: To use non-interactive authentication in CI/CD pipelines, see the install command.

Procedure

1. Create a local directory for your application.
2. In Visual Studio Code, open the directory.
3. From the application directory, open an integrated Terminal window.
4. Specify the instance to authenticate to with the `auth` command.

```
npx @servicenow/sdk auth --add <instance>
```

Example

For example:

```
npx @servicenow/sdk auth --add
 https://myinstance.service-now.com
```

Note: Using the `npx` command installs the ServiceNow SDK in your application directory instead of globally.

5. Respond to the following series of prompts.

Prompt	Response
`Type of authentication to use`	Select `oauth`.
`Alias for these credentials`	Enter an alias for your credentials and the instance. The alias can be used for authentication with the `init`, `transform`, `dependencies`, and `install` commands.

The ServiceNow SDK opens a web browser to authenticate with the instance.

6. Navigate to the web page that opens and log in to the instance if you aren't currently logged in.
7. Select **Accept** to allow the ServiceNow SDK to connect to the instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The page refreshes and includes an authentication code.

8. Select **Copy** to copy the authentication code provided.
9. In the command line, paste the authentication code.

Result
The alias and credentials are stored in the device keychain or credential manager on your system and are set as the default credentials.

What to do next
Add an application to your local system with the ServiceNow SDK. For more information, see Adding applications with the ServiceNow SDK.

Adding applications with the ServiceNow SDK

Add applications to your local system to get started with development in source code with the ServiceNow SDK.

- Create an application with the ServiceNow SDK

  Create an application that supports development in source code.

- Convert an application with the ServiceNow SDK

  Convert an existing application to support development in source code.

You can also clone an existing application that has already been converted to support development in source code from a remote Git repository in Visual Studio Code Desktop. For more information, see Introduction to Git in VS Code in the Visual Studio Code documentation.

Create an application with the ServiceNow SDK

Create an application to develop in source code with the ServiceNow SDK.

Before you begin
Use the ServiceNow SDK to authenticate to a ServiceNow instance. For more information, see Authenticating to a ServiceNow instance with the ServiceNow SDK.

Role required: admin

About this task
This procedure uses the ServiceNow SDK command-line interface (CLI). From a command-line tool, enter `now-sdk --help` to get information about the available commands and global options. To get additional information about a command and its parameters, enter the command and `--help` or `-h`. For example, `now-sdk auth --help`. For more information about the CLI, see ServiceNow SDK CLI.

Procedure
1. Create a local directory for your application.
2. In Visual Studio Code, open the directory.
3. From the application directory, open an integrated Terminal window.
4. Create an application following a guided set of prompts with the `init` command:

```
npx @servicenow/sdk init
```

Note: Using the `npx` command installs the ServiceNow SDK in your application directory instead of globally.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

5. Respond to the following series of prompts.

Prompt	Response
`Select a template`	Select a template that determines the default application structure, such as whether to create a full-stack application that supports UI development ands whether to use JavaScript or TypeScript to create modules.
`Name of ServiceNow Application`	Enter a name for the application.
`NPM package name`	Enter a name for the application package used in the `package.json` file. The package name must adhere to Node Package Manager (npm) package naming standards.
`Create a Global/Scoped App?`	Select whether to create a scoped or global application. ○ Scoped: Create a scoped application that is protected by identifying and restricting access to application files and data. ○ Global: Create an application in the global scope to allow it to be accessible to other global applications. Global applications are supported only with instances beginning with the Australia release.
`Scope name`	For scoped applications, enter the scope of the application. The scope name must be unique on the instance, begin with x_<prefix>, and be 18 characters or fewer. For more information, see Namespace identifier.

Example
In the following example, a scoped application named Example App (x_snc_example_app) is created.

```
$ npx @servicenow/sdk init
[now-sdk] Bootstrapping a new ServiceNow application project...
? Select a template: now-sdk + basic
? Name of ServiceNow Application:  Example App
? NPM package name:  example-app
? Scope name:  x_snc_example_app
[now-sdk] Application created successfully.
          Install the required dependencies with your preferred
 package manager before running "$now-sdk build".
          Ex: Run "npm install" if using npm.
```

6. Install the required third-party dependencies using your preferred package manager.

Example
For example, if you use Node Package Manager (npm), run `npm install`.

7. Build the application with the `build` command:

```
now-sdk build
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

8. **Optional:** Install the application on an instance with the `install` command:

```
now-sdk install --auth <alias>
```

Result
An application with the default application structure is available locally. For information about the application structure, see the Application structure section of the Building applications in source code topic.

If you installed the application successfully, it's available on the instance. For more information about installing applications, see Build and install an application with the ServiceNow SDK.

What to do next
In Visual Studio Code, start developing your application in source code with ServiceNow Fluent, writing custom JavaScript modules, or adding third-party libraries. For more information, see Developing applications with the ServiceNow SDK.

Convert an application with the ServiceNow SDK

Convert an existing application to support development in source code with the ServiceNow SDK.

Before you begin
Use the ServiceNow SDK to authenticate to a ServiceNow instance. For more information, see Authenticating to a ServiceNow instance with the ServiceNow SDK.

Role required: admin

About this task
Existing applications that weren't created with the ServiceNow IDE or ServiceNow SDK must be converted to support development in source code. Converting an application adds the necessary files and directories for developing it in source code. You can choose whether to convert existing application metadata into ServiceNow Fluent code. The application isn't changed on the instance until you build and install it on the instance.

This procedure uses the ServiceNow SDK command-line interface (CLI). From a command-line tool, enter `now-sdk --help` to get information about the available commands and global options. To get additional information about a command and its parameters, enter the command and `--help` or `-h`. For example, `now-sdk auth --help`. For more information about the CLI, see ServiceNow SDK CLI.

Procedure
1. Create a local directory for your application.
2. In Visual Studio Code, open the directory.
3. From the application directory, open an integrated Terminal window.
4. Convert an existing application from an instance or a local directory using the `init` command.

   With the `--from` parameter, provide a sys_id of an application on the instance or a path to a local directory that contains an application to convert to support development in source code.

```
npx @servicenow/sdk init --from <sys_id or path>
```

Example
For example:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
npx @servicenow/sdk init --from
 aadfdd904748a6500ff14ee4316d4369
```

Note: Using the `npx` command installs the ServiceNow SDK in your application directory instead of globally.

5. At the prompt, enter a name for the application package used in the `package.json` file.

   The package name must adhere to Node Package Manager (npm) package naming standards.

   Example
   In the following example, an application is downloaded from the instance and converted using its sys_id.

```
$ npx @servicenow/sdk init --from
 aadfdd904748a6500ff14ee4316d4369
[now-sdk] Bootstrapping a new ServiceNow application project...
[now-sdk] Please provide the missing required fields:
? NPM package name:  example-app
[now-sdk] Access Token has expired, refreshing token
[now-sdk] Downloading application package for
 aadfdd904748a6500ff14ee4316d4369 from
 https://myinstance.service-now.com
[now-sdk] Unzipping downloaded package...
[now-sdk] Successfully downloaded and unzipped application
 aadfdd904748a6500ff14ee4316d4369
[now-sdk] Summary of downloaded metadata files
Metadata: 13 (Total: 13)
  Copied To Metadata Directory::
    1. dictionary/x_snc_example_app_mytable.xml
    2. package_inventory.csv
    3. sys_app_aadfdd904748a6500ff14ee4316d4369.xml
    4. update/sys_db_object_7030a5d04748a6500ff14ee4316d4338.xml
    5. update/sys_dictionary_x_snc_example_app_mytable_null.xml
    6.
 update/sys_documentation_x_snc_example_app_mytable__en.xml
... and 7 more

[now-sdk] Successfully converted application.
[now-sdk] Install the required dependencies with your preferred
 package manager before running "$now-sdk build".
          Ex: Run "npm install" if using npm.
```

   The application is available locally with the default application structure and supports development in source code. Existing application metadata is downloaded into in the `metadata` directory. For information about the application structure, see the Application structure section of the Building applications in source code topic.

6. Install the required third-party dependencies using your preferred package manager.

   Example
   For example, if you use Node Package Manager (npm), run `npm install`.

7. **Optional:** Transform existing application metadata (XML) into ServiceNow Fluent code using the `transform` command.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

(Optional) With the `--from` parameter, provide a path to a local directory or file that contains metadata XML to transform into ServiceNow Fluent code.

```
now-sdk transform [--from <path>] [--directory <package path>]
 [--preview <flag>] [--auth <alias>]
```

Tip: You can set the `--preview` parameter to true to preview the transformed ServiceNow Fluent code from the command line without saving the changes.

Example
For example:

```
now-sdk transform --from metadata/update --auth devuser1
```

Application metadata is defined in ServiceNow Fluent code in the `src/fluent/generated` directory and removed from the specified directory.

Note: A limited number of metadata types, such as Metadata Snapshots [sys_metadata_link] and UX Assets [sys_ux_lib_asset], can't be represented as ServiceNow Fluent code and aren't transformed. These metadata types remain as metadata XML files in the `metadata` directory of your application.

8. Build the application with the `build` command.

```
now-sdk build
```

9. **Optional:** Update the application on the instance with the `install` command.

```
now-sdk install --auth <alias>
```

Result
The converted application is added to your local directory with the necessary files and directories to support development in source code. If you installed the application successfully, it's updated on the instance. For more information about installing applications, see Build and install an application with the ServiceNow SDK. After installing a converted application, the **Package JSON** field of the custom application record [sys_app] contains the path to the `package.json` file for the application.

New application metadata added after converting an application is automatically transformed into source code in the `src/fluent/generated` directory when you use the `transform` command. If metadata exists in the local application as both XML and source code, the XML version takes precedence when installed on the instance.

What to do next
In Visual Studio Code, start developing your application in source code with ServiceNow Fluent, writing custom JavaScript modules, or adding third-party libraries. For more information, see Developing applications with the ServiceNow SDK.

Developing applications with the ServiceNow SDK

Create and develop applications in source code locally in Visual Studio Code Desktop with the ServiceNow SDK. Build and install applications to make your changes available on an instance.

After installing the ServiceNow SDK, you can authenticate to an instance to create applications or convert existing applications using the ServiceNow SDK command-line interface (CLI).

In Visual Studio Code, start developing your application in source code with ServiceNow Fluent, writing custom JavaScript modules, or adding third-party libraries.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

When your changes are ready, build and install the local application on the instance with the ServiceNow SDK CLI. Download and transform changes to the application metadata from the instance to keep the application up to date locally before making additional changes.

Learn how to get started developing applications with the ServiceNow SDK in the following topics.

Related topics

Using TypeScript in JavaScript modules with the ServiceNow SDK

ServiceNow SDK CLI

ServiceNow Fluent API reference

Define application metadata in code with ServiceNow Fluent and the ServiceNow SDK

Define application metadata in code with ServiceNow Fluent and the ServiceNow SDK.

Before you begin

- Install the ServiceNow Fluent Language server to get language processing and validation for ServiceNow Fluent in Visual Studio Code. For more information, see Install the ServiceNow Fluent Language server in Visual Studio Code.
- Add an application to your local system with the ServiceNow SDK. For more information, see Adding applications with the ServiceNow SDK.

Role required: admin

About this task

ServiceNow Fluent is a domain-specific language (DSL) based on TypeScript for defining the metadata files [sys_metadata] that make up applications and includes APIs for the different types of metadata, such as tables, roles, ACLs, business rules, and Automated Test Framework tests. For more information about ServiceNow Fluent APIs and examples, see ServiceNow Fluent API reference.

Procedure

1. In Visual Studio Code, open your scoped application directory.
2. In the `src/fluent` directory, add a file with the `.now.ts` extension.
   You can define application metadata in a single file or as many `.now.ts` files as you want and organize files in directories within the `fluent` directory.

   Tip: In the `fluent` directory, you can refer to an example file named `index.now.ts`.

3. In the `.now.ts` file, use objects in the ServiceNow Fluent APIs to define metadata in the application.

   Example

   For example, to create a table [sys_db_object] in the application, use the Table API. The following example includes a simple table for to-do lists using a *`Table`* object with the necessary properties and values.

```
export const x_snc_example_to_do = Table({
    name: 'x_snc_example_to_do',
    schema: {
        deadline: DateColumn({ label: 'Deadline' }),
        state: StringColumn({
            label: 'State',
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
            choices: {
                ready: { label: 'Ready' },
                completed: { label: 'Completed' },
                inProgress: { label: 'In Progress' },
            },
        }),
        task: StringColumn({ label: 'Task', maxLength: 120 }),
    },
})
```

Note: You can use the Record API to define application metadata that doesn't have a dedicated API.

4. Add the required imports for the APIs from `@servicenow/sdk/core`.

   Example

   For example, to define a table with a date column and string column, import the corresponding objects to use from the Table API:

   ```
   import { Table, DateColumn, StringColumn }
    from '@servicenow/sdk/core'
   ```

5. **Optional:** For ServiceNow Fluent APIs that use server-side scripts, such as business rules, import code from JavaScript modules to call them from an object's *`script`* property.

   Example
   This example imports the `showStateUpdate` function, which can then be referenced from the *`script`* property.

   ```
   import { showStateUpdate } from '../server/script.js'
   ```

6. Save your changes.
7. **Optional:** Build and install your application to compile source code into application metadata and make your changes available on the instance. For more information, see Build and install an application with the ServiceNow SDK.

Trouble?

Tip: You can use the following directives in a code comment to help manage your code:

- `@fluent-ignore`: Suppresses ServiceNow Fluent diagnostic warnings and errors in the following line of code.
- `@fluent-disable-sync`: Turns off syncing changes to a ServiceNow Fluent object. Use before a call expression (for example, `Record({ ... })`) to turn off syncing for that object and its child objects. Only use this directive if you want to ignore changes made outside of the source code to the object and never update it when syncing.
- `@fluent-disable-sync-for-file`: Turns off syncing changes to a ServiceNow Fluent file (`.now.ts`). Use in the first line of the file to turn off syncing for all code in the file. Only use this directive if you want to ignore changes made outside of the source code to the file and never update it when syncing.

Example: Defining application metadata in source code with ServiceNow Fluent

In files with the `.now.ts` extension, use objects in the ServiceNow Fluent APIs to define metadata in the application. You must also include the required imports for the APIs from

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

`@servicenow/sdk/core`. For objects with server-side scripts, such as the *`BusinessRule`* object, you can import and use code from JavaScript modules.

Tip: You can use content from another file in ServiceNow Fluent APIs by referring to the file from a property using the syntax `Now.include('path/to/file')`. Referring to content in a different file instead of including the content inline can be helpful for records with fields that support JavaScript, HTML, CSS, and so on to make developing different file types easier with the appropriate syntax highlighting. The `Now.include` utility supports two-way synchronization so changes to fields from other ServiceNow AI Platform user interfaces are synced into the referenced file's source code and changes to the code are synced back to metadata across the instance.

The following example includes the definitions of a table, client script, and business rule in the application. The client script uses a script from the `client-script.js` file. The business rule uses a function from the `script.js` JavaScript module.

```
import '@servicenow/sdk/global'
import { BusinessRule, ClientScript, DateColumn, StringColumn,
 Table } from '@servicenow/sdk/core'
import { showStateUpdate } from '../server/script.js'

//creates todo table, with three columns (deadline, status and task)
export const x_snc_example_to_do = Table({
    name: 'x_snc_example_to_do',
    schema: {
        deadline: DateColumn({ label: 'Deadline' }),
        state: StringColumn({
            label: 'State',
            choices: {
                ready: { label: 'Ready' },
                completed: { label: 'Completed' },
                inProgress: { label: 'In Progress' },
            },
        }),
        task: StringColumn({ label: 'Task', maxLength: 120 }),
    },
})

//creates a client script that pops up 'Table loaded successfully!!' message everytime todo
 record is loaded
ClientScript({
    $id: Now.ID['cs0'],
    name: 'my_client_script',
    table: 'x_snc_example_to_do',
    active: true,
    appliesExtended: false,
    global: true,
    uiType: 'all',
    description: 'Custom client script generated by Now SDK',
    isolateScript: false,
    type: 'onLoad',
    script: Now.include('../client/client-script.js'),
})

//creates a business rule that pops up state change message whenever a todo record is
 updated
BusinessRule({
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    $id: Now.ID['br0'],
    action: ['update'],
    table: 'x_snc_example_to_do',
    script: showStateUpdate,
    name: 'LogStateChange',
    order: 100,
    when: 'after',
    active: true,
})
```

The client script referenced from the *ClientScript* object:

```
function onLoad() {
    g_form.addInfoMessage("Table loaded successfully!!")
}
```

The JavaScript module referenced from the *BusinessRule* object:

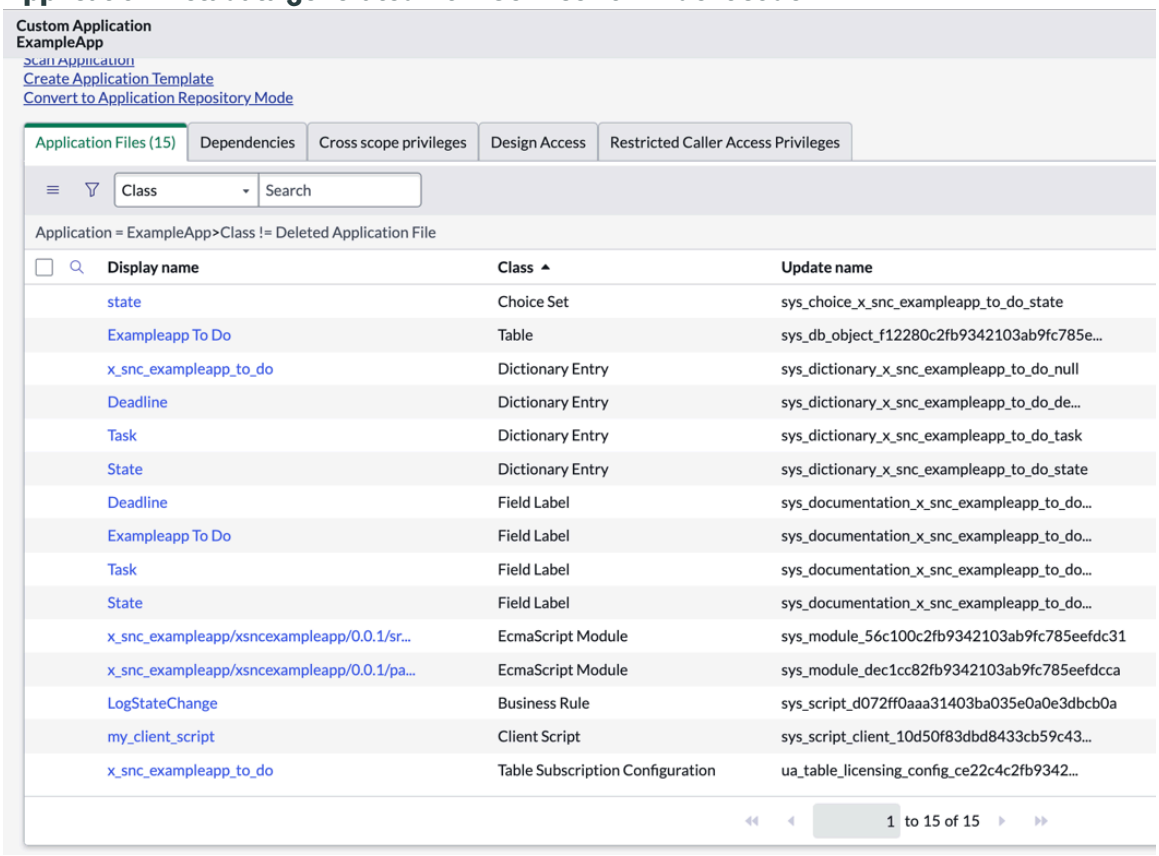
```
import { gs } from '@servicenow/glide'

export function showStateUpdate(current, previous) {
    const currentState = current.getValue('state')
    const previousState = previous.getValue('state')

    gs.addInfoMessage(`state updated from "${previousState}"
 to "${currentState}"`)
}
```

After building and installing the application, this source code generates the following application metadata files on the instance.

Application metadata generated from ServiceNow Fluent code



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Related topics

ServiceNow Fluent

Create and use JavaScript modules in applications with the ServiceNow SDK

Optimize your codebase by defining reusable code blocks with JavaScript modules and the ServiceNow SDK.

Before you begin

Add an application to your local system with the ServiceNow SDK. For more information, see Adding applications with the ServiceNow SDK.

Role required: admin

About this task

To learn about support for using JavaScript modules in scoped applications, including some limitations, see JavaScript modules and third-party libraries. For general information about the syntax used to create JavaScript modules, see the JavaScript modules page on the MDN Web Docs website.

Procedure

1. In Visual Studio Code, open your scoped application directory.
2. In the `src/server` directory of the application, create a JavaScript or TypeScript file to contain the module code you want to reuse.
3. **Optional:** Import server APIs or script includes to call them from your module.

   (Optional) Glide APIs can be imported from the `@servicenow/glide` package or their namespace in the package. Script includes can be imported from their application scope or the global scope in the `@servicenow/glide` package.

   For example:

   ```
   import { API } from "@servicenow/glide";
   import { API } from "@servicenow/glide/<namespace>";
   import { ScriptInclude } from "@servicenow/glide/<scope>";
   import { global } from "@servicenow/glide/global";
   ```

4. In the module, identify the code to export with *`export`* statements.

   You can use a named export or default export. Named exports can be variables, constants, functions, or classes whereas default exports can be functions or classes only.

   Example

   The following example is one way of adding a named export for multiple features (a function and a variable) in a module:

   ```
   export { myFunction, myVariable };
   ```

5. Use code from the exported module in other modules or server-side scripts.

File	Steps
Module	**a.** From the `src/server` directory, create or open a JavaScript module. **b.** Import the module code with *`import`* statements.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

File	Steps
	The following example is one way that you could import an exported feature in a module: `import { feature } from "path/to/module";` **Note:** To import code from one TypeScript file to another TypeScript file, you must include the `.ts` file extension. For example, `import { feature } from './module.ts'`. **c.** Call the module code from this module to reuse it.
Server-side script in source code	**a.** Create or open the definition of application metadata that includes a server-side script, such as a business rule, in source code (`.now.ts` file). **b.** In the *`script`* property, import and call the module code to reuse it. You can import a function or provide an inline script. ▪ Import an exported function, function expression, or default function. For example: `script: FunctionExport,` ▪ Inline scripts must use *`require`* statements to import the module code. For example: `script: ` `  const { process } = require('./dist/modules/server/handler.js')` `  process(request, response)`, For more information about server-side scripts in source code, see ServiceNow Fluent API reference.
Server-side script record	**a.** Open the record for a server-side script, such as a business rule. **b.** Import the module code with *`require`* statements. The following example is one way that you could import an exported feature in a script: `const { feature } = require("path/to/module");` **c.** Call the module code from this script to reuse it.

6. Save your changes.

What to do next

To use third-party libraries in a JavaScript module, see Use third-party libraries in applications with the ServiceNow SDK.

To build your application and add the modules to the EcmaScript Module [sys_module] table, see Build and install an application with the ServiceNow SDK.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Related topics

JavaScript modules and third-party libraries

Using TypeScript in JavaScript modules with the ServiceNow SDK

Use TypeScript when creating JavaScript modules with the ServiceNow SDK.

TypeScript uses static typing and type annotations to support developers catching errors earlier while writing code in Visual Studio Code.

For general information about using TypeScript, see the TypeScript Documentation on the typescriptlang.org website.

Use TypeScript in JavaScript modules

Use TypeScript in JavaScript modules by adding support for TypeScript in your application.

Before you begin

Install TypeScript version 4.8.4 or later. For installation instructions, see Download TypeScript on the typescriptlang.org website.

Role required: admin

About this task

Follow this procedure to update existing applications that weren't created using a TypeScript template to use TypeScript in modules. Beginning with the ServiceNow SDK version 3.0, applications support using TypeScript in JavaScript modules by default using default compiler options. To use a `tsconfig.json` file with custom options for transpiling TypeScript into JavaScript during the build process, configure the `tsconfigPath` parameter in the `now.config.json` file. If you want to use a custom transpilation step before building the application, configure the `modulePaths` parameter in the `now.config.json` file.

Procedure

1. In Visual Studio Code, open your scoped application directory.
2. Open the application's `package.json` file and in the `devDependencies` object, add the TypeScript package and version.

   Example

   ```
   "devDependencies": {
         "typescript": "<version>",
         "@servicenow/sdk": "2.0.0",
         "@servicenow/glide": "26.0.1",
         "eslint": "8.50.0",
         "@servicenow/eslint-plugin-sdk-app-plugin": "2.0.0"
      }
   ```

3. In your application's `src/server` directory, add a `tsconfig.json` file that defines the options for compiling the application.

   Example

   ```
   {
         "compilerOptions": {
               "rootDir": "./",
               "outDir": "../../dist/server",
               "module": "es2022",
               "target": "es2022",
   ```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
            "moduleResolution": "bundler",
            "allowJs": true,
            "declaration": false,
            "sourceMap": false,
            "skipLibCheck": true,
            "allowImportingTsExtensions": true,
            "noEmit": true
        },
        "include": [
            "./**/*.ts",
    "../../@types/**/*.modules.d.ts"
        ],
        "exclude": [
            "**/*.now.ts"
        ]
}
```

4. In the application's `now.config.json` file, set the `tsconfigPath` parameter to the location of the `tsconfig.json` for JavaScript modules. .

   Example

```
{
   "scope": "x_snc_example_app",
   "scopeId": "2f8400eb07426110f736e28f69d3017a",
   "name": "ExampleApp",
   "tsconfigPath": "./src/server/tsconfig.json"

}
```

5. In the `src/server` directory, add at least one `.ts` file to contain module code.
   For information about creating modules, see Create and use JavaScript modules in applications with the ServiceNow SDK.

What to do next

Build and install the application to compile the TypeScript files into JavaScript modules and add the modules to the EcmaScript Module [sys_module] table. For more information, see Build and install an application with the ServiceNow SDK.

Add type definitions for APIs

Get type-ahead support for APIs and scriptable objects outside of Glide APIs.

Before you begin

Role required: admin

About this task

Note: You can download type definitions for most APIs, script includes, and other scriptable objects using the `now-sdk dependencies` command. For more information, see Download module and script dependencies.

Follow this procedure to manually add types definitions that the `now-sdk dependencies` doesn't download. You can declare modules directly in the ServiceNow SDK application to stub access to the APIs for type-ahead support. These modules aren't packaged in the application package, but they can be tracked in a source control repository for the application and shared between developers.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. In Visual Studio Code, open your scoped application directory.
2. In your application, add a TypeScripe (`.ts`) file for type definitions.
3. In your TypeScript file, declare modules for APIs and scriptable objects.

   Example

   This example declares a module for an API using the API namespace (`sn_app_api`):

   ```
   declare module '@servicenow/glide/sn_app_api' {
    class AppStoreAPI {
     static canUpgradeAnyStoreApp(): boolean
    }
   }
   ```

   This example declares a module to access objects defined in script includes using the scope of the script include (`x_1234_scope`):

   ```
   declare module '@servicenow/glide/x_1234_scope' {
    class MyLogItemClass {
     myLogFunction()
    }
   }
   ```

4. In JavaScript modules in your application, import the declared modules.

   Example
   This example imports the declared module for the AppStoreAPI.

   ```
   import { gs } from '@servicenow/glide'
   import { AppStoreAPI } from '@servicenow/glide/sn_app_api'

   export const canUpgradeStoreApp = function () {
    var canUpgrade = AppStoreAPI.canUpgradeAnyStoreApp()
    if (canUpgrade) {
     gs.addInfoMessage(`You can upgrade store apps!`)
    } else {
     gs.addInfoMessage(`You cannot upgrade store apps!`)
    }
   }
   ```

   This example imports the declared module for the MyLogItemClass object.

   ```
   import { MyLogItemClass } from '@servicenow/glide/x_1234_scope'

   export const myLogFunction = function (status) {
    const myLogItem = new MyLogItemClass()
    myLogItem.myLogFunction(status)
   }
   ```

   Note: Modules can access only global scriptable objects or scriptable objects in the same application scope.

Use third-party libraries in applications with the ServiceNow SDK

Call third-party libraries in your application to use existing open-source functionality with the ServiceNow SDK.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

Create a JavaScript module. For more information, see Create and use JavaScript modules in applications with the ServiceNow SDK.

Role required: admin

About this task

Third-party libraries are added to applications as JavaScript modules. For general information about the syntax used to create JavaScript modules, see the JavaScript modules page on the MDN Web Docs website.

Important: You can't use third-party libraries that rely on unsupported functionality, such as unsupported APIs or ECMAScript features. For more information about unsupported functionality, see Third-party library support in Brazil and JavaScript modules and third-party libraries.

Procedure

1. In Visual Studio Code, open your scoped application directory.
2. From the application directory, open an integrated Terminal window.
3. Install the npm package for the library.

   Example

   ```
   npm install <package name>
   ```

   The library's packages are added to your application in the `node_modules` directory, and the library is added as a dependency in the `package.json` file for the application.

   ```
   "dependencies": {
       "<package name>": "<version>"
     }
   ```

4. In a JavaScript module, import the library using an *`import`* statement.

   Example

   In this example, the module includes a namespace import for the lodash module.

   ```
   import * as lodash from "lodash"
   ```

   In this example, the module includes a named import for the camelCase function in the lodash module.

   ```
   import camelCase from 'lodash'
   ```

5. Call code imported from the library in your module to reuse it.

What to do next

Build and install your changes on an instance. For more information, see Build and install an application with the ServiceNow SDK.

Related topics

JavaScript modules and third-party libraries

Downloading dependencies with the ServiceNow SDK

Download application dependencies and TypeScript definitions from an instance for IntelliSense and validation of scripts and ServiceNow Fluent code.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Throughout the development process, you should download dependencies and TypeScript definitions from an instance to support coding against those dependencies in an application.

To download all script and ServiceNow Fluent dependencies for an application, you can use the `dependencies` command with no parameters. If needed, provide the application directory and authentication alias too.

```
now-sdk dependencies [--directory <package path>] [--auth
 <alias>]
```

Download ServiceNow Fluent application dependencies

Download TypeScript definitions for dependencies in other application scopes to get IntelliSense support and validation for ServiceNow Fluent code.

Before you begin

Add an application to your local system with the ServiceNow SDK. For more information, see Adding applications with the ServiceNow SDK.

Role required: admin

About this task

This procedure uses the ServiceNow SDK command-line interface (CLI). From a command-line tool, enter `now-sdk --help` to get information about the available commands and global options. To get additional information about a command and its parameters, enter the command and `--help` or `-h`. For example, `now-sdk auth --help`. For more information about the CLI, see ServiceNow SDK CLI.

Procedure

1. In Visual Studio Code, open your scoped application directory.
2. In the `now.config.json` file, add the items that your application depends on in the `dependencies` object.

   The `dependencies` object has the following structure. You must specify the application scope and the dependency type and names or sys_ids. Only tables and roles can be specified by name. You can use a wildcard (`*`) to add all items from a specified table and scope.

   ```
   "dependencies": {
        "<scope>": {
           "<type>": ["<sys_id or name>"],
           ...
        },
        ...
   }
   ```

   Example

   For example:

   ```
   {
     "dependencies": {
        "global": {
           "tables": ["incident", "problem"],
           "roles": ["admin"],
           "sys_script_client": ["fa776f6d97700100f309124eda2975bc"]
        },
        "x_custom": {
           "tables": ["custom_table"],
   ```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
        "sys_security_acl": "*",
      }
    }
}
```

3. Save your changes.
4. From the application directory, open an integrated Terminal window.
5. Download dependencies of the application with the `dependencies` command.

Example

```
now-sdk dependencies --auth <alias> --fluent-only
```

Note: If you want to download script dependencies too, exclude the `--fluent-only` parameter.

The `dependencies` command downloads the dependencies specified in the `now.config.json` file and generates TypeScript definitions for them in the `@types/servicenow/fluent` directory with the `.d.now.ts` file extension. When building the application, these files aren't compiled like source code files.

What to do next

You can reference dependencies in ServiceNow Fluent files using the `#now:{scope}/{category}` subpath import format. For example:

```
// Import roles from global scope
import { role as globalRole } from '#now:global/security'

// Import roles from custom app scope
import { role as xExampleAppRole } from '#now:x_example_app/security'

// Use in ACLs, flows, etc.
Acl({
    $id: Now.ID['my_acl'],
    type: 'record',
    table: 'incident',
    operation: 'read',
    roles: [globalRole.admin, xExampleAppRole.xExampleAppCool],
})
```

Your application's `package.json` file must include the following `imports` configuration to do so:

```
{
  "imports": {
    "#now:*": "./@types/servicenow/fluent/*/index.js"
  }
}
```

Download module and script dependencies

Download TypeScript definitions for module and script dependencies to get IntelliSense support and validation for Glide APIs and script includes.

Before you begin

Add an application to your local system with the ServiceNow SDK. For more information, see Adding applications with the ServiceNow SDK.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Role required: admin

About this task

This procedure uses the ServiceNow SDK command-line interface (CLI). From a command-line tool, enter `now-sdk --help` to get information about the available commands and global options. To get additional information about a command and its parameters, enter the command and `--help` or `-h`. For example, `now-sdk auth --help`. For more information about the CLI, see ServiceNow SDK CLI.

Procedure

1. In Visual Studio Code, open your scoped application directory.
2. From the application directory, open an integrated Terminal window.
3. Download dependencies for scripts in the application with the `dependencies` command.

   Example

   ```
   now-sdk dependencies --auth <alias> --type-defs-only
   ```

   Note: If you want to download application dependencies too, exclude the `--type-defs-only` parameter.

   The `dependencies` command downloads type definitions for all Glide APIs and scans the modules and scripts in your application and creates type definitions for the script includes that they use. Type definitions are added in the `@types/servicenow` directory. To add any other type definitions, you can create them manually. For more information, see Add type definitions for APIs.

4. In the `src/fluent` directory, create a `tsconfig.server.json` file for server-side scripts.
   In the `include` object, include the type definitions that you downloaded for server-side Glide APIs (`glide.server.d.ts`) and script includes (`script-includes.server.d.ts`).

   Example

   ```
   {
     "compilerOptions": {
       "lib": [
         "ES2021"
       ],
       "noEmit": true,
       "checkJs": false,
       "allowJs": true,
       "noEmitHelpers": true,
       "esModuleInterop": false,
       "module": "None",
       "types": []
     },
     "include": [
       "./**/*.server.js",
       "../../@types/servicenow/*.server.d.ts",
     ]
   }
   ```

5. In the `src/fluent` directory, create a `tsconfig.client.json` file for client-side scripts.
   In the `include` object, include the type definitions that you downloaded for client-side Glide APIs (`glide.client.d.ts`).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example

```
{
  "compilerOptions": {
    "target": "ES6",
    "lib": [
      "DOM",
      "ES6"
    ],
    "checkJs": false,
    "allowJs": true,
    "noEmit": true,
    "noEmitHelpers": true,
    "esModuleInterop": false,
    "module": "None",
    "types": []
  },
  "include": [
    "./**/*.client.js",
    "../../@types/servicenow/*.client.d.ts",
  ]
}
```

6. In the `src/fluent` directory, create a `tsconfig.json` file.

   Note: This `tsconfig.json` file is separate from a `tsconfig.json` file used for applications that use TypeScript to create JavaScript modules.

   In the `references` object, add the paths to the `tsconfig.server.json` and `tsconfig.client.json` files.

   Example

```
{
  "files": [],
  "references": [
    {
      "path": "./tsconfig.server.json"
    },
    {
      "path": "./tsconfig.client.json"
    }
  ]
}
```

7. Save your changes.

Result

With this configuration, you can use the `.server.js` file extension for server-side scripts and `.client.js` file extension for client-side scripts and get type-ahead support during development.

Build and install an application with the ServiceNow SDK

Build and install a local application developed with the ServiceNow SDK on a ServiceNow instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

Add an application to your local system with the ServiceNow SDK. For more information, see Adding applications with the ServiceNow SDK.

Role required: admin

About this task

This procedure uses the ServiceNow SDK command-line interface (CLI). From a command-line tool, enter `now-sdk --help` to get information about the available commands and global options. To get additional information about a command and its parameters, enter the command and `--help` or `-h`. For example, `now-sdk auth --help`. For more information about the CLI, see ServiceNow SDK CLI.

Note: To build global applications with the ServiceNow SDK, you should use Node Package Manager (npm) as your package manager, especially if your application has more complex dependencies.

Procedure

1. In Visual Studio Code, open your scoped application directory.
2. From the application directory, open an integrated Terminal window.
3. Download changes to the application from the instance with the `transform` command to stay up to date with changes locally.

   Example

   ```
   now-sdk transform --auth <alias>
   ```

   Changes to application metadata are downloaded and synchronized into source code. If metadata exists in the local application as both XML and source code, the XML version takes precedence when installed on the instance.

   Note: Updates to JavaScript modules aren't included when downloading application metadata from your instance.

4. Build your application using the `build` command.

   ```
   now-sdk build
   ```

   The build artifacts are output to the `dist/app` directory, including the metadata as XML files in the `dist/app/update` directory.

5. Install your application using the `install` command.

   ```
   now-sdk install --auth <alias>
   ```

   Warning: If you set the `--reinstall` parameter to true with the `install` command, application metadata created by other developers on your instance can be removed. Any application metadata on your instance that is not present in the local installment package generated during the build process is removed during installment. Before building your application, use the `now-sdk transform` command to synchronize your local metadata with the metadata on your instance.

Result

The application is updated on your instance to reflect your local changes. ServiceNow Fluent code is compiled into application metadata, and modules in the application are added to the EcmaScript Module [sys_module] table on the instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: If you have previous versions of modules in the EcmaScript Module [sys_module] table that aren't needed, re-installing an application removes previous versions of the application's modules from the table.

ServiceNow SDK reference

Reference topics provide additional information about ServiceNow SDK CLI commands and ServiceNow Fluent APIs.

ServiceNow SDK CLI

Use the ServiceNow SDK command-line interface (CLI) to manage changes between a local application and the application on an instance.

Note: For the latest ServiceNow SDK CLI documentation, see the ServiceNow SDK CLI reference on GitHub.

From the command-line tool on your system, enter `now-sdk` to start the CLI and return a list of available commands or `now-sdk [command]` to begin using the ServiceNow SDK.

Note: If you use `npx` to install the ServiceNow SDK in application rather than globally, you must use `npx @servicenow/sdk [command]` rather than `now-sdk [command]` with the ServiceNow SDK CLI.

The CLI includes the following commands and global options:

Global options

Option	Description
--version, -v	Return the version of the CLI.
--help, -h	Return information about commands, sub-commands, and parameters.
--debug, -d	Return the debug logs generated with a command.

auth

Authenticate to an instance and store, update, or view user credentials for accessing an instance on your system.

The `auth` command has parameters for adding credentials, deleting credentials, listing credentials, and setting credentials to use by default. To use non-interactive authentication in CI/CD pipelines, see the install command.

For more information, see Authenticating to a ServiceNow instance with the ServiceNow SDK.

add

Store credentials in the device keychain or credential manager on your system.

The `auth` command has the following structure with the `--add` parameter:

```
npx @servicenow/sdk auth [--add <instance url>] [--type <auth
 method>] [--alias <alias>]
```

Note: Using the `npx` command installs the ServiceNow SDK in your application directory instead of globally.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Required parameters

Parameter	Type	Description	Default value
--add	String	The URL of the target instance to access and to which you install applications. The instance must be on the Washington DC release or later.	—

Optional parameters

Parameter	Type	Description	Default value
--type	String	The method to use to authenticate with the target instance. Specify `basic` for basic authentication or `oauth` for OAuth 2.0 authentication.	oauth
--alias	String	The alias for the instance and user credentials. The alias can be used for authentication with the `init`, `transform`, `dependencies`, and `install` commands.	—

For example:

```
npx @servicenow/sdk auth --add
 https://myinstance.service-now.com --type oauth --alias devuser1
```

delete

Remove credentials in the device keychain or credential manager on your system.

The `auth` command has the following structure with the `--delete` parameter:

```
now-sdk auth [--delete <alias or all>]
```

Required parameters

Parameter	Type	Description	Default value
--delete	String	The alias for the instance and user credentials. To delete all credentials, set this parameter to `--delete all`.	—

For example:

```
now-sdk auth --delete devuser1
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

list

View credentials saved in the device keychain or credential manager on your system. Passwords and authentication codes aren't returned.

The `auth` command has the following structure with the `--list` parameter:

```
now-sdk auth [--list]
```

Required parameters

Parameter	Type	Description	Default value
--list	String	Lists all available authentication credentials.	—

For example:

```
now-sdk auth --list
```

use

Set the credentials to be used by commands by default.

The `auth` command has the following structure with the `--use` parameter:

```
now-sdk auth [--use <alias>]
```

Required parameters

Parameter	Type	Description	Default value
--use	String	The alias for the instance and user credentials.	—

For example:

```
now-sdk auth --use devuser1
```

init

Create an application or convert an existing application to support development in source code. The application is added in the current directory.

To create an application, you can run the `init` command without any optional parameters (`now-sdk init`). To convert an existing application, you must include the `--from` parameter to provide the sys_id of an application on an instance or a path to a local directory that contains an application.

After initializing an application, you must install the required third-party dependencies using your preferred package manager before building the application. For example, if you use Node Package Manager (npm), run `npm install`.

The `init` command has the following structure:

```
npx @servicenow/sdk init [--from <sys_id or path>] [--appName
 <name>] [--packageName <name>] [--scopeName <name>] [--auth
 <alias>] [--template <template>]
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: Using the `npx` command installs the ServiceNow SDK in your application directory instead of globally.

Optional parameters

Parameter	Type	Description	Default value
--from	String	A sys_id of an application on the instance or a path to a local directory that contains an application to convert to support development in source code. Converting an application adds the necessary files and directories for using the ServiceNow SDK locally and downloads the application metadata. The application isn't changed on the instance until you build and install it on the instance. After installing a converted application, the **Package JSON** field of the custom application record [sys_app] contains the path to the `package.json` file for the application.	—
--appName	String	A name for the application.	—
--packageName	String	A name for the application package used in the `package.json` file. The package name must adhere to Node Package Manager (npm) package naming standards.	—
--scopeName	String	The scope of the application. The scope name must be unique on the instance, begin with x_<prefix>, and be 18 characters or fewer. For more information, see Namespace identifier.	—
--auth, -a	String	An alias for the credentials to use to authenticate to the instance.	If set, the default alias.
--template	String	A template for the default structure of the application. • base: An application with only the basic structure necessary for development in source code. • javascript.basic: An application configured for development in ServiceNow Fluent and JavaScript.	—

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Optional parameters (continued)

Parameter	Type	Description	Default value
		• javascript.react: An application configured for development in ServiceNow Fluent, JavaScript, and React. • typescript.basic: An application configured for development in ServiceNow Fluent and TypeScript. TypeScript source files in the `src/server` directory are transpiled into JavaScript modules. • typescript.react: An application configured for development in ServiceNow Fluent, TypeScript, and React. TypeScript source files in the `src/server` directory are transpiled into JavaScript modules. • partial.javascript.react: Adds template files and directories to an existing application for development in ServiceNow Fluent, JavaScript, and React. • partial.typescript.react: Adds template files and directories to an existing application for development in ServiceNow Fluent, TypeScript, and React.	

For example:

```
npx @servicenow/sdk init --from dbce0f6a3b3fda107b45b5d355e45af6
 --appName Example App --packageName example-app --scopeName
 x_snc_example --auth devuser1 --template base
```

For more information, see Create an application with the ServiceNow SDK or Convert an application with the ServiceNow SDK.

build

Compile source files and output build artifacts. Third-party library dependencies are converted into XML files that can be installed with the application.

The `build` command has the following structure:

```
now-sdk build <source> [--frozenKeys <flag>] [--errorOnConflict <flag>] [--skipClean
 <flag>]
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Optional parameters

Parameter	Type	Description	Default value
source	String	The path to the directory that contains the `package.json` file for your application. The `package.json` should be in the base directory of your application.	Current working directory
--frozenKeys	Boolean	An option to validate that the `keys.ts` file is up to date for continuous integration (CI) builds. If true and changes were made to the application's ServiceNow Fluent code, the `keys.ts` file isn't updated and the build fails. The `keys.ts` file is automatically generated in the `src/fluent/generated` directory.	false
--errorOnConflict	Boolean	An option to treat sys_id conflicts between ServiceNow Fluent code and metadata XML as errors instead of warnings.	false
--skipClean	Boolean	An option to skip cleaning the output directory before building the application, which can reduce the build time for incremental builds.	false

For example:

```
now-sdk build /path/to/package --frozenKeys true --errorOnConflict
 true --skipClean true
```

For more information, see Build and install an application with the ServiceNow SDK.

install

Package the build artifacts and install or update an application on an instance. Before using the `install` command, you must use the `build` command to generate an installable package.

The `install` command has the following structure:

```
now-sdk install [--source <package path>] [--reinstall
 <flag>] [--auth <alias>] [--open-browser <flag>] [--info
 <flag>] [--demoData <flag>] [--skip-flow-activation <flag>]
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Optional parameters

Parameter	Type	Description	Default value
--source	String	The path to the directory that contains the `package.json` file for your application. The `package.json` should be in the base directory of your application.	Current working directory
--reinstall, -r	Boolean	An option to uninstall and reinstall the application on the instance to ensure that the metadata on the instance matches the metadata in the installation package. ⚠ **Warning:** Metadata that is on the instance but not in your local application is removed. If you have previous versions of modules in the EcmaScript Module [sys_module] table that aren't needed, re-installing an application removes previous versions of the application's modules from the table.	false
--auth, -a	String	An alias for the credentials to use to authenticate to the instance.	If set, the default alias.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Optional parameters (continued)

Parameter	Type	Description	Default value
		Note: For CI/CD pipelines, you can set the following environment variables to authenticate with an instance at runtime: • SN_SDK_NODE_ENV: Set to `SN_SDK_CI_INSTALL` to enable CI server support. • SN_SDK_INSTANCE_URL: The URL of the target instance to access and to which you install applications. • SN_SDK_AUTH_TYPE: Set to `oauth_client_credentials` to enable OAuth 2.0 authentication. The default value is `basic` for basic authentication. • SN_SDK_CLIENT_ID: An OAuth 2.0 client ID. • SN_SDK_CLIENT_SECRET: An OAuth 2.0 client secret. • SN_SDK_USER: A username to access the instance using basic authentication. • SN_SDK_USER_PWD: The password associated with the user.	
--open-browser, -b	Boolean	An option to open the application record in your default browser after successfully installing the application.	false
--info, -i	Boolean	An option to return details about the most recent installation of this application, such as the status and records updated. When this parameter is used, the application isn't installed.	false
--demoData	Boolean	An option to install demo data with an application.	true

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Optional parameters (continued)

Parameter	Type	Description	Default value
--skip-flow-activation	Boolean	An option to skip activating, or publishing, flows when installing an application. By default, flows are activated on the instance when an application is installed. **Note:** ServiceNow IDE version 4.1.1 or later must be installed to activate flows with this command.	false

For example:

```
now-sdk install --source /path/to/package --reinstall false
 --auth devuser1 --open-browser true --info true --demoData
false --skip-flow-activation true
```

For more information, see Build and install an application with the ServiceNow SDK.

dependencies

Download application dependencies and TypeScript definitions from an instance to enable IntelliSense and code validation for an application.

The `dependencies` command downloads dependencies and TypeScript definitions for scripts and ServiceNow Fluent code:

- For scripts, this command downloads TypeScript definitions for all Glide APIs and scans the modules and scripts in your application and generates type definitions for the script includes that they use. Type definitions are added in the `@types/servicenow` directory. After downloading script dependencies, you must update your `tsconfig.json` file to include the type definitions.
- For ServiceNow Fluent, this command downloads the dependencies listed in an application's `now.config.json` file and generates TypeScript definitions for them in the `@types/servicenow/fluent` directory.

To download all script and ServiceNow Fluent dependencies for an application, you can use the `dependencies` command with no parameters. If needed, provide the application directory and authentication alias too.

The `dependencies` command has the following structure:

```
now-sdk dependencies [--directory <package path>] [--auth
 <alias>] [--type-defs-only <flag>] [--fluent-only <flag>] [--add <table> <sys_ids or names>]
[--scope <name>]
```

Optional parameters

Parameter	Type	Description	Default value
--directory	String	The path to the directory that contains the `package.json` file for your application.	Current working directory

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Optional parameters (continued)

Parameter	Type	Description	Default value
		The `package.json` should be in the base directory of your application.	
--auth, -a	String	An alias for the credentials to use to authenticate to the instance.	If set, the default alias.
--type-defs-only	Boolean	An option to download TypeScript definitions for only script dependencies. Script types are downloaded in the `@types/servicenow` directory.	false
--fluent-only	Boolean	An option to download dependencies and TypeScript definitions for only ServiceNow Fluent dependencies from other application scopes. You must list an application's dependencies in its `now.config.json` file. ServiceNow Fluent types are generated in the `@types/servicenow/fluent` directory.	false
--add	String	Adds the dependencies that you want to download to the application's `now.config.json` file. You must specify the type of dependencies to add using the table name and then the names or sys_ids of each item to add, separated by spaces. `--add <table_name> <sys_id1> <sys_id2> <sys_id3>` • To add table dependencies, you can use the `table` alias and names of tables instead of the sys_ids. For example: `--add tables incident problem change_request` • To add role dependencies, you can use the `role` alias and	—

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Optional parameters (continued)

Parameter	Type	Description	Default value
		names of roles instead of the sys_ids. For example: `--add roles admin user itil` Use a wildcard (`*`) to add all items from a specified table and scope. For example: `--add sys_security_acl "*"`	
sysIds	Array	A list of sys_ids of dependencies to download and generate TypeScript definitions from. This parameter only applies if you use the `--add` parameter.	—
--scope	String	The application scope from which to download dependencies. This parameter is required if you use the `--add` parameter.	—

For example:

```
now-sdk dependencies --directory /path/to/package --auth devuser1
 --add sys_ui_view fa776f6d97700100f309124eda2975bc --scope global
```

For more information, see Downloading dependencies with the ServiceNow SDK.

transform

Download application metadata (XML) from the instance and transform the metadata into ServiceNow Fluent source code to synchronize the application changes on the instance into your local application.

After initializing an application, you can run the `transform` command without any parameters to transform new application metadata from the instance into source code in the `src/fluent/generated` directory and synchronize changes to metadata into source code in the `src/fluent` directory. To transform metadata that existed when the application was initialized into source code, use the `--from` parameter to provide the path to a local directory or file that contains XML. If metadata exists in the local application as both XML and source code, the XML version takes precedence when installed on the instance.

The `transform` command has the following structure:

```
now-sdk transform [--from <path>] [--directory <package path>]
 [--auth <alias>] [--format <flag>] [--table <name>]
```

Optional parameters

Parameter	Type	Description	Default value
--from	String	A path to a local directory or file that contains metadata XML to	—

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Optional parameters (continued)

Parameter	Type	Description	Default value
		transform into ServiceNow Fluent code.	
--directory	String	The path to the directory that contains the `package.json` file for your application. The `package.json` should be in the base directory of your application.	Current working directory
--auth, -a	String	An alias for the credentials to use to authenticate to the instance.	If set, the default alias.
--format, -f	Boolean	An option to format new and updated ServiceNow Fluent source code when it's transformed.	true
--table	String	A comma-separated list of table names to transform to ServiceNow Fluent, including child tables. For example, `--table sys_script,sys_security_acl --from ./metadata`.	—

For example:

```
now-sdk transform --from metadata/update
 --directory /path/to/package --auth devuser1 --format true --table
sys_security_acl
```

For more information, see Convert an application with the ServiceNow SDK or Build and install an application with the ServiceNow SDK.

move

Transform application metadata (XML) in the global scope from an instance into ServiceNow Fluent source code in a local global application.

To customize the global metadata, the global application in which you want to move metadata must already be installed on the instance.

The `move` command has the following structure:

```
now-sdk move [--ids <sys_ids>] [--source <package path>] [--auth
 <alias>]
```

Required parameters

Parameter	Type	Description	Default value
--ids	Array	A list of sys_ids of global application metadata to move into the local global application and	—

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Required parameters (continued)

Parameter	Type	Description	Default value
		transform into ServiceNow Fluent code.	

Optional parameters

Parameter	Type	Description	Default value
--source	String	The path to the directory that contains the `package.json` file for your application. The `package.json` should be in the base directory of your application.	Current working directory
--auth, -a	String	An alias for the credentials to use to authenticate to the instance.	If set, the default alias.

For example:

```
now-sdk move
 --
ids 9f4e7402317d470da09ac34c0693d02d,c4e34f348b1843899ca5efb2a31
666f0,5057309fccef409cb30692c86a9aaac6 --auth devuser1
```

download

Download all application metadata (XML) from an application on an instance to compare with the metadata in your local application.

Updates to JavaScript modules aren't included when downloading application metadata from your instance.

The `download` command has the following structure:

```
now-sdk download <directory> [--source <package path>]
 [--incremental <flag>]
```

Required parameters

Parameter	Type	Description	Default value
directory	String	A path to any directory in which to download the metadata. **Note:** This directory should be a different directory from the `metadata` directory in your application.	—

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Optional parameters

Parameter	Type	Description	Default value
--source	String	The path to the directory that contains the `package.json` file for your application. The `package.json` should be in the base directory of your application.	Current working directory
--incremental	Boolean	An option to download only changes to application metadata made on the instance and recorded in the Customer Updates [sys_update_xml] table.	false

For example:

```
now-sdk download /path/to/directory --source /path/to/package
 --incremental true
```

clean

Remove the build artifacts that were output with the previous build.

The `clean` command has the following structure:

```
now-sdk clean <source>
```

Optional parameters

Parameter	Type	Description	Default value
source	String	The path to the directory that contains the `package.json` file for your application. The `package.json` should be in the base directory of your application.	Current working directory

For example:

```
now-sdk clean /path/to/package
```

pack

Package the build artifacts that were output with the previous build into an installable ZIP file.

The `pack` command has the following structure:

```
now-sdk pack <source> [--skip-package-inventory <flag>]
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Optional parameters

Parameter	Type	Description	Default value
source	String	The path to the directory that contains the `package.json` file for your application. The `package.json` should be in the base directory of your application.	Current working directory
--skip-package-inventory	Boolean	An option to skip generating the `package_inventory.csv` file in the packaged artifact. The `package_inventory.csv` is used to optimize performance during upgrades.	false

For example:

```
now-sdk pack /path/to/package --skip-package-inventory true
```

explain

View documentation and examples for a ServiceNow Fluent API.

The `explain` command has the following structure:

```
now-sdk explain <topic> [--list <keyword>] [--format <output type>] [--peek <flag>]
```

Required parameters

Parameter	Type	Description	Default value
topic	String	A keyword or topic name to search for to view related ServiceNow Fluent API, guide, or skill documentation, such as `table-api` or `table-guide`. To get a list of available topics, use the `--list` parameter.	—

Optional parameters

Parameter	Type	Description	Default value
--list, -l	Boolean	An option to view a list of available topics, which you can filter by providing a topic name or keyword, such as `now-sdk explain flow --list`. The list returned includes matches that contain the keyword and possibly related matches.	—

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Optional parameters (continued)

Parameter	Type	Description	Default value
--format	String	An option to format the output for the terminal (`pretty`) or as Markdown (`raw`).	pretty
--peek, -p	Boolean	An option to display a short description of a topic instead of the complete topic content. You can use this parameter with the `--list` parameter to display summaries of the listed topics.	—

For example:

```
now-sdk explain uipage-api --format raw
```

query

Runs a read-only REST Table API query on an instance to return a filtered list of records.

The `query` command can help with referring to instance data while writing ServiceNow Fluent code, such as looking up sys_ids, inspecting table schemas, checking existing records, and reading choice values. You can use this command with the `explain` command to provide an AI agent with instance data.

The `query` command has the following structure:

```
now-sdk query <table> --query <encoded query> [--limit <number>]
 [--offset <number>] [--fields <field list>] [--display-value
 <value>] [--exclude-reference-link <flag>] [--no-count <flag>]
 [--timeout <time>] [--view <view>] [--query-category <category>]
 [--query-no-domain <flag>] [--auth <alias>] [--output <format>]
```

Required parameters

Parameter	Type	Description	Default value
table	String	The name of a table to query, such as incident or sys_dictionary.	—
--query, -q	String	An encoded query string (sysparm_query) used to filter records, such as `"active=true^priority<=2"`.	—

Optional parameters

Parameter	Type	Description	Default value
--limit	Number	The maximum number of records to return per page (sysparm_limit).	100
--offset	Number	The starting record offset for pagination (sysparm_offset). If --output is `json`, use the `nextOffset` value returned in	0

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Optional parameters (continued)

Parameter	Type	Description	Default value
		a previous response to fetch the next page.	
--fields, -f	String	A comma-separated list of fields to return (sysparm_fields). If omitted, all fields are returned.	—
--display-value	String	An option to return display values instead of raw values (sysparm_display_value). Specify `true` to return display values, `false` to return raw values, or `all` to return both.	false
--exclude-reference-link	Boolean	An option to exclude reference link metadata in the response (sysparm_exclude_reference_link).	true
--no-count	Boolean	An option to skip calculating the total record count (sysparm_no_count). Skipping the count can improve performance for large tables.	false
--timeout	Number	The per-request timeout in milliseconds. Each page fetch is bounded by this value.	30000
--view	String	A UI view used to determine which fields to return (sysparm_view).	—
--query-category	String	A query category for extended queries (sysparm_query_category).	—
--query-no-domain	Boolean	An option to ignore domain separation when querying (sysparm_query_no_domain).	false
--auth, -a	String	An alias for the credentials to use to authenticate to the instance.	If set, the default alias.
--output, -o	String	The output format for the response. Specify `json` to paginate the response and return a machine-readable JSON envelope with `ok`, `hasMore`, `nextOffset`, and `records` fields. When this parameter is used, log output is suppressed.	—

For example:

```
now-sdk query incident --query 'active=true^priority<=2'
 --limit 50 --fields 'number,short_description,priority' --auth
 devuser1 --output json
```

For more information, see Table API.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ServiceNow Fluent API reference

Use ServiceNow Fluent APIs to define the metadata that makes up scoped applications in source code with the ServiceNow IDE or ServiceNow SDK.

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

Related topics

ServiceNow Fluent

ServiceNow Fluent language constructs

ServiceNow Fluent language constructs provide additional functionality for development in source code with ServiceNow Fluent APIs.

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

Now.ID

The `Now.ID` construct is used to specify human-readable unique IDs for metadata defined in source code. `Now.ID` uses the format `Now.ID['String' or Number]` with the `$id` property in ServiceNow Fluent APIs. When you build an application, the ID is hashed into a unique sys_id.

You can use `Now.ID` only to define IDs for metadata and not to reference other metadata in an application. To reference other metadata within the same application, you can assign the object to a `const` variable and reference the variable identifier in other objects.

```
import { Role } from "@servicenow/sdk/core"

const managerRole = Role({
   $id: Now.ID['manager_role'],
   name: 'x_snc_example.manager'
})

Record({
  table: 'some_table',
  data: {
     role: managerRole //"role" is a reference field to sys_user_role
  }
})
```

Now.ref

The `Now.ref` construct is used to reference metadata in a different application that's not defined in source code. `Now.ref` uses the format `Now.ref['table', 'sys_id' or {column: 'value'}, {column: 'value'}]` with Reference properties in ServiceNow Fluent APIs.

```
import { Role } from "@servicenow/sdk/core"

Role({
    name: 'x_test.admin',
    contains_roles: [
        'x_test.manager',
        Now.ref('sys_user_role', { name: 'x_test.itil' }),            //
Coalescing ID reference
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
        Now.ref('sys_user_role', '${itomId}'),                    //
GUID-based reference
        Now.ref('sys_user_role', '3D82d1a88947942a90b6d8aa25126d439b',
 { name: 'x_test.csm' }),     // GUID with coalescing ID reference
    ],
})
```

Now.include

The *Now.include* construct is used to refer to text content in another file in the same application rather than including the content inline. *Now.include* uses relative file paths in the format Now.include('./path/to/file') with any property in ServiceNow Fluent APIs. Using *Now.include* allows you to code with the appropriate syntax highlighting for the file type and is especially helpful with properties that support other languages like JavaScript, HTML, CSS, and more.

Now.include supports two-way synchronization so that changes to fields from other ServiceNow AI Platform user interfaces are synced into the referenced file's source code and changes to the source code are synced back to metadata across the instance.

```
import { ScriptInclude } from '@servicenow/sdk/core'

ScriptInclude({
    $id: Now.ID['my-script-include'],
    name: 'MyScriptInclude',
    active: true,
    apiName: 'x_scriptincludes.MyScriptInclude',
    script: Now.include('./MyScriptInclude.server.js')  //The actual content of the
"script" field is contained in the file specified
})
```

Now.attach

The *Now.attach* construct is used to refer to an image file in the same application and attach it to records associated with metadata defined in source code. *Now.attach* uses relative file paths in the format Now.attach('./path/to/file') with properties that support user images (the user_image field type) in ServiceNow Fluent APIs. *Now.attach* supports the following file formats: .jpg, .jpeg, .png, .gif, .bmp, .ico, and .svg.

Now.attach supports two-way synchronization so that changes to image files from other ServiceNow AI Platform user interfaces are synced into the application in source code and changes to image files in source code are synced back to metadata across the instance.

```
import { Record } from '@servicenow/sdk/core'

Record({
    $id: Now.ID['sample-service-portal'],
    table: 'sp_portal',
    data: {
        title: 'Sample Portal',
        url_suffix: 'sample',
        icon: Now.attach('../../assets/servicenow.jpg')
    }
})
```

You can also reuse an image in an application by assigning the *Now.attach* construct to a *const* variable and referencing the variable identifier in other properties. For example:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
import { Record } from '@servicenow/sdk/core'

const image = Now.attach('../../assets/servicenow.jpg')

Record({
    $id: Now.ID['test'],
    table: 'sp_portal',
    data: {
        title: 'Test Portal',
        url_suffix: 'test',
        icon: image,
        logo: image,
    }
})
```

Access Control List API - ServiceNow Fluent

The Access Control List API defines access control lists [sys_security_acl] that secure parts of an application.

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

For general information about access control lists (ACLs), see Access Control Lists (ACLs).

Related topics

ServiceNow Fluent

ACL object

Configure a custom ACL rule [sys_security_acl] to secure access to new objects or to change the default security behavior.

ACLs must include one or more roles, a security attribute, a condition, or a script.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
operation	String	Required. The operation that this ACL rule secures. An ACL rule can only secure one operation. To secure multiple operations, create a separate ACL rule for each. The operation must be `execute` if the *type* property is client_callable_flow_object, client_callable_script_include, graphql, processor, or rest_endpoint. Valid values:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		• execute: Allow users to execute scripts on a record or UI page. • create: Allow users to insert new records (rows) into a table. • read: Allow users to display records from a table. • write: Allow users to update records in a table. • delete: Allow users to remove records from a table or drop a table. • conditional_table_query_range: Enables users to give partial ACL-access based on read ACLs. Created for the tables that have the read ACLs without Data condition and script. • data_fabric: Enable users to allow Data Fabric connectors to access data on a particular table. • query_match: Enables users to submit match queries (such as "is", "is not", and "is empty"). • query_range: Enables users to submit range queries (such as "starts with", "ends with", and "contains") and sorting is unrestricted. • edit_task_relations: Allow users to extend the Task [task] table. • edit_ci_relations: Allow users to extend the Configuration Item [cmdb_ci] table. • save_as_template: Allow users to save a record as a template. • add_to_list: Allow users from viewing or personalizing specific columns in the list mechanic. Conditions and scripts aren't supported. • report_on: Allow users to report on tables. • list_edit: Allow users to update records (rows) from a list. • report_view: Allow users to report on field ACLs. • personalize_choices: Allow users to configure the table or field.
type	String	Required. The type of object that this ACL rule secures. The type determines which operations are available. After creating an ACL rule, if you want to change the type, you must delete the ACL and create a new one with the correct type. Valid values: • record • rest_endpoint

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		• ui_page • processor • graphql • pd_action • ux_data_broker • ux_page • ux_route • client_callable_flow_object • client_callable_script_include Default: record
active	Boolean	Flag that indicates whether the ACL rule is enforced. Valid values: • true: The ACL rule is enforced. • false: The ACL rule isn't enforced. Default: true
adminOverrides	Boolean	Flag that indicates whether users with the admin role automatically pass the permissions check for this ACL rule. Valid values: • true: Administrators automatically pass the permissions check. If true, admin users pass the permissions check regardless of what script or role restrictions apply. However, the nobody role, which only ServiceNow personnel can assign, takes precedence over the admin override option. If an ACL is assigned the nobody role, admin users cannot access the resource even when `adminOverrides` is true. For more information, see Base system roles. • false: Administrators must meet the permissions defined in this ACL rule to gain access to the secured object. Use the `condition` or `script` properties to create a permissions check that administrators must pass. Default: true
script	Script	A custom script that defines the permissions required to access the object. This property supports a function from a JavaScript module, a reference to another

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		file in the application that contains a script, or inline JavaScript. ACLs must include one or more roles, a security attribute, a condition, or a script. **Note:** If the *type* property is `graphql`, scripts aren't supported. The script can use the values of the `current` and `previous` global variables and system properties. The script must generate a true or false response in one of two ways: • return an `answer` variable set to a value of true or false • evaluate to true or false In either case, users only gain access to the object when the script evaluates to true and the user meets any conditions the ACL rule has. Both the conditions and the script must evaluate to true for a user to access the object. **Note:** If the evaluated item is in a related list, `current` points to the item the related list is on, not to the current item the ACL is for. However, If the item you are evaluating the ACL for is not in a related list, `current` points to the actual item. Format: • For functions, use the name of a function, function expression, or default function exported from a JavaScript module and import it into the `.now.ts` file. For information about JavaScript modules, see JavaScript modules and third-party libraries. • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide an inline script, use string literals or template literals for multiple lines of code: `'Script'` or `` `Script` ``.
description	String	A description of the object or permissions this ACL rule secures.
localOrExisting	String	The type of security attribute to apply. Valid values:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		• Local: A security attribute based on the `condition` property that is saved only for the ACL it is created in. • Existing: An existing security attribute to reference in the `security_attribute` property. Default: Local
decisionType	String	An option for whether the ACL should allow or deny access. Valid values: • allow: The ACL allows access. • deny: The ACL denies access. Default: allow
condition	String	A filter query that specifies the fields and values that must be true for users to access the object. For more information, see Operators available for filters and queries. ACLs must include one or more roles, a security attribute, a condition, or a script.
roles	Array	A list of variable identifiers of `Role` objects or sys_ids of roles that a user must have to access the object. For more information, see Role API - ServiceNow Fluent. ACLs must include one or more roles, a security attribute, a condition, or a script. **Note:** Users with the admin role always pass this permissions check because the admin role automatically grants users all other roles.
securityAttribute	String	Pre-defined conditions for the ACL to use. For example, whether a user is impersonating another user. For more information about security attributes, see OOB (Out-of-Box) Security Attributes. ACLs must include one or more roles, a security attribute, a condition, or a script. **Note:** For security attributes with the `Is localized` field set to true, the `localOrExisting` property of the ACL should be set to Local. If the `Is localized` field is false, the `localOrExisting` property should be set to Existing.
table	String	The name of the table to which the ACL applies.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		This property only applies and is required if the *type* property is one of the following values: ux_data_broker, ux_page, ux_route, pd_action, or record.
field	String	The name of a field on the table to secure. You can use the wildcard character (`"*"`) to select all fields.
name	String	The name of the ACL. This property only applies and is required if the *type* property is one of the following values: rest_endpoint, ui_page, processor, graphql, client_callable_flow_object, or client_callable_script_include.
protectionPolicy	String	A policy that determines whether someone can view or edit the script include after the application is installed on their instance. If undefined, other application developers can customize the script include. Valid values: • read: Allows anyone to read values from this downloaded or installed script include. No one can change script values on the instance on which they download or install the script include. • protected: Provides intellectual property protection for application developers. Customers who download the script include cannot see the contents of the script field. The script is encrypted in memory to prevent unauthorized users from seeing it in plain text.
$meta	Object	Metadata for the application metadata. With the *installMethod* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }` Valid values for *installMethod*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
import { Acl } from "@servicenow/sdk/core";

export default Acl({
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    $id: Now.ID['task_delete_acl'],
    active: true,
    adminOverrides: true,
    type: 'record',
    table: 'task',
    field: 'description',
    operation: 'delete',
    roles: [adminRole, managerRole],
})
```

The roles referenced are defined using the *Role* object:

```
import { Role } from "@servicenow/sdk/core";

const managerRole = Role({
   $id: Now.ID['manager_role'],
   name: 'x_snc_example.manager'
})

const adminRole = Role({
   $id: Now.ID['admin_role'],
   name: 'x_snc_example.admin',
   containsRoles: [managerRole]
})
```

Application Menu API - ServiceNow Fluent

The Application Menu API defines menus in the application navigator [sys_app_application].

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

For general information about application menus, see Create an application menu.

Related topics

ServiceNow Fluent

ApplicationMenu object

Create a menu for an application [sys_app_application].

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
title	String	Required. The label for the menu in the application navigator.
active	Boolean	Flag that indicates whether the menu appears in the application navigator. Valid values:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		• true: The menu appears. • false: The menu is hidden. Default: true
roles	Array	A list of variable identifiers of *Role* objects or names of roles that can access the menu. For more information, see Role API - ServiceNow Fluent.
category	Reference	The variable identifier of a menu category [sys_app_category] that defines the navigation menu style. To define a menu category, use the Record API - ServiceNow Fluent. For general information about menu categories, see Customize menu categories.
hint	String	A short description of the menu that displays as tooltip when hovering over it.
description	String	Additional information about what the application does.
name	String	An internal name to differentiate between applications with the same title.
order	Number	The relative position of the application menu in the application navigator. Default: 100
$meta	Object	Metadata for the application metadata. With the *installMethod* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }` Valid values for *installMethod*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
import { ApplicationMenu } from "@servicenow/sdk/core";

ApplicationMenu({
   $id: Now.ID['my_app_menu'],
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    title: 'My App Menu',
    hint: 'This is a hint',
    description: 'This is a description',
    category: appCategory,
    roles: ['admin'],
    active: true,
})
```

The category referenced is defined using the *Record* object:

```
import { Record } from "@servicenow/sdk/core";

export const appCategory = Record({
    table: 'sys_app_category',
    $id: Now.ID[9],
    data: {
        name: 'example',
        style: 'border-color: #a7cded; background-color: #e3f3ff;',
    },
})
```

Automated Test Framework Test API - ServiceNow Fluent

The Automated Test Framework Test API defines automated tests [sys_atf_test] that you can run to confirm that your instance works after making a change.

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

For general information about Automated Test Framework tests, see Automated Test Framework (ATF).

Related topics

ServiceNow Fluent

Test object

Create an automated test [sys_atf_test] containing a series of steps to execute.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
name	String	A unique name for the test.
description	String	A description of what the test does.
active	Boolean	Flag that indicates whether the test is active. Valid values: • true: The test is active. • false: The test isn't active.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Default: true
failOnServerError	Boolean	Flag that indicates whether to fail when a server error occurs during the test. Valid values: • true: The test fails when a server error occurs. • false: The test doesn't fail when a server error occurs. Default: true
configurationFunction	Function	The steps of the test. Test steps are passed as statements within the *atf* function. For example: `(atf) => { atf.form.openNewForm({ table: 'sn_example_table', formUI: 'standard_ui', view: '', }) }` For more information about test steps, see Supported test steps.
$meta	Object	Metadata for the application metadata. With the *installMethod* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }` Valid values for *installMethod*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

The output values of test steps with output variables can be saved as variables and used as inputs to other steps using the syntax `output.<output-variable>`. The output variables can be used both directly as inputs on appropriate fields or inside of a template string, such as with `atf.server.log` in the following example.

```
import { Test } from '@servicenow/sdk/core'

Test({
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
        active: true,
        failOnServerError: true,
        name: 'Simple example',
        description: 'An illustrative test written in fluent',
        $id: Now.ID[1],
    },
    (atf) => {
        atf.form.openNewForm({
            table: 'sn_table_app_reptile_table',
            formUI: 'standard_ui',
            view: '',
        })
        atf.form.setFieldValue({
            table: 'sn_table_app_reptile_table',
            formUI: 'standard_ui',
            fieldValues: {
                reptiles: 'lizard' as any,
            },
        })
        const output =
atf.form.submitForm({ assertType: 'form_submitted_to_server',
formUI: 'standard_ui' })
        atf.server.recordValidation({
            recordId: output.record_id,
            table: 'sn_table_app_reptile_table',
            assertType: 'record_validated',
            enforceSecurity: true,
            fieldValues: 'reptiles=lizard',
        })
        atf.server.log({
            log: `Submitted record with sys_id:
${output.record_id} to table ${output.table}`
        })
    }
)
```

Supported test steps

The following test steps are supported. For information about step properties, see the Automated Test Framework (ATF) test step categories documentation.

Note: Some fields available for test steps on forms aren't available as properties in ServiceNow Fluent.

Test steps

Category	Steps
Application Navigator category	• atf.applicationNavigator.applicationMenuVisibility • atf.applicationNavigator.moduleVisibility • atf.applicationNavigator.navigateToModule
Email category	• atf.email.generateInboundEmail • atf.email.generateInboundReplyEmail

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Test steps (continued)

Category	Steps
	• atf.email.generateRandomString • atf.email.validateOutboundEmail • atf.email.validateOutboundEmailGeneratedByFlow • atf.email.validateOutboundEmailGeneratedByNotification
Form category	• atf.form.addAttachmentsToForm • atf.form.clickDeclarativeAction • atf.form.clickModalButton • atf.form.clickUIAction • atf.form.declarativeActionVisibility • atf.form.fieldStateValidation • atf.form.fieldValueValidation • atf.form.openExistingRecord • atf.form.openNewForm • atf.form.setFieldValue • atf.form.submitForm • atf.form.uiActionVisibility
Forms in Service Portal category	• atf.form_SP.addAttachmentsToForm • atf.form_SP.clickUIAction_SP • atf.form_SP.fieldStateValidation_SP • atf.form_SP.fieldValueValidation_SP • atf.form_SP.openForm_SP • atf.form_SP.openServicePortalPage • atf.form_SP.setFieldValue_SP • atf.form_SP.submitForm_SP • atf.form_SP.uiActionVisibilityValidation_SP
Quick start tests for Dashboards	• atf.reporting.responsiveDashboard • atf.reporting.responsiveDashboardSharing
REST category	• atf.rest.assertJsonResponsePayloadElement • atf.rest.assertResponseHeader • atf.rest.assertResponseJSONPayloadIsValid • atf.rest.assertResponsePayload

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Test steps (continued)

Category	Steps
	• atf.rest.assertResponseTime • atf.rest.assertResponseXMLPayloadIsWellFormed • atf.rest.assertStatusCode • atf.rest.assertStatusCodeName • atf.rest.assertXMLResponsePayloadElement • atf.rest.sendRestRequest
Server category	• atf.server.addAttachmentsToExistingRecord • atf.server.checkoutShoppingCart • atf.server.createUser • atf.server.impersonate • atf.server.log • atf.server.recordDelete • atf.server.recordInsert • atf.server.recordQuery • atf.server.recordUpdate • atf.server.recordValidation • atf.server.replayRequestItem • atf.server.runServerSideScript • atf.server.searchForCatalogItem • atf.server.setOutputVariables
Service Catalog category	• atf.catalog.addItemToShoppingCart • atf.catalog.openCatalogItem • atf.catalog.openRecordProducer • atf.catalog.orderCatalogItem • atf.catalog.setCatalogItemQuantity • atf.catalog.setVariableValue • atf.catalog.submitRecordProducer • atf.catalog.validatePriceAndRecurringPrice • atf.catalog.variableStateValidation • atf.catalog.validateVariableValue
Service Catalog in Service Portal category	• atf.catalog_SP.addItemtoShoppingCart_SP • atf.catalog_SP.addOrderGuidetoShoppingCart_SP • atf.catalog_SP.addRowToMultiRowVariableSet_SP

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Test steps (continued)

Category	Steps
	• atf.catalog_SP.navigatewithinOrderGuide_SP • atf.catalog_SP.openCatalogItem_SP • atf.catalog_SP.openOrderGuide_SP • atf.catalog_SP.openRecordProducer_SP • atf.catalog_SP.orderCatalogItem_SP • atf.catalog_SP.reviewIteminOrderGuide_SP • atf.catalog_SP.reviewOrderGuideSummary_SP • atf.catalog_SP.saveCurrentRowOfMultiRowVariableSet_SP • atf.catalog_SP.setCatalogItemQuantity_SP • atf.catalog_SP.setVariableValue_SP • atf.catalog_SP.submitOrderGuide_SP • atf.catalog_SP.submitRecordProducer_SP • atf.catalog_SP.validateOrderGuideItem_SP • atf.catalog_SP.validatePriceAndRecurringPrice_SP • atf.catalog_SP.variableStateValidation_SP • atf.catalog_SP.validateVariableValue_SP
UI category	atf.uiTestScript.runTest

Business Rule API - ServiceNow Fluent

The Business Rule API defines server-sides scripts [sys_script] that run when a record is displayed, inserted, updated, or deleted, or when a table is queried.

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

For general information about business rules, see Classic Business rules.

Related topics

ServiceNow Fluent

BusinessRule object

Create a business rule [sys_script] that automatically changes values in form fields when certain server-side conditions are met.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Format: `Now.ID['String' or Number]`
name	String	Required. A name for the business rule.
table	String	Required. The name of the table on which the business rule runs.
script	Script	A custom script runs when the defined conditions are true. This property supports a function from a JavaScript module, a reference to another file in the application that contains a script, or inline JavaScript. Format: • For functions, use the name of a function, function expression, or default function exported from a JavaScript module and import it into the `.now.ts` file. For information about JavaScript modules, see JavaScript modules and third-party libraries. • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide an inline script, use string literals or template literals for multiple lines of code: `'Script' or `Script``.
order	Number	A number indicating the sequence in which the business rule should run. If there are multiple rules on a particular activity, the rules run in the order specified, from lowest to highest. Default: 100
active	Boolean	Flag that indicates whether the business rule is enabled. Valid values: • true: The business rule is enforced. • false: The business rule isn't enforced. Default: true
when	String	The time that the business rule should execute. For more information about when business rules run, see Classic Business rules. Valid values: • before • after • async • display

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Default: before
action	Array	The record options that the business rule applies to. For more information about business rule actions, see Classic Business rules. Valid values: • insert • update • delete • query
addMessage	Boolean	Flag that indicates whether to add a message that appears when the business rule runs. Valid values: • true: Displays a message. • false: Doesn't display a message. Default: false
abortAction	Boolean	Flag that indicates whether to abort the current database transaction. For example, on a before insert business rule, if the conditions are met, don't insert the record into the database. Valid values: • false: Doesn't abort the current database transaction. • true: Aborts the current database transaction. You can't perform additional actions on the record, such as setting field values and running scripts. You can still display a message to users with the *addMessage* and *message* properties. Default: false
message	String	A message that appears when the business rule runs.
roleConditions	Array	A list of variable identifiers of *Role* objects that users who are modifying records in the table must have for the business rule to run. For more information, see Role API - ServiceNow Fluent.
condition	String	A JavaScript conditional statement that specifies the fields and values that must be true for the script to run. This property supports inline JavaScript or a reference to another file in the application that contains a script. **Note:** Don't use this property if you include the condition statement with the *filterCondition* or *script* properties.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		By specifying the condition statement with this property, the condition is evaluated separately, and the business rule runs only if the condition is true. To have the condition statement reevaluated a second time before running an async business rule, set the *glide.businessrule.async_condition_check* system property to true. Format: • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide an inline script, use string literals or template literals for multiple lines of code: `'Script' or `Script``.
filterCondition	String	A filter query that specifies the fields and values that must be true for the business rule to run. For more information, see Operators available for filters and queries.
setFieldValue	String	The values to set for fields in the table. This can be provided as an encoded query, such as `setFieldValue: 'sec_created=false^EQ'`.
description	String	A description of what the business rule does.
protectionPolicy	String	A policy that determines whether someone can view or edit the script include after the application is installed on their instance. If undefined, other application developers can customize the script include. Valid values: • read: Allows anyone to read values from this downloaded or installed script include. No one can change script values on the instance on which they download or install the script include. • protected: Provides intellectual property protection for application developers. Customers who download the script include cannot see the contents of the script field. The script is encrypted in memory to prevent unauthorized users from seeing it in plain text.
$meta	Object	Metadata for the application metadata. With the *installMethod* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Valid values for *installMethod*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
import { BusinessRule } from '@servicenow/sdk/core'
import { FunctionExport, FunctionExpression } from '../server/scripts.js'
import DefaultExportFunction from '../server/scripts.js'

const BR1 = BusinessRule({
    name: 'exportedFunction',
    table: 'x_snc_table',
    when: 'after',
    action: ['update', 'delete', 'insert'],
    script: FunctionExport,
    order: 100,
    active: true,
    addMessage: false,
    message: '<p>message</p>',
    abortAction: false,
    $id: Now.ID[0],
})

const BR2 = BusinessRule({
    name: 'businessrule2',
    table: 'x_snc_table',
    script: FunctionExpression,
    when: 'after',
    action: ['update'],
    $id: Now.ID[1],
})

const BR3 = BusinessRule({
    name: 'businessrule3',
    table: 'x_snc_table',
    script: DefaultExportFunction,
    when: 'after',
    action: ['update'],
    filterCondition:
 `sys_updated_onSTARTSWITHb^sys_updated_bySTARTSWITHm^EQ
    <item goto="false" or="false" field="sys_updated_on" endquery="false"
 value="b" operator="STARTSWITH" newquery="false"/>
    <item goto="false" or="false" field="sys_updated_by" endquery="false"
 value="m" operator="STARTSWITH" newquery="false"/>
    <item goto="false" or="false" field="" endquery="true" value=""
 operator="=" newquery="false"/>`,
    $id: Now.ID[2],
})
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
const BR4 = BusinessRule({
    name: 'templateBR',
    action: ['insert'],
    when: 'after',
    table: 'x_snc_table',
    roleConditions: [admin],
    order: 100,
    active: true,
    addMessage: true,
    message: '<p>message</p>',
    script: `gs.info('info')`,
    abortAction: false,
    $id: Now.ID[3],
})
```

Client Script API - ServiceNow Fluent

The Client Script API defines client-side scripts [sys_script_client] that run JavaScript on the client (web browser) when client-based events occur, such as when a form loads, after form submission, or when a field changes value.

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

For general information about client scripts, see Client scripts.

Related topics

ServiceNow Fluent

ClientScript object

Create a client script [sys_script_client] to configure forms, form fields, and field values while the user is using the form.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
table	String	Required. The name of the table on which the client script runs.
name	String	Required. The name of the client script.
active	Boolean	Flag that indicates whether the client script is enabled. Valid values: • true: The script is enabled. • false: The script isn't enabled. Default: true

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
appliesExtended	Boolean	Flag that indicates whether the client script applies to tables extended from the specified table. Valid values: • true: The script applies to extended tables. • false: The script doesn't apply to extended tables. Default: false
uiType	String	The user interface to which the client script applies. Valid values: • desktop • mobile_or_service_portal • all Default: desktop
description	String	A description of the functionality and purpose of the client script.
messages	String	Text strings that are available to the client script as localized messages using `getmessage('[message]')`. For more information, see Translate a client script message.
isolateScript	Boolean	Flag that indicates whether the script runs in strict mode, with access to direct DOM, jQuery, prototype, and the window object turned off. Valid values: • true: Isolate the script and don't run it in strict mode. • false: Run the script in strict mode. Default: false
script	Script	A client-side script that runs in the browser. This property supports inline JavaScript or a reference to another file in the application that contains a script. Format: • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide an inline script, use string literals or template literals for multiple lines of code: `'Script'` or `` `Script` ``.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
global	Boolean	Flag that indicates on which views of the table the client script runs. Valid values: • true: The script runs on all views. • false: The script runs only on the specified views. Default: true
view	String	The views of the table on which the client script runs. This property applies only when the *global* property is set to false.
type	String	The type of client script, which defines when it runs. For more information about the supported types, see Client scripts. Valid values: • onCellEdit: Runs when the list editor changes a cell value. • onChange: Runs when a particular field value changes on the form. • onLoad: Runs when the system first renders the form and before users can enter data. Typically, *onLoad()* client scripts perform client-side-manipulation of the current form or set default record values. • onSubmit: Runs when a form is submitted. Typically, *onSubmit()* scripts validate things on the form and ensure that the submission makes sense. An *onSubmit()* client script can cancel form submission by returning a value of false.
field	String	A field on the table that the client script applies to. This property applies only when the *type* property is set to onChange or onCellEdit.
$meta	Object	Metadata for the application metadata. With the *installMethod* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }` Valid values for *installMethod*:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		• demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
import { ClientScript } from '@servicenow/sdk/core'

export const cs = ClientScript({
    $id: Now.ID['my_scripts'],
    name: 'my_scripts',
    table: 'incident',
    active: true,
    appliesExtended: false,
    global: true,
    uiType: 'all',
    messages: '',
    isolateScript: false,
    type: 'onLoad',
    script: Now.include('../client/client-script.js'),
})
```

The client script is defined in the `client-script.js` file referenced from the *`script`* property. For example:

```
function onLoad() {
     const x = 'util' g_form.addInfoMessage(x)
}
```

Cross-Scope Privilege API - ServiceNow Fluent

The Cross-Scope Privilege API defines cross-scope privileges [sys_scope_privilege] for runtime access tracking.

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

Runtime access tracking allows administrators to manage script access to application resources by creating a list of script operations and targets that the system authorizes to run. For general information about cross-scope privileges, see Cross-scope privilege record.

Related topics

ServiceNow Fluent

CrossScopePrivilege object

Configure cross-scope privileges [sys_scope_privilege] that determine which script operations and targets the system allows to run in the application.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
status	String	Required. The authorization for this record. Valid values: • requested • allowed • denied
operation	String	Required. The operation that the script performs on the target. The target type determines the available operations. Tables [sys_db_object] support the read, write, create, and delete operations. Script includes [sys_script_include] and script objects [sys_db_object] only support the execute operation. Valid values: • create • delete • read • write • execute
targetName	String	Required. The name of the table, script include, or script object being requested.
targetScope	String	Required. The application scope from which resources are requested.
targetType	String	Required. The type of request: script include, script object, or table. Valid values: • sys_script_include • scriptable • sys_db_object
$meta	Object	Metadata for the application metadata. With the *`installMethod`* property, you can map the application metadata to an output directory that loads only in specific circumstances.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		`$meta: { installMethod: 'String' }` Valid values for *`installMethod`*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
import { CrossScopePrivilege } from '@servicenow/sdk/core'

CrossScopePrivilege({
   $id: Now.ID['cross_1'],
   status: 'allowed',
   operation: 'execute',
   targetName: 'Script type',
   targetScope: 'x_snc_example',
   targetType: 'scriptable',
})
```

Dashboard API - ServiceNow Fluent

The Dashboard API defines dashboards [par_dashboard] for organizing and sharing data visually.

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

A dashboard consists of tabs, widgets, visibilities, and permissions. Each tab contains widgets that display data visualizations, headings, rich text, and other components.

Dashboards can be used as the home page of a workspace by referring to one or more workspaces from the *`visibilities`* array of the *`Dashboard`* object. To create a workspace, see Workspace API - ServiceNow Fluent.

For general information about dashboards, see Dashboards in Platform Analytics.

Related topics

ServiceNow Fluent

Dashboard object

Create a shareable dashboard [par_dashboard] with data visualizations, filters, tabs, widgets, permissions, and visibility rules.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
name	String	Required. A name to display for the dashboard.
active	Boolean	Flag that indicates whether the dashboard is active. Default: true
tabs	Array	A list of tabs to display in the dashboard. For more information, see tabs array.
permissions	Array	A list of user permissions required to access the dashboard. For more information, see permissions array.
visibilities	Array	A list of visibility rules that control which UX experiences display the dashboard. For more information, see visibilities array. Default: A default visibility rule with sys_id 08c73d60537101100834ddeeff7b1287 is used.
$meta	Object	Metadata for the application metadata. With the *`installMethod`* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }` Valid values for *`installMethod`*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
import { Dashboard } from "@servicenow/sdk/core";

Dashboard({
  $id: Now.ID["incident_management_dashboard"],
  name: "Incident Management Dashboard",
  tabs: [
    {
      $id: Now.ID["incident_overview_tab"],
      name: "Overview",
      widgets: [
        // Single Score: Total Open Incidents
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
        {
          $id: Now.ID["total_open_incidents_widget"],
          component: "single-score",
          componentProps: {
            dataSources: [
              {
                label: "Incident",
                sourceType: "table",
                tableOrViewName: "incident",
                filterQuery: "",
                reportSourceSysId: "d94d6f62d7632100b96d45a3ce6103d2",
                id: "dGFibGVpbmNpZGVudDE3NjMzOTI0OTY5ODE="
              }
            ],
            headerTitle: "Open Incidents",
            metrics: [
              {
                dataSource: "dGFibGVpbmNpZGVudDE3NjMzOTI0OTY5ODE=",

 id: "ZEdGaWJHVnBibU5wWkdWdWRERTNOak16T1RJME9UWTVPREU9MTc2MzM5MjQ5Nzc5OQ==",
                aggregateFunction: "COUNT",
                axisId: "primary"
              }
            ],
            sortBy: "value"
          },
          height: 14,
          width: 14,
          position: { x: 0, y: 0 }
        },
        // Vertical Bar: Incidents by Priority
        {
          $id: Now.ID["incidents_by_priority_widget"],
          component: "vertical-bar",
          componentProps: {
            dataSources: [
              {
                label: "Incident",
                sourceType: "table",
                tableOrViewName: "incident",
                filterQuery: "",
                reportSourceSysId: "d94d6f62d7632100b96d45a3ce6103d2",
                id: "dGFibGVpbmNpZGVudDE3NjI3OTU4OTg3NDA="
              }
            ],
            headerTitle: "Incidents by Priority",
            metrics: [
              {
                dataSource: "dGFibGVpbmNpZGVudDE3NjI3OTU4OTg3NDA=",

 id: "ZEdGaWJHVnBibU5wWkdWdWRERTNOakkzT1RVNE9UZzNOREE9MTc2Mjc5NTg5OTM5OA==",
                aggregateFunction: "COUNT",
                axisId: "primary"
              }
            ],
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
            groupBy: [
              {
                groupBy: [
                  {

dataSource: "dGFibGVpbmNpZGVudDE3NjI3OTU4OTg3NDA=",
                    groupByField: "priority"
                  }
                ],
                maxNumberOfGroups: 10,

numberOfGroupsBasedOn: "NO_OF_GROUP_BASED_ON_PER_METRIC",
                showOthers: false,
                disableRanges: false
              }
            ],
            sortBy: "value"
          },
          height: 14,
          width: 17,
          position: { x: 14, y: 0 }
        },
        // Vertical Bar: Incidents by State
        {
          $id: Now.ID["incidents_by_state_widget"],
          component: "vertical-bar",
          componentProps: {
            dataSources: [
              {
                label: "Incident",
                sourceType: "table",
                tableOrViewName: "incident",
                filterQuery: "",
                reportSourceSysId: "d94d6f62d7632100b96d45a3ce6103d2",
                id: "dGFibGVpbmNpZGVudDE3NjI3OTU4OTg3NDA="
              }
            ],
            headerTitle: "Incidents by State",
            metrics: [
              {
                dataSource: "dGFibGVpbmNpZGVudDE3NjI3OTU4OTg3NDA=",

id: "ZEdGaWJHVnBibU5wWkdWdWRERTNOakkzT1RVNE9UZzNOREE9MTc2Mjc5NTg5OTM5OA==",
                aggregateFunction: "COUNT",
                axisId: "primary"
              }
            ],
            groupBy: [
              {
                groupBy: [
                  {

dataSource: "dGFibGVpbmNpZGVudDE3NjI3OTU4OTg3NDA=",
                    groupByField: "state"
                  }
                ],
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
                maxNumberOfGroups: 10,

 numberOfGroupsBasedOn: "NO_OF_GROUP_BASED_ON_PER_METRIC",
                showOthers: false,
                disableRanges: false
              }
            ],
            sortBy: "value"
          },
          height: 14,
          width: 17,
          position: { x: 31, y: 0 }
        }
      ]
    },
    {
      $id: Now.ID["incident_trends_tab"],
      name: "Trends",
      widgets: [] // Can be populated later or left empty
    }
  ],
  visibilities: [
    {
      $id: Now.ID["incident_dashboard_visibility"],
      experience: incidentWorkspace
    }
  ],
  permissions: []
})
```

The workspace referenced is defined using the *Workspace* object:

```
import { Workspace } from "@servicenow/sdk/core";

export const myWorkspace = Workspace({
  $id: Now.ID["my_workspace"],
  title: "My Workspace",
  path: "my-workspace",
  tables: ["incident"],
  listConfig: myListConfig
})
```

tabs array

Create tabs [par_dashboard_tab] that contain widgets for a dashboard.

Within a dashboard, tabs are ordered using their position in the array.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
name	String	Required. A name to display on the tab.
active	Boolean	Flag that indicates whether the tab is active. Default: true **Note:** This flag is not used in the application and its value does not change the status of the tab.
widgets	Array	A list of widgets to display in the tab. For more information, see widgets array.

```
tabs: [
    {
      $id: Now.ID["my_dashboard_tab1"],
      name: "Overview",
      widgets: [
        {
          $id: Now.ID["my_dashboard_widget_1"],
          component: "single-score",
          componentProps: {
            dataSources: [
              {
                label: "Incident",
                sourceType: "table",
                tableOrViewName: "incident",
                filterQuery: "",
                reportSourceSysId: "d94d6f62d7632100b96d45a3ce6103d2",
                id: "dGFibGVpbmNpZGVudDE3NjMzOTI0OTY5ODE="
              }
            ],
            headerTitle: "Open Incidents",
            metrics: [
              {
                dataSource: "dGFibGVpbmNpZGVudDE3NjMzOTI0OTY5ODE=",

 id: "ZEdGaWJHVnBibU5wWkdWdWRERTNOak16T1RJME9UWTVPREU9MTc2MzM5MjQ5Nzc5OQ==",
                aggregateFunction: "COUNT",
                axisId: "primary"
              }
            ],
            sortBy: "value"
          },
          height: 14,
          width: 14,
          position: { x: 0, y: 0 }
        }
      ]
    }
]
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

widgets array

Create widgets [par_dashboard_widget] within a tab in a grid layout.

Dashboards use a 48-point grid system for positioning widgets.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
component	Reference or String	Required. The name of a component (such as line or single-score) or the sys_id of a component from the UX Macroponent Definition [sys_ux_macroponent] table to display as a widget. Component names aren't case sensitive and resolve to their sys_ids during the build process.
height	Number	Required. The height of the widget in grid units.
width	Number	Required. The width of the widget in grid units. Maximum value: 48
position	Object	Required. The position of the widget in the grid layout with `x` and `y` properties. For example, a value of `{ x: 0, y: 0 }` positions the widget in the top-left corner of the grid.
componentProps	Object	The property configuration of a component. For more information, see componentProps object.

```
widgets: [
                {
                    $id: Now.ID['incident-count-chart'],
                    component: 'vertical-bar',  // Vertical bar chart
                    componentProps: {
                        title: 'Incident Count',
                        dataSource: 'incident',
                        aggregation: 'count'
                    },
                    height: 8,
                    width: 6,
                    position: { x: 0, y: 0 },
                },
                {
                    $id: Now.ID['recent-incidents-list'],
                    component: 'list',  // List component
                    componentProps: {
                        table: 'incident',
                        filter: 'active=true',
                        limit: 10,
                        columns:
 ['number', 'short_description', 'priority', 'state']
                    },
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
                    height: 8,
                    width: 6,
                    position: { x: 6, y: 0 },
                }
            ],
        },
        {
            $id: Now.ID['analytics'],
            name: 'Analytics',
            widgets: [
                {
                    $id: Now.ID['metrics-widget'],
                    component: 'single-score', // Single score component
                    componentProps: {
                        title: 'Key Metrics',
                        scoreSize: 'large'
                    },
                    height: 12,
                    width: 12,
                    position: { x: 0, y: 0 },
                }
]
```

componentProps object

Configure the properties of widgets [par_dashboard_widget].

The available properties are determined by the component specified with the *component* property of the *Dashboard* object.

- Trend data components require the *dataSources*, *metrics*, and *trendBy* properties. The *groupBy* property is optional.
- Group data components require the *dataSources*, *metrics*, and *groupBy* properties. The *trendBy* property isn't supported in these visualizations.
- Simple data components require the *dataSources* and *metrics* properties. The *groupBy* and *trendBy* properties aren't supported in these visualizations.

Properties

Name	Type	Description
dataSources	Array	A list of data sources for the component. For example: `dataSources: [ { label: "Incident", // Human-readable label sourceType: "table", // Type of data source tableOrViewName: "incident", // ServiceNow table name filterQuery: "", // Optional encoded query filter id: "data_source_1" // Unique data source ID }, ... ]`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
headerTitle	String	A title to display with the widget.
metrics	Array	A list of metrics to measure for the data source. For example: `metrics: [{ dataSource: 'data_source_1', // Must match dataSource id id: 'metric_1', // Unique metric ID aggregateFunction: 'AVG', // COUNT, SUM, AVG, MIN, MAX, COUNT_DISTINCT aggregateField: 'business_duration' // Field to be used for aggregation. axisId: 'primary', // Which axis to display the series }, ...]`
groupBy	Array	A list of configurations for grouping and organizing data by data source. For example: `groupBy: [ { groupBy: [ { dataSource: "data_source_1", // Must match dataSource id groupByField: "state" // Field to group by } ], maxNumberOfGroups: 10, // Maximum categories to show showOthers: false // Show "Others" category }, ... ]`
trendBy	Object	A configuration for trend charts. For example: `trendBy: { "trendByFrequency": "year", // Frequency of the trend (date, week, month, year) "trendByFields": [ { "field": "sys_created_on", // Field to trend on (from the table of the dataSource) "metric": "metric_1" // ID of the metric } ] }`
sortBy	String	The method of sorting data.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Valid values: • value • label • field

In the following example, the `line` component, which is a trend data type visualization, is used to show how a metric changes over time.

```
{
    component: 'line',
    componentProps: {
        dataSources: [
            {
                label: 'Incident',
                sourceType: 'table',
                tableOrViewName: 'incident',
                filterQuery: '',
                id: 'data_source_1',
            },
        ],
        headerTitle: 'Incidents by State',
        metrics: [
            {
                dataSource: 'data_source_1',
                id: 'metric_1',
                aggregateFunction: 'COUNT',
                axisId: 'primary',
            },
        ],
        trendBy: {
            trendByFrequency: "year",
            trendByFields: [
                {
                    field: "sys_created_on",
                    metric: "metric_1"
                }
            ]
        },
    },
    height: 14,
    width: 17,
    position: { x: 0, y: 0 },
}
```

In the following example, the `single-score` component, which is a simple data type visualization, is used to show a single count metric.

```
{
    component: 'single-score',
    componentProps: {
        dataSources: [
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
            {
                label: 'Incident',
                sourceType: 'table',
                tableOrViewName: 'incident',
                filterQuery: '',
                id: 'data_source_1',
            },
        ],
        headerTitle: 'Open Incidents',
        metrics: [
            {
                dataSource: 'data_source_1',
                id: 'metric_1',
                aggregateFunction: 'COUNT',
                axisId: 'primary',
            },
        ]
    },
    height: 14,
    width: 14,
    position: { x: 0, y: 0 },
}
```

permissions array

Define permissions [par_dashboard_permission] to read, edit, and share a dashboard.

At least one of the *`user`*, *`group`*, or *`role`* properties must be specified for each permission in the array.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
user	Reference or String	The variable identifier or sys_id of a user [sys_user] to which to grant permissions. To define a user, use the Record API - ServiceNow Fluent.
group	Reference or String	The variable identifier or sys_id of a user group [sys_user_group] to which to grant permissions. To define a user, use the Record API - ServiceNow Fluent.
role	Reference or String	The variable identifier or sys_id of a role [sys_user_role] to which to grant permissions. To define a user, use the Role API - ServiceNow Fluent.
canRead	Boolean	Flag that indicates whether the user, group, or role can view the dashboard. Default: true
canWrite	Boolean	Flag that indicates whether the user, group, or role can edit the dashboard.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Default: false
canShare	Boolean	Flag that indicates whether the user, group, or role can share the dashboard. Default: false
owner	Boolean	Flag that indicates whether the user, group, or role is the owner of the dashboard. For at least one user, the *owner* property should be set to true. Default: false

```
permissions: [
        {
            $id: Now.ID['manager-user-permission'],
            user: 'a8f98bb0eb32010045e1a5115206fe3a', // sys_id of manager
user
            canRead: true,
            canWrite: true,
            owner: true,
        },
        {
            $id: Now.ID['itil-role-permission'],
            role: '2831a114c611228501d4ea6c309d626d', // sys_id of itil role
            canRead: true,
            canWrite: false,
        },
        {
            $id: Now.ID['support-group-permission'],
            group: 'd625dccec0a8016700a222a0f7900d06', // sys_id of Service
Desk group
            canRead: true,
            canWrite: false,
        }
]
```

visibilities array

Define visibility rules [par_dashboard_visibility] for which UX experiences display the dashboard.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
experience	Reference or String	Required. The variable identifier of a *Workspace* object or sys_id of a UX application [sys_ux_page_registry]. For more information, see Workspace API - ServiceNow Fluent.

```
visibilities: [
  {
    $id: Now.ID["dashboard_visibility_1"],
    experience: myWorkspace // Reference to Workspace object
  }
]
```

Email Notification API - ServiceNow Fluent

The Email Notification API defines notifications [sysevent_email_action] that send automated emails based on database operations, custom events, or manual triggers.

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

For general information about email notifications, see Email and SMS notifications.

Related topics

ServiceNow Fluent

EmailNotification object

Create an email notification [sysevent_email_action] specifying when to send it, who receives it, what it contains, and if it can be delivered in an email digest.

For general information about creating email notifications, see Create an email notification.

Properties

Name	Type	Description
table	Reference or String	Required. The variable identifier or name of a table to which the notification applies. To define a table, use the Table API - ServiceNow Fluent. **Note:** Don't select the Task [task] table, which is for extending other tables. Notifications that run on the Task table directly aren't supported.
triggerConditions	Object	Required. The conditions that trigger the notification. For more information, see triggerConditions object.
name	String	A unique name for the email notification.
description	String	A description of the purpose of the email notification.
category	Reference or String	The variable identifier or name of a notification category for grouping notifications. To define a notification category, use the Record API - ServiceNow Fluent.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Default: The default email category (c97d83137f4432005f58108c3ffa917a)
notificationType	String	The type of notification. Valid values: • email: A standard email. • vcalendar: A meeting invitation. Meeting invitations aren't supported with email digests. Default: email
active	Boolean	Flag that indicates whether the notification is active. Default: true
mandatory	Boolean	Flag that indicates whether the notification is required. Default: false
enableDynamicTranslation	Boolean	Flag that indicates whether to enable dynamic translation for the notification. Default: false
emailContent	Object	The email content and formatting. For more information, see emailContent object.
recipientDetails	Object	The email recipients. For more information, see recipientDetails object.
digest	Object	The email digest content and formatting. For more information, see digest object.
$meta	Object	Metadata for the application metadata. With the `installMethod` property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }` Valid values for `installMethod`: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
import { EmailNotification } from '@servicenow/sdk/core'
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
EmailNotification({
    table: 'incident',
    name: 'Custom Event Notification',
    description: 'Triggered by custom event',
    category: 'c97d83137f4432005f58108c3ffa917a',    // Default email category
sys_id
    triggerConditions: {
        generationType: 'event',
        eventName: 'custom.incident.escalated',
        order: 100
    },
    recipientDetails: {
        recipientUsers: ['6816f79cc0a8016401c5a33be04be441'],    // Admin
user sys_id
        eventParm1WithRecipient: true,    // Event param 1 contains recipient
        isSubscribableByAllUsers: true
    },
    emailContent: {
        contentType: 'text/html',
        subject: 'Incident Escalated',
        messageHtml: '<p>An incident has been escalated.</p>'
    }
})
```

triggerConditions object

Configure the conditions that must be met for the notification to be sent.

Note: If the same trigger generates multiple notifications, the system sends only one notification. The system considers all other notifications, even if they have a different subject and body, as duplicates. The Ignore Duplicates business rule controls this functionality.

Properties

Name	Type	Description
generationType	String	Required. The method of triggering the email notification. Valid values: • engine: Sends the notification when a record is inserted or updated. • event: Sends the notification when custom events occur. • triggered: Manually triggered notifications. The following properties don't apply if this property is set to `triggered`.
onRecordInsert	Boolean	Flag that indicates whether to send the notification when a record is inserted. **Note:** Either this property or the *`onRecordUpdate`* property must be true. Default: false
onRecordUpdate	Boolean	Flag that indicates whether to send the notification when a record is updated.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Note: Either this property or the *onRecordInsert* property must be true. Default: false
eventName	String	The name of a custom event to trigger sending the notification. This property is required if the value of the *generationType* property is `event`.
affectedFieldOnEvent	String	The event parameter that contains the affected field. This property applies only if the *generationType* property is set to `event`. Valid values: • parm1 • parm2
weight	Number	The notification priority relative to duplicate notifications. Notifications that have the same target table and recipients are considered duplicates if the weights are different. If weights are same, an additional evaluation is done to check if the subject and the body (excluding watermark) are the same to qualify as duplicate notification. When there are duplicate notifications, the system only sends the notification with the highest weight. All other notifications are moved from the Outbox to the Skipped mailbox. The default value of 0 causes the system to send the notification (assuming the conditions are met). Maximum value: 1000 Default: 0
condition	String	A filter query that specifies the fields and values that must be true for users to access the object. For more information, see Operators available for filters and queries.
advancedCondition	String	A JavaScript conditional statement that must return true or set a global answer variable to true to send the notification. For more information, see Advanced conditions for email notifications. **Note:** This property is evaluated in addition to other conditions that you set on the notification. Both the condition and advanced condition must evaluate to true to send the notification.
itemTable	Reference or String	The variable identifier or name of the table to which the notification item refers. To define a table, use the Table API - ServiceNow Fluent.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
item	String	The item to use for the notification context.
order	Number	The execution order of the notification. Maximum value: 9999 Default: 100

```
triggerConditions: {
    generationType: "engine",
    onRecordInsert: false,
    onRecordUpdate: true,
    weight: 100,
    condition: "priority=1^ORpriority=2^state!=6^state!=7", // High/Critical priority,
 not resolved/closed
    order: 100
}
```

emailContent object

Configure the contents of an email notification.

Properties

Name	Type	Description
contentType	String	The type of email content. Valid values: • text/html • text/plain • multipart/mixed Default: text/html
template	Reference or String	The variable identifier or sys_id of an email template [sysevent_email_template]. To define an email template, use the Record API - ServiceNow Fluent. You can only specify an email template that meets one of the following conditions: • The template has the same scope and table as the notification. • The template has the same scope but has no specified table. • The template has the same table and is in the global scope.
style	Reference or String	The variable identifier or sys_id of an email style [sys_email_style]. To define an email style, use the Record API - ServiceNow Fluent.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
subject	String	A subject line for the email. If empty, the system uses the Subject value from the email template. You can use the `\${variable}` format for variable references. Variables map to column names available from the notification table, its parent tables, and reference tables. Use variables to include values from a record in the table, such as the short description or comments and work notes.
smsAlternate	String	A notification message to send specifically to SMS devices. The SMS alternate message is limited to 140 characters. If empty, the system uses the SMS alternate value from the email template. You can use the `\${variable}` format for variable references. Variables map to column names available from the notification table, its parent tables, and reference tables. Use variables to include values from a record in the table, such as the short description or comments and work notes.
importance	String	The level of importance of the message. Valid values: • low • high
includeAttachments	Boolean	Flag that indicates whether to include attachments from the notification trigger record with the email. Default: false
omitWatermark	Boolean	Flag that indicates whether to omit the watermark attached to each email. For more information, see Watermarks on notification emails. Default: false
from	String	The email address from which to send the notification.
replyTo	String	The email address to which recipients can reply.
pushMessageOnly	Boolean	Flag that indicates whether to send the notification only as a push notification to a mobile device. Default: false
pushMessageList	Array	A list of variable identifiers or sys_ids of push messages [sys_push_notif_msg] to associate with the notification. To define a push notification, use the Record API - ServiceNow Fluent. **Note:** The push message and notification must be for the same table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
forceDelivery	Boolean	Flag that indicates whether to bypass user notification preferences that would prevent a notification and send the notification anyway. Default: false
messageHtml	String	The HTML content of the notification message. If empty, the system uses the message HTML from the email template. You can use the `\${variable}` format for variable references. Variables map to column names available from the notification table, its parent tables, and reference tables. Use variables to include values from a record in the table, such as the short description or comments and work notes. This property is required if the value of the *contentType* property is `text/html` or `multipart/mixed`.
messageText	String	The plain text content of the notification message. If empty, the system uses the message text from the email template. You can use the `\${variable}` format for variable references. Variables map to column names available from the notification table, its parent tables, and reference tables. Use variables to include values from a record in the table, such as the short description or comments and work notes. This property is required if the value of the *contentType* property is `text/plain` or `multipart/mixed`.
message	String	Deprecated. Message contents.

```
emailContent: {
    contentType: "text/html",
    subject: "CRITICAL: Incident \${number} - \${short_description}",
    messageHtml: `
            <div style="background-color: #ff4444; color: white; padding: 10px;
 border-radius: 5px;">
                <h2>CRITICAL INCIDENT ALERT</h2>
                <p><strong>Incident:</strong> \${number}</p>
                <p><strong>Priority:</strong> \${priority}</p>
                <p><strong>Description:</strong>
 \${short_description}</p>
                <p><strong>Assigned To:</strong>
 \${assigned_to.name}</p>
                <p><strong>Created:</strong>
 \${sys_created_on}</p>
                <p><a href="\${instance_url}/incident.do?sys_id=\${sys_id}"
 style="color: #ffffff; text-decoration: underline;">View Incident</a></p>
            </div>
        `,
    smsAlternate:
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
        "CRITICAL: Incident \${number} - \${short_description}. Priority: \${priority}.
Assigned: \${assigned_to.name}",
    pushMessageList: ["mobile_push_notification_sys_id"],
    forceDelivery: true,
    importance: "high"
}
```

digest object

Configure the contents of an email digest that summarizes the activity for a selected notification and its target records during a specified time interval.

For general information about email digests, see Email digests.

Properties

Name	Type	Description
allow	Boolean	Flag that indicates whether users to receive this notification as a digest. If false, all other properties of the *digest* object are ignored. Default: false
default	Boolean	Flag that indicates whether digest mode is enabled by default for this notification. Default: false
type	String	The type of digest. Valid values: • single: Sends the notification in an email digest when it's triggered multiple times during the selected interval for a single target record, for example INC001 only. • multiple: Sends the notification in an email digest when it's triggered multiple times during the selected interval for a multiple target record, for example INC001, INC002, and more.
defaultInterval	Reference or String	The variable identifier or sys_id of a digest interval [sys_email_digest_interval] to use as the default time interval for digest delivery. To define a digest interval, use the Record API - ServiceNow Fluent.
subject	String	A subject line for the email digest. If empty, the system uses the Subject value from the email template. You can use the `\${variable}` format for variable references. Variables map to column names available from the notification table, its parent tables, and reference tables. Use variables to include values from a record in the table, such as the short description or comments and work notes.
html	String	The HTML content of the email digest. You can use the `\${variable}` format for variable references. Variables map to column names available from the

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		notification table, its parent tables, and reference tables. Use variables to include values from a record in the table, such as the short description or comments and work notes.
text	String	The plain text content of the email digest. You can use the `\${variable}` format for variable references. Variables map to column names available from the notification table, its parent tables, and reference tables. Use variables to include values from a record in the table, such as the short description or comments and work notes.
template	String	The variable identifier or sys_id of an email template [sysevent_email_template]. To define an email template, use the Record API - ServiceNow Fluent.
separatorHtml	String	An HTML separator that appears between each item summarized in the digest. You can use the `\${variable}` format for variable references. Variables map to column names available from the notification table, its parent tables, and reference tables. Use variables to include values from a record in the table, such as the short description or comments and work notes. Default: `<p> </p>\n<hr>\n<p> </p>`
separatorText	String	A plain text separator that appears between each item summarized in the digest. You can use the `\${variable}` format for variable references. Variables map to column names available from the notification table, its parent tables, and reference tables. Use variables to include values from a record in the table, such as the short description or comments and work notes. Default: `----------\n`
from	String	The email address from which to send the notification.
replyTo	String	The email address to which recipients can reply.

```
digest: {
    allow: true,
    defaultInterval: hourlyDigest,
    template: escalationTemplate,
    subject: "Incident Escalation Digest - \${digest_count} incidents require attention",
    html: `<div>
           <div>Incident Escalation Digest</div>
           </div>`,
    separatorHtml: '<hr style="margin: 15px 0;">',
    from: "noreply@company.com",
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    replyTo: "itsupport@company.com"
}
```

recipientDetails object

Configure who receives an email notification.

Notification recipients must be active users defined as active users in the User [sys_user] table. They must also have a valid email address defined for their primary channel (device) in the Notification Device [cmn_notif_device] table. Your notification recipients must also have the appropriate notification preferences enabled.

Properties

Name	Type	Description
recipientUsers	Array	A list of variable identifiers or sys_ids of users [sys_user] or a list of email addresses to receive the notification. To define a user, use the Record API - ServiceNow Fluent.
recipientFields	Array	A list of fields that reference the users or user groups to receive the notification. The fields must be reference fields. For example, if a notification uses the Incident [incident] table, you can specify the `opened_by` field to send the notification to users or groups who opened the incident. This list of users or groups is variable and depends on the values of the associated task record. You can also select a field that includes an email address string to send a notification to that address.
recipientGroups	Array	A list of variable identifiers or sys_ids of user groups [sys_user_group] to receive the notification. To define a user group, use the Record API - ServiceNow Fluent. **Note:** Group members receive individual notifications only if **Include members** is selected in the group record.
excludeDelegates	Boolean	Flag that indicates whether to exclude delegated users. Set this property to true to help prevent the instance from sending email notifications to delegates of the users and members of the groups you selected. Default: false
isSubscribableByAllUsers	Boolean	Flag that indicates whether to allow all users to subscribe to the notification. For more information, see Subscription-based notifications. **Note:** If the record contains sensitive or protected data, consider restricting the recipient list only to those users and groups who normally have access to it, leaving the value of this property as false. You can also configure your notification content so that private or sensitive data isn't exposed. For example, you could insert a link back to the associated record, so that details aren't revealed in the notification.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Default: false
sendToCreator	Boolean	Flag that indicates whether to send the notification to the person who performed the action that started the notification process, if the person is also a recipient. If the event creator isn't specified in one of the recipient fields, the event creator doesn't receive a notification regardless of the setting in this field. Default: false
eventParm1WithRecipient	Boolean	Flag that indicates whether the event parameter 1 contains one or more notification recipients. This property only applies to event-based notifications.
eventParm2WithRecipient	Boolean	Flag that indicates whether the event parameter 2 contains one or more notification recipients. This property only applies to event-based notifications.

```
recipientDetails: {
    recipientGroups: ["d625dccec0a8016700a222a0f7900d06"], // IT Support group sys_id
    recipientFields: ["assigned_to", "caller_id"],
    sendToCreator: false,
    isSubscribableByAllUsers: false
}
```

Flow API - ServiceNow Fluent

The Flow API defines flows and subflows [sys_hub_flow], which automate business processes with reusable multiple-step components.

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

Create a flow using the Flow object. For more information, see Flow object.

Create a subflow using the Subflow object. For more information, see Subflow object.

For general information about flows, see Exploring flows. For general information about subflows, see Exploring subflows.

Related topics

ServiceNow Fluent

Flow object

Create a flow [sys_hub_flow] to run a sequence of actions and flow logic when a set of trigger conditions occur.

Flows only run when their trigger conditions are met. Use flows for event-driven automation that requires a consistent and predefined trigger.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

A Flow object consists of these components.

- One configuration object that defines the flow configuration properties.
- One wfa.trigger function that defines when to run the flow.
- One Body function that defines what actions and flow logic to run.
  - Zero or more wfa.action functions
  - Zero or more wfa.flow_logic functions

Properties

Name	Type	Description
configuration	Object	Required. An object containing the metadata configuration properties for the Fluent object or function.
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
name	String	Required. Display name of the flow. The flow name should be meaningful and descriptive to easily identify its purpose.
description	String	Description of what the flow does.
runAs	String	Determines under which user context the flow actions are run. The system user setting runs as a privileged user that bypasses role-based ACLs. The user setting runs with the privileges and roles of the user who started the flow. Valid values: system, user Default: user
runWithRoles	String	Specify the roles that the flow uses while running. For more information about running a flow with roles, see Flow roles.
flowPriority	String	Defines the execution priority of the flow in the Flow Engine. Valid values: LOW, MEDIUM, HIGH Default: MEDIUM
protection	String	Defines whether the flow is read-only (read) or editable (empty string). Valid values: read, '' Default: ''
access	String	Defines whether a flow is public or private. Valid values: public, package_private

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Default: public
flowVariables	Object	Object defining data variables available to the flow using Column types. Use the get_table_schema function to fetch field definitions.
wfa.trigger	Function	Defines when to run the flow. When the trigger conditions are met, the system runs the flow using the data provided by the trigger. For more information about the *`wfa.trigger`* function, see wfa.trigger function.
Flow body	Function	*`Flow body`* is an *`Arrow function`* in TypeScript that represents the execution steps in the flow. The Flow Body receives the *`params`* parameter as its input, which contains the *`wfa.trigger`* and *`flowVariables`* objects.The steps in the *`Flow body`* consist of these function types: • *`wfa.action`* function: Execute specific tasks and operations within a flow, such as creating records, sending emails, requesting approvals, or integrating with external systems. For more information about the *`wfa.action`* function, see wfa.action function. • *`wfa.flow_logic`* function: Control how flows execute, providing conditional branching, iteration, and flow structuring capabilities. For more information about *`wfa.flow_logic`* function, see wfa.flow_logic function. You can use these parameter values in the Flow body function. **params.trigger.current** Data type: Record. Parameter that stores the current record that started a create record, update record, or create or update record flow. **params.trigger.previous** Data type: Record. Parameter that stores the previous record values that started an update record flow. **params.trigger.table_name** Data type: String. Parameter that stores the table name of the current record that started the flow. **params.trigger.flowVariables** Data type: Object. Parameter that stores the flow variables defined in this flow.

This example creates a flow that runs when an incident is created with an empty origin field value. The flow alerts managers and team members based on the incident severity.

```
import { action, Flow, wfa, trigger } from '@servicenow/sdk/automation'

export const incidentSeverityAlertFlow = Flow(
    {
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
        $id: Now.ID['incident_severity_alert_flow'],
        name: 'Incident Severity Alert Flow',
        description: 'Alerts managers and team members based on incident severity
for incidents with empty origin',
    },
    wfa.trigger(
        trigger.record.created,
        { $id: Now.ID['empty_origin_incident_trigger'] },
        {
            table: 'incident',
            condition: 'origin=NULL',
            run_flow_in: 'background',
            run_on_extended: 'false',
            run_when_setting: 'both',
            run_when_user_setting: 'any',
            run_when_user_list: [],
        }
    ),
    (params) => {
        wfa.action(
            action.core.log,
            { $id: Now.ID['log_incident_short_description'] },
            {
                log_level: 'info',
                log_message: `Incident created:
 ${wfa.dataPill(params.trigger.current.short_description, 'string')
}`,
            }
        )

        wfa.flowLogic.if(
            {
                $id: Now.ID['check_severity_high'],
                condition:
 `${wfa.dataPill(params.trigger.current.severity, 'string')}=1`,
                annotation: 'High severity (1)',
            },
            () => {
                const assignmentGroup = wfa.action(
                    action.core.lookUpRecord,
                    { $id: Now.ID['lookup_assignment_group'] },
                    {
                        table: 'sys_user_group',
                        conditions:
 `sys_id=${wfa.dataPill(params.trigger.current.assignment_group,
 'reference')}`,
                        sort_type: 'sort_asc',

 if_multiple_records_are_found_action: 'use_first_record',
                    }
                )

                const manager = wfa.action(
                    action.core.lookUpRecord,
                    { $id: Now.ID['lookup_manager_details'] },
                    {
                        table: 'sys_user',
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
                        conditions:
 `sys_id=${wfa.dataPill(assignmentGroup.Record.manager, 'reference')
}`,
                        sort_type: 'sort_asc',

 if_multiple_records_are_found_action: 'use_first_record',
                    }
                )

                wfa.action(
                    action.core.sendNotification,
                    { $id: Now.ID['send_urgent_email_to_manager'] },
                    {
                        table_name: 'incident',
                        record:
 wfa.dataPill(params.trigger.current.sys_id, 'reference'),

 notification: 'high_severity_incident_manager_notification',
                    }
                )

                wfa.flowLogic.forEach(

 wfa.dataPill(params.trigger.current.additional_assignee_list, 'a
rray.string'),
                    { $id: Now.ID['foreach_additional_assignees_high'] },
                    () => {
                        const assignee = wfa.action(
                            action.core.lookUpRecord,
                            { $id:
 Now.ID['lookup_assignee_details_high'] },
                            {
                                table: 'sys_user',
                                conditions:
 `sys_id=${wfa.dataPill(params.trigger.current.additional_assign
ee_list, 'array.string')}`,
                                sort_type: 'sort_asc',

 if_multiple_records_are_found_action: 'use_first_record',
                            }
                        )

                        wfa.action(
                            action.core.sendSms,
                            { $id:
 Now.ID['send_sms_to_assignee_high'] },
                            {
                                recipients:
 `${wfa.dataPill(assignee.Record.phone, 'string')}`,
                                message: `High severity
 incident:
 ${wfa.dataPill(params.trigger.current.short_description, 'string')
}`,
                            }
                        )
                    }
                )
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
                wfa.action(
                    action.core.updateRecord,
                    { $id: Now.ID['update_work_notes_high'] },
                    {
                        table_name: 'incident',
                        record:
wfa.dataPill(params.trigger.current.sys_id, 'reference'),
                        values: TemplateValue({
                            work_notes: `Manager
${wfa.dataPill(manager.Record.name, 'string')} notified via email
and team notified via SMS.`,
                        }),
                    }
                )
            }
        )

        wfa.flowLogic.elseIf(
            {
                $id: Now.ID['check_severity_medium'],
                condition:
`${wfa.dataPill(params.trigger.current.severity, 'string')}=2`,
                annotation: 'Medium severity (2)',
            },
            () => {
                wfa.flowLogic.forEach(

wfa.dataPill(params.trigger.current.additional_assignee_list, 'a
rray.string'),
                    { $id:
Now.ID['foreach_additional_assignees_medium'] },
                    () => {
                        const assignee = wfa.action(
                            action.core.lookUpRecord,
                            { $id:
Now.ID['lookup_assignee_details_medium'] },
                            {
                                table: 'sys_user',
                                conditions:
`sys_id=${wfa.dataPill(params.trigger.current.additional_assign
ee_list, 'array.string')}`,
                                sort_type: 'sort_asc',

if_multiple_records_are_found_action: 'use_first_record',
                            }
                        )

                        wfa.action(
                            action.core.sendSms,
                            { $id:
Now.ID['send_sms_to_assignee_medium'] },
                            {
                                recipients:
`${wfa.dataPill(assignee.Record.phone, 'string')}`,
                                message: `Medium severity
incident:
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
 ${wfa.dataPill(params.trigger.current.short_description, 'string')
}`,
                                }
                            )
                        }
                    )
                }
            )

            wfa.action(
                action.core.updateRecord,
                { $id: Now.ID['update_incident_to_in_progress'] },
                {
                    table_name: 'incident',
                    record:
 wfa.dataPill(params.trigger.current.sys_id, 'reference'),
                    values: TemplateValue({ state: '2' }),
                }
            )
        }
)
```

This example runs when a change request is approved and notifies the requester.

```
import { action, Flow, wfa, trigger } from '@servicenow/sdk/automation'

export const changeRequestApprovalNotificationFlow = Flow(
    {
        $id: Now.ID['change_request_approval_notification_flow'],
        name: 'Change Request Approval Notification Flow',
        description: 'Sends formatted notification to requester when change
 request is approved',
    },
    wfa.trigger(
        trigger.record.updated,
        { $id: Now.ID['change_request_approved_trigger'] },
        {
            table: 'change_request',
            condition: 'approval=approved',
            run_flow_in: 'background',
            trigger_strategy: 'unique_changes',
            run_when_user_list: [],
            run_when_setting: 'both',
            run_on_extended: 'false',
            run_when_user_setting: 'any',
        }
    ),
    (params) => {
        const requester = wfa.action(
            action.core.lookUpRecord,
            { $id: Now.ID['lookup_requester_details'] },
            {
                table: 'sys_user',
                conditions:
 `sys_id=${wfa.dataPill(params.trigger.current.requested_by, 'refer
ence')}`,
                sort_type: 'sort_asc',
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
if_multiple_records_are_found_action: 'use_first_record',
            }
        )

        wfa.action(
            action.core.sendEmail,
            { $id: Now.ID['send_approval_notification_email'] },
            {
                table_name: 'change_request',
                watermark_email: true,
                ah_subject: `Change Request
${wfa.dataPill(params.trigger.current.number, 'string')} -
Approved`,
                ah_body: `Your change request has been
approved.`,
                record:
wfa.dataPill(params.trigger.current.sys_id, 'reference'),
                ah_to:
wfa.dataPill(requester.Record.email, 'string'),
            }
        )

        wfa.action(
            action.core.updateRecord,
            { $id: Now.ID['update_work_notes_notification_sent'] },
            {
                table_name: 'change_request',
                record:
wfa.dataPill(params.trigger.current.sys_id, 'reference'),
                values: TemplateValue({
                    work_notes: `Approval notification
sent to ${wfa.dataPill(requester.Record.name, 'string')}
(${wfa.dataPill(requester.Record.email, 'string')})`,
                }),
            }
        )
    }
)
```

Subflow object

Create a subflow [sys_hub_flow] to run a reusable sequence of actions and flow logic when called by a flow or API.

Subflows run when called by a flow or an API. Use subflows for on-demand automation that can be called by multiple flows.

Subflows consist of these components.

- One configuration object that defines the subflow configuration properties.
  - Zero or one inputs object that defines subflow inputs.
  - Zero or one outputs object that defines subflow outputs.
  - Zero or one flow variables object that defines flow variables.
- One Body function that defines what actions, flow logic, and subflows to run.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Zero or more wfa.action functions
- Zero or more wfa.flow_logic functions

Properties

Name	Type	Description
configuration	Object	Required. An object containing the metadata configuration properties for the Fluent object or function.
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
name	String	Required. Display name of the subflow. The subflow name should be meaningful and descriptive to easily identify its purpose. This property is part of the Subflow configuration object.
description	String	Description of what the flow does. This property is part of the Subflow configuration object.
runAs	String	Determines under which user context the subflow actions are executed. Valid values: system, user Default: user This property is part of the Subflow configuration object.
runWithRoles	Array	Specify the roles that the subflow uses while running. This property is part of the Subflow configuration object.
flowPriority	String	Defines the execution priority of the subflow in the Flow Engine. Valid values: LOW, MEDIUM, HIGH Default: Medium This property is part of the Subflow configuration object.
protection	String	Defines whether the subflow is read-only (read) or editable (empty string). Valid values: read, '' This property is part of the Subflow configuration object.
access	String	Defines the visibility/accessibility of the subflow to other subflow authors. Valid values: public, package_private

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Default: public This property is part of the Subflow configuration object.
category	String	Category for organizing the subflow. This property is part of the Subflow configuration object.
inputs	Object	Object defining input parameters using Column types. This property is part of the Subflow configuration object.
outputs	Object	Object defining output parameters using Column types. This property is part of the Subflow configuration object.
flowVariables	Object	Object defining data variables available to the subflow using Column types. This property is part of the Subflow configuration object.
Flow body	Function	*`Flow body`* is an *`Arrow function`* in TypeScript that represents the execution steps in the flow. The Flow Body receives the *`_params`* parameter as its input, which contains the *`inputs`* and *`flowVariables`* objects.The steps in the *`Flow body`* consist of these function types: • *`wfa.action`* function: Execute specific tasks and operations within a flow, such as creating records, sending emails, requesting approvals, or integrating with external systems. For more information about the *`wfa.action`* function, see wfa.action function. • *`wfa.flow_logic`* function: Control how flows execute, providing conditional branching, iteration, and flow structuring capabilities. For more information about *`wfa.flow_logic`* function, see wfa.flow_logic function. You can use these parameter values in the Flow body function. **_params.trigger.current** Data type: Record. Parameter that stores the current record that started a create record, update record, or create or update record flow. **_params.trigger.previous** Data type: Record. Parameter that stores the previous record values that started an update record flow. **_params.trigger.table_name** Data type: String. Parameter that stores the table name of the current record that started the flow. **_params.trigger.flowVariables** Data type: Object. Parameter that stores the flow variables defined in this flow.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

This example uses a new user's name and office location to look up a User record, send a welcome notification, and look up an unassigned hardware asset and an available office location. If there is an available laptop asset and office location, then each are assigned to the new user.

```
import { Subflow, action, wfa } from '@servicenow/sdk/automation'
import { BooleanColumn, ReferenceColumn, StringColumn }
 from '@servicenow/sdk/core'

export const newUserOnboardingSubflow = Subflow(
    {
        $id: Now.ID['new_user_onboarding_subflow'],
        name: 'New User Onboarding Subflow',
        description: 'Sends welcome notification, assigns laptop and desk, returns
assignment results',
        inputs: {
            user_sys_id: ReferenceColumn({
                label: 'User',
                referenceTable: 'sys_user',
                mandatory: true,
            }),
            office_location: ReferenceColumn({
                label: 'Office Location',
                referenceTable: 'cmn_location',
                mandatory: true,
            }),
        },
        outputs: {
            laptop_assigned: BooleanColumn({ label: 'Laptop
Assigned' }),
            desk_assigned: BooleanColumn({ label: 'Desk
Assigned' }),
            laptop_number: StringColumn({ label: 'Laptop Asset
Number', maxLength: 40 }),
            desk_number: StringColumn({ label: 'Desk Asset Number',
 maxLength: 40 }),
        },
        flowVariables: {
            laptop_found: BooleanColumn({ label: 'Laptop Found
Flag', default: false }),
            desk_found: BooleanColumn({ label: 'Desk Found
Flag', default: false }),
        },
    },
    (params) => {
        const user = wfa.action(
            action.core.lookUpRecord,
            { $id: Now.ID['lookup_user'] },
            {
                table: 'sys_user',
                conditions:
 `sys_id=${wfa.dataPill(params.inputs.user_sys_id, 'reference')}`,
                sort_type: 'sort_asc',

 if_multiple_records_are_found_action: 'use_first_record',
            }
        )
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
        wfa.action(
            action.core.sendNotification,
            { $id: Now.ID['send_welcome_notification'] },
            {
                table_name: 'sys_user',
                record:
 wfa.dataPill(params.inputs.user_sys_id, 'reference'),
                notification: 'new_user_welcome_notification',
            }
        )

        const availableLaptop = wfa.action(
            action.core.lookUpRecord,
            { $id: Now.ID['lookup_available_laptop'] },
            {
                table: 'alm_hardware',

 conditions: 'assigned_to=NULL^install_status=1^substatus=available',
                sort_type: 'sort_asc',

 if_multiple_records_are_found_action: 'use_first_record',
            }
        )

        wfa.flowLogic.if(
            {
                $id: Now.ID['check_laptop_available'],
                condition:
 `${wfa.dataPill(availableLaptop.Record.sys_id, 'reference')}!
=NULL`,
                annotation: 'Available laptop found',
            },
            () => {
                wfa.action(
                    action.core.updateRecord,
                    { $id: Now.ID['assign_laptop'] },
                    {
                        table_name: 'alm_hardware',
                        record:
 wfa.dataPill(availableLaptop.Record.sys_id, 'reference'),
                        values: TemplateValue({
                            assigned_to:
 wfa.dataPill(params.inputs.user_sys_id, 'reference'),
                            install_status: '2',
                            substatus: 'in_use',
                        }),
                    }
                )
            }
        )

        wfa.flowLogic.else(
            { $id: Now.ID['no_laptop_available'] },
            () => {
                wfa.action(
                    action.core.log,
                    { $id: Now.ID['log_no_laptop'] },
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
                    {
                        log_level: 'warn',
                        log_message: `No available laptop
found for ${wfa.dataPill(user.Record.name, 'string')} — manual
assignment required.`,
                    }
                )
            }
        )

        const availableDesk = wfa.action(
            action.core.lookUpRecord,
            { $id: Now.ID['lookup_available_desk'] },
            {
                table: 'alm_asset',
                conditions:
`assigned_to=NULL^location=${wfa.dataPill(params.inputs.office_
location, 'reference')}^install_status=1^substatus=available`,
                sort_type: 'sort_asc',

if_multiple_records_are_found_action: 'use_first_record',
            }
        )

        wfa.flowLogic.if(
            {
                $id: Now.ID['check_desk_available'],
                condition:
`${wfa.dataPill(availableDesk.Record.sys_id, 'reference')}!=NULL`,
                annotation: 'Available desk found at office location',
            },
            () => {
                wfa.action(
                    action.core.updateRecord,
                    { $id: Now.ID['assign_desk'] },
                    {
                        table_name: 'alm_asset',
                        record:
wfa.dataPill(availableDesk.Record.sys_id, 'reference'),
                        values: TemplateValue({
                            assigned_to:
wfa.dataPill(params.inputs.user_sys_id, 'reference'),
                            install_status: '2',
                            substatus: 'in_use',
                        }),
                    }
                )
            }
        )

        wfa.flowLogic.else(
            { $id: Now.ID['no_desk_available'] },
            () => {
                wfa.action(
                    action.core.log,
                    { $id: Now.ID['log_no_desk'] },
                    {
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
                        log_level: 'warn',
                        log_message: `No available desk at
office location for ${wfa.dataPill(user.Record.name, 'string')} —
manual assignment required.`,
                    }
                )
            }
        )

        wfa.action(
            action.core.sendNotification,
            { $id: Now.ID['send_onboarding_complete'] },
            {
                table_name: 'sys_user',
                record:
wfa.dataPill(params.inputs.user_sys_id, 'reference'),
                notification: 'user_onboarding_complete_notification',
            }
        )

        wfa.flowLogic.assignSubflowOutputs(
            {
                $id: Now.ID['assign_outputs'],
                annotation: 'Return laptop and desk assignment results',
            },
            params.outputs,
            {
                laptop_assigned: true,
                desk_assigned: true,
                laptop_number:
wfa.dataPill(availableLaptop.Record.asset_tag, 'string'),
                desk_number:
wfa.dataPill(availableDesk.Record.asset_tag, 'string'),
            }
        )
    }
)
```

wfa.trigger function

Run a flow when the start conditions of a specific trigger type are met. Triggers determine when a flow runs and what data is available from the flow start conditions.

Add a *`wfa.trigger`* function to the *`Flow Body`* function of a *`Flow`* object.

The following types of action instances are supported:

- *`trigger.record.created`*
- *`trigger.record.updated`*
- *`trigger.record.createdOrUpdated`*
- *`trigger.scheduled.daily`*
- *`trigger.scheduled.weekly`*
- *`trigger.scheduled.monthly`*
- *`trigger.scheduled.repeat`*
- *`trigger.scheduled.runOnce`*

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- `trigger.application.inboundEmail`
- `trigger.application.serviceCatalog`
- `trigger.application.slaTask`
- `trigger.application.knowledgeManagement`
- `trigger.application.remoteTableQuery`

.

For more information about available actions, see Workflow Studio flow trigger types.

Properties

Name	Type	Description
trigger	String	Name of the specific trigger to run. All action names use a dot notation that starts with `trigger`. For example, `trigger.record.created`, `trigger.scheduled.daily`, or `trigger.application.serviceCatalog`.
configuration	Object	Required. An object containing the metadata configuration properties for the Fluent object or function.
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
inputs	Object	An object containing any input parameters required by the trigger such as `table` or `condition`.

This example runs a flow when a high priority incident record is created.

```
// Trigger definition - defines when the flow starts
wfa.trigger(
    trigger.record.created,  // TriggerDefinition - record creation event
    {
        $id: Now.ID['incident_trigger'],  // string | guid, mandatory - unique identifier
        annotation: 'When high priority incident is created'  // string, optional -
describes trigger purpose
    },
    {
        table: 'incident',  // string, mandatory - table name to monitor
        condition: 'priority=1^ORpriority=2',  // string, optional - encoded query filter
        run_on_extended: 'false',  // string, optional, default: 'false' - run on child
tables
        run_flow_in: 'any',  // string, optional, default: 'any' - 'any', 'background', or
'foreground'
        run_when_setting: 'both',  // string, optional, default: 'both' - 'both',
'non_interactive', or 'interactive'
        run_when_user_list: [],  // Array, optional, default: [] - need to use
runQuery to get user sys_ids from sys_user table
        run_when_user_setting: 'any'  // string, optional, default: 'any' - 'any',
'one_of', or 'not_one_of'
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    }
)
```

This examples starts a flow daily at 9 AM.

```
// # CORRECT - Daily trigger at specific time
wfa.trigger(
    trigger.scheduled.daily,
    {
        $id: Now.ID['daily_report_trigger'],
        annotation: 'Runs daily at 9 AM'
    },
    {
        time: Time({ hours: 9, minutes: 0, seconds: 0 })
    }
)
```

This examples starts a flow when there is an email reply to an incident record.

```
// Trigger when email arrives for incident table
wfa.trigger(
    trigger.application.inboundEmail, // TriggerDefinition - inbound email
event
    {
        $id: Now.ID['incident_email_trigger'], // string | guid, mandatory
        annotation: 'Process emails for incident table' // string, optional
    },
    {
        target_table: 'incident', // string, optional - table for email replies/forwards
        email_conditions: '', // string, optional - filter which emails trigger the flow
        order: 100, // number, optional - execution order when multiple email triggers
exist
        stop_condition_evaluation: true // boolean, optional - stop
evaluating other triggers if this one matches
    }
)

// Access email data in flow body:
(_params) => {
    const emailSubject = _params.trigger.email.subject // Email subject
    const emailBody = _params.trigger.email.body // Email body
    const emailFrom = _params.trigger.email.from // Sender email address
}
```

This example starts a flow when a Service Catalog item is requested. This example assumes that you have configured the Service Catalog item to run a flow.

```
import { Flow, wfa, trigger, action } from "@servicenow/sdk/automation";

Flow(
  {
    $id: Now.ID["laptop_fulfillment"],
    name: "Laptop Fulfillment Workflow"
  },

  wfa.trigger(
    trigger.application.serviceCatalog,
    { $id: Now.ID["catalog_trigger"] },
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    {
      run_flow_in: "background"
    }
  ),

  _params => {
    // Log the catalog request
    wfa.action(
      action.core.log,
      { $id: Now.ID["log_request"] },
      {
        log_level: "info",
        log_message: `Processing catalog request:
${wfa.dataPill(_params.trigger.request_item, "string")}`
      }
    );

    // Create fulfillment task
    wfa.action(
      action.core.createCatalogTask,
      { $id: Now.ID["create_task"] },
      {
        ah_requested_item: wfa.dataPill(
          _params.trigger.request_item,
          "reference"
        ),
        ah_short_description: "Fulfill laptop request",
        ah_fields: TemplateValue({
          assignment_group: "Hardware Fulfillment Team",
          priority: "3"
        })
      }
    );
  }
);
```

wfa.action function

Run a specific action instance from a flow or subflow. Actions determine what data is generated, updated, or retrieved.

Add *wfa.action* functions to the *Flow Body* function of a *Flow* or *Subflow* object.

The following types of action instances are supported:

- *action.core.createRecord*
- *action.core.createOrUpdateRecord*
- *action.core.deleteRecord*
- *action.core.lookUpRecord*
- *action.core.lookUpRecords*
- *action.core.updateRecord*
- *action.core.updateMultipleRecords*
- *action.core.sendNotification*
- *action.core.sendEmail*

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- `action.core.sendSms`
- `action.core.askForApproval`
- `action.core.waitForApproval`
- `action.core.createTask`
- `action.core.createCatalogTask`
- `action.core.submitCatalogItemRequest`
- `action.core.getCatalogVariables`
- `action.core.recordProducer`
- `action.core.log`
- `action.core.waitForCondition`
- `action.core.waitForMessage`
- `action.core.slaPercentageTimer`

.

For more information about available actions, see Workflow Studio actions.

Properties

Name	Type	Description
action	String	Name of the specific action to run. All action names use a dot notation that starts with `action.core`. For example, `action.core.lookUpRecord`, `action.core.lookUpRecords`, `action.core.createRecord`, or `action.core.updateRecord`.
configuration	Object	Required. An object containing the metadata configuration properties for the Fluent object or function.
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
inputs	Object	An object containing any input parameters required by the action such as `table_name` or `values`.

This example runs when a change request is approved and notifies the requester. The flow uses the Look Up Records, Send Email, and Update Record actions.

```
import { action, Flow, wfa, trigger } from '@servicenow/sdk/automation'

export const changeRequestApprovalNotificationFlow = Flow(
    {
        $id: Now.ID['change_request_approval_notification_flow'],
        name: 'Change Request Approval Notification Flow',
        description: 'Sends formatted notification to requester when change
 request is approved',
    },
    wfa.trigger(
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
        trigger.record.updated,
        { $id: Now.ID['change_request_approved_trigger'] },
        {
            table: 'change_request',
            condition: 'approval=approved',
            run_flow_in: 'background',
            trigger_strategy: 'unique_changes',
            run_when_user_list: [],
            run_when_setting: 'both',
            run_on_extended: 'false',
            run_when_user_setting: 'any',
        }
    ),
    (params) => {
        const requester = wfa.action(
            action.core.lookUpRecord,
            { $id: Now.ID['lookup_requester_details'] },
            {
                table: 'sys_user',
                conditions:
 `sys_id=${wfa.dataPill(params.trigger.current.requested_by, 'refer
ence')}`,
                sort_type: 'sort_asc',

 if_multiple_records_are_found_action: 'use_first_record',
            }
        )

        wfa.action(
            action.core.sendEmail,
            { $id: Now.ID['send_approval_notification_email'] },
            {
                table_name: 'change_request',
                watermark_email: true,
                ah_subject: `Change Request
 ${wfa.dataPill(params.trigger.current.number, 'string')} -
 Approved`,
                ah_body: `Your change request has been
 approved.`,
                record:
 wfa.dataPill(params.trigger.current.sys_id, 'reference'),
                ah_to:
 wfa.dataPill(requester.Record.email, 'string'),
            }
        )

        wfa.action(
            action.core.updateRecord,
            { $id: Now.ID['update_work_notes_notification_sent'] },
            {
                table_name: 'change_request',
                record:
 wfa.dataPill(params.trigger.current.sys_id, 'reference'),
                values: TemplateValue({
                    work_notes: `Approval notification
 sent to ${wfa.dataPill(requester.Record.name, 'string')}
 (${wfa.dataPill(requester.Record.email, 'string')})`,
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
                }),
            }
        )
    }
)
```

This example runs a flow when a change request record is created or updated as high impact. The flow runs the Update Record to mark the change request as high risk, the Look Up Record to find the group with the name CAB, the Send Notification to send a high risk change notification to the CAB group, and the Log action to document both the record update and the notification sent.

```
import { action, Flow, wfa, trigger } from '@servicenow/sdk/automation'

export const changeRiskTaggingFlow = Flow(
    {
        $id: Now.ID['change_risk_tagging_flow'],
        name: 'Change Risk Tagging Flow',
        description: 'Tags change requests with high-risk label when created or
updated with high impact',
    },
    wfa.trigger(
        trigger.record.createdOrUpdated,
        { $id: Now.ID['change_risk_trigger'] },
        {
            table: 'change_request',
            condition: 'active=true^impact=1',
            run_flow_in: 'background',
            run_on_extended: 'false',
            run_when_setting: 'both',
            run_when_user_setting: 'any',
            run_when_user_list: [],
        }
    ),
    (params) => {
        wfa.action(
            action.core.updateRecord,
            { $id: Now.ID['tag_high_risk'] },
            {
                table_name: 'change_request',
                record:
 wfa.dataPill(params.trigger.current.sys_id, 'reference'),
                values: TemplateValue({
                    risk: 'high',
                    work_notes: 'Automatically tagged as high-risk due to high
impact.',
                }),
            }
        )

        const cab = wfa.action(
            action.core.lookUpRecord,
            { $id: Now.ID['lookup_cab_group'] },
            {
                table: 'sys_user_group',
                conditions: 'name=CAB^active=true',
                sort_type: 'sort_asc',
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
if_multiple_records_are_found_action: 'use_first_record',
            }
        )

        wfa.action(
            action.core.sendNotification,
            { $id: Now.ID['notify_cab_high_risk'] },
            {
                table_name: 'change_request',
                record:
wfa.dataPill(params.trigger.current.sys_id, 'reference'),
                notification: 'high_risk_change_cab_notification',
            }
        )

        wfa.action(
            action.core.log,
            { $id: Now.ID['log_risk_tagged'] },
            {
                log_level: 'warn',
                log_message: `Change
${wfa.dataPill(params.trigger.current.number, 'string')} tagged
as high-risk. CAB group ${wfa.dataPill(cab.Record.name, 'string')}
notified.`,
            }
        )
    }
)
```

wfa.flow_logic function

Run a specific flow logic instance from a flow or subflow. Flow logic determines how and when data is used.

Add `wfa.flow_logic` functions to the *Flow Body* function of a `Flow` or `Subflow` object.

The following types of flow logic instances are supported:

- `if`
- `elseIf`
- `else`
- `forEach`
- `waitForADuration`
- `exitLoop`, `endFlow`, `skipIteration`, `setFlowVariables`, and `assignSubflowOutputs`

.

For more information about available flow logic, see Workflow Studio flow logic.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties

Name	Type	Description
configuration	Object	Required. An object containing the metadata configuration properties for the Fluent object or function.
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
annotation	String	Describe what this instance of flow logic does. For example, `annotation: 'Priority is critical'`.
condition	String	Required in a *`wfa.flow_logic.if`* and *`wfa.flow_logic.elseIf`* function. Data values that must be true to run this flow logic. For example, `condition: \`\${_params.trigger.current.priority}=1\`,`.
item	Array	Required in a *`wfa.flow_logic.forEach`* function. An array or a collection of records to iterate over. Typically, this list of records is dynamically generated from a *`wfa.action`* function call to the *`action.core.lookUpRecords`* action. For example, a `problems.Records` array can be generated from an *`action.core.lookUpRecords`* action. `(_params) => { //Find all the newly created problem records for the past day const problems = wfa.action(action.core.lookUpRecords, { $id: Now.ID['daily_unassigned_new_problems_triage_flow_step1'], annotation: 'Find all the newly created problem records for the past day', }, { table: 'problem', conditions: 'sys_created_onONYesterday@javascript:gs.beginningOfYesterday()@javascript:gs.endOfYesterday()', }) //Iterate over the problems wfa.flow_logic.forEach(problems.Records, { ... }`
label	String	Display value describing this instance of flow logic. For example, `label: 'If Priority is critical',`.
variables	Schema	Required in a *`wfa.flow_logic.assignSubflowOutputs`* and

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		wfa.flow_logic.setFlowVariables function. A schema produced by the FlowVariables API that defines the data structure of the flow variables. For example, the `flowVars` schema can be generated from FlowVariables. `import { FlowVariables } from '@servicenow/sdk/automation'` `// Define flow variables` `const flowVars = FlowVariables({ status: StringColumn({ label: 'Status' }), count: IntegerColumn({ label: 'Count' }), isActive: BooleanColumn({ label: 'Is Active' }), })` `// Flow body` `(_params) => { // Set flow variables with literal values wfa.flow_logic.setFlowVariables( { $id: Now.ID['initialize_variables'], annotation: 'Initialize flow variables with default values', }, _params.flowVariables, { ... }` **Important:** Always use `_params.flowVariables` as the argument for this property. Passing the wrong schema will cause validation errors.
Flow logic body	Function	*Flow logic body* is an *Arrow function* in TypeScript that represents the execution steps in the flow logic. The Flow Body receives the *_params* parameter as its input, which contains the *wfa.trigger* and *flowVariables* objects. The steps in the *Flow logic body* consist of these functions calls: • *wfa.action* function: Run a specific action instance from a flow or subflow. Actions determine what data is generated, updated, or retrieved.. For more information about the *wfa.action* function, see wfa.action function. • *wfa.flow_logic* function: Run a specific flow logic instance from a flow or subflow. Flow logic determines how and when data is used. For more information about the wfa.flow_logic function, see wfa.flow_logic function.

This example shows checking for conditions in the trigger record. If the current record has a priority of 1, it assigns the record to a critical priority team. Else If the current record has a priority

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

value of 2, it assigns the record to a high priority team. If neither condition is met, the record is assigned to a general team.

```
// Flow body showing if/elseIf/else usage
(_params) => {
    wfa.flow_logic.if(
        {
            $id: Now.ID['check_critical_priority'],
            condition: `\${_params.trigger.current.priority}=1`,
            annotation: 'Priority is critical',
        },
        () => {
            wfa.action(
                action.core.updateRecord,
                { $id: Now.ID['assign_critical_team'],
 annotation: 'Assign to critical team' },
                { record: _params.trigger.current,
 table_name: 'incident', values: 'assignment_group=critical_team' }
            )
        }
    )
    wfa.flow_logic.elseIf(
        { $id: Now.ID['check_high_priority'], condition:
 `\${_params.trigger.current.priority}=2`, annotation: 'Priority is
high' },
        () => {
            wfa.action(
                action.core.updateRecord,
                { $id: Now.ID['assign_high_team'], annotation: 'Assign
to high team' },
                { record: _params.trigger.current,
 table_name: 'incident', values: 'assignment_group=high_team' }
            )
        }
    )
    wfa.flow_logic.else(
        { $id: Now.ID['assign_default'], annotation: 'Default assignment' },
        () => {
            wfa.action(
                action.core.updateRecord,
                { $id: Now.ID['assign_general_team'],
 annotation: 'Assign to general team' },
                { record: _params.trigger.current,
 table_name: 'incident', values: 'assignment_group=general_team' }
            )
        }
    )
}
```

This example iterates through a list of problem records to find unassigned records, assign them to a problem triage group, and send an email to the group.

```
// Flow body showing forEach usage with lookUpRecords
(_params) => {
    //Find all the newly created problem records for the past day
    const problems = wfa.action(
        action.core.lookUpRecords,
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
        {
            $id: Now.ID['daily_unassigned_new_problems_triage_flow_step1'],
            annotation: 'Find all the newly created problem records for the past
day',
        },
        {
            table: 'problem',
            conditions:

  'sys_created_onONYesterday@javascript:gs.beginningOfYesterday()@javascript:gs.endOf
Yesterday()',
        }
    )

    //Iterate over the problems
    wfa.flow_logic.forEach(
        problems.Records,
        { $id: Now.ID['daily_unassigned_new_problems_triage_flow_step2'],
 annotation: 'Iterate each problem' },
        (item) => {
            //Check if the problem is not assigned
            wfa.flow_logic.if(
                {
                    $id:
 Now.ID['daily_unassigned_new_problems_triage_flow_step3'],
                    condition:
 `\${item.assignment_group}ISEMPTY`,
                    annotation: 'Check if problem is not assigned',
                },
                () => {
                    wfa.action(
                        action.core.updateRecord,
                        {
                            $id:
 Now.ID['daily_unassigned_new_problems_triage_flow_step4'],
                            annotation: 'Update the problem record state',
                        },
                        { table_name: 'problem', record: item,
 values: 'state=1' }
                    )

                    //Send notification to the group
                    wfa.action(
                        action.core.sendNotification,
                        {
                            $id:
 Now.ID['daily_unassigned_new_problems_triage_flow_step5'],
                            annotation: 'Send notification to the group',
                        },
                        {
                            table_name: 'problem',
                            record: item,
                            notification: '',
                        }
                    )
                }
            )
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
        }
    )
}
```

This example sets flow variables to dynamic values stored in data pills.

```
import { FlowVariables } from '@servicenow/sdk/automation'

const flowVars = FlowVariables({
    incidentPriority: IntegerColumn({ label: 'Incident Priority' }),
    assignedGroup: StringColumn({ label: 'Assigned Group' }),
    threshold: IntegerColumn({ label: 'Threshold' }),
    actionResult: StringColumn({ label: 'Action Result' }),
})

(_params) => {
    // Lookup action result
    const lookupResult = wfa.action(
        action.core.lookUpRecord,
        { $id: Now.ID['lookup_config'], annotation: 'Get configuration' },
        { table: 'sys_properties', conditions: 'name=incident.threshold' }
    )

    // Set flow variables using datapills from trigger, action outputs, and other flow variables
    wfa.flow_logic.setFlowVariables(
        {
            $id: Now.ID['set_variables_with_datapills'],
            annotation: 'Set variables from trigger and action outputs',
        },
        _params.flowVariables,
        {
            incidentPriority: _params.trigger.current.priority,
       // Trigger datapill
            assignedGroup:
 _params.trigger.current.assignment_group,    // Trigger datapill
            threshold: lookupResult.Record.value,
       // Action output datapill
            actionResult: _params.flowVariables.status,
       // Flow variable datapill
        }
    )
}
```

This exampe assign subflow Outputs using flow variables.

```
// Flow body showing assignSubflowOutputs usage with flow variables
import { FlowVariables } from '@servicenow/sdk/automation'

const flowVars = FlowVariables({
    userEmail: StringColumn({ label: 'User Email' }),
    userName: StringColumn({ label: 'User Name' }),
    isApproved: BooleanColumn({ label: 'Is Approved' }),
})

(_params) => {
    // Call a subflow that returns user information
    const subflowResult = wfa.subflow(
        userInfoSubflow,
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
        { $id: Now.ID['get_user_info'], annotation: 'Get user
information' },
        { userId: _params.trigger.current.caller_id }
    )

    // Assign subflow outputs to flow variables in one operation
    wfa.flow_logic.assignSubflowOutputs(
        { $id: Now.ID['assign_outputs'], annotation: 'Assign user info to flow
variables' },
        flowVars,
        {
            userEmail: subflowResult.email,
            userName: subflowResult.name,
            isApproved: subflowResult.approved,
        }
    )

    // Now flow variables can be used throughout the flow
    // Access via _params.flowVariables.userEmail, etc.
}
```

wfa.dataPill function

Reference a specific runtime data pill value from an action or flow logic input.

Add `wfa.dataPill` functions to the input parameters of a `action` or `flow_logic` function. Define a constant for each data pill you want to use. Typically, this constant stores the output of a wfa.action or wfa.flow_logic function.

Properties

Name	Type	Description
expression	String	Required. Dot notation reference to the runtime data that you want use as an input value. Commonly used runtime values include trigger data, action outputs, and subflow outputs. **Trigger record data** Field values from a trigger record. For example, `_params.trigger.current.fieldName`. **Trigger record table** Table name of the trigger record. For example, `_params.trigger.table_name`. **Action output values** Output values from previously run actions. For example, `actionResult.fieldName` **Subflow output values** Output values from previously run subflows. For example, `subflowResult.fieldName`
Type	String	Required. Data type of the data such as string, boolean, integer, or reference. For a list of available field data types, see Field types reference.

This example uses an unbound email record to create either an incident or a task based on the values of the inbound email. This example uses the params object to refer to data in the trigger.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

This example also defines the constants sender, senderManager, p3Incident, and attachments to store the output of actions and flow logic for use in later wfa.dataPill functions.

```
import { action, Flow, wfa, trigger } from '@servicenow/sdk/automation'

export const emailIncidentTaskFlow = Flow(
    {
        $id: Now.ID['email_incident_task_flow'],
        name: 'Email Incident/Task Creation Flow',
        description: 'Creates incidents or tasks based on email content with "raise
 incident or task" in subject',
    },
    wfa.trigger(
        trigger.application.inboundEmail,
        { $id: Now.ID['inbound_email_trigger'] },
        {
            email_conditions: 'subjectLIKEraise incident or task',
            target_table: 'incident',
        }
    ),
    (params) => {
        wfa.action(
            action.core.log,
            { $id: Now.ID['log_email_received'] },
            {
                log_level: 'info',
                log_message: `Email received from:
 ${wfa.dataPill(params.trigger.from_address, 'string')}, Subject:
 ${wfa.dataPill(params.trigger.subject, 'string')}`,
            }
        )

        const sender = wfa.action(
            action.core.lookUpRecord,
            { $id: Now.ID['lookup_sender'] },
            {
                table: 'sys_user',
                conditions:
 `email=${wfa.dataPill(params.trigger.from_address, 'string')}`,
                sort_type: 'sort_asc',

 if_multiple_records_are_found_action: 'use_first_record',
            }
        )

        // Internal P1 path — two tasks
        wfa.flowLogic.if(
            {
                $id: Now.ID['check_internal_p1'],
                condition:
 `
${wfa.dataPill(sender.Record.email, 'string')}LIKEservicenow^${wfa.d
ataPill(params.trigger.subject, 'string')}LIKEP1`,
                annotation: 'Internal sender with P1 subject',
            },
            () => {
                const senderManager = wfa.action(
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
                    action.core.lookUpRecord,
                    { $id: Now.ID['lookup_sender_manager'] },
                    {
                        table: 'sys_user',
                        conditions:
`sys_id=${wfa.dataPill(sender.Record.manager, 'reference')}`,
                        sort_type: 'sort_asc',

if_multiple_records_are_found_action: 'use_first_record',
                    }
                )

                wfa.action(
                    action.core.createTask,
                    { $id: Now.ID['task_for_manager'] },
                    {
                        task_table: 'incident',
                        field_values: TemplateValue({
                            priority: 1,
                            assigned_to:
wfa.dataPill(senderManager.Record.sys_id, 'reference'),
                            short_description:
wfa.dataPill(params.trigger.inbound_email.body, 'reference'),
                            urgency: 1,
                            impact: 1,
                        }),
                    }
                )

                wfa.action(
                    action.core.createTask,
                    { $id: Now.ID['task_for_sender'] },
                    {
                        task_table: 'incident',
                        field_values: TemplateValue({
                            priority: 1,
                            assigned_to:
wfa.dataPill(sender.Record.sys_id, 'reference'),
                            short_description:
wfa.dataPill(params.trigger.subject, 'string'),
                            urgency: 1,
                            impact: 1,
                        }),
                    }
                )
            }
        )

        // External / non-P1 path — P3 incident + copy attachments
        wfa.flowLogic.else({ $id: Now.ID['create_p3_incident_branch'] },
() => {
            const p3Incident = wfa.action(
                action.core.createRecord,
                { $id: Now.ID['create_p3_incident'] },
                {
                    table_name: 'incident',
                    values: TemplateValue({
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
                        priority: 4,
                        short_description:
wfa.dataPill(params.trigger.subject, 'string'),
                        description: `From:
${wfa.dataPill(params.trigger.from_address, 'string')}\n\n
${wfa.dataPill(params.trigger.inbound_email.body, 'reference')}`,
                        contact_type: 'email',
                        caller_id:
wfa.dataPill(sender.Record.sys_id, 'reference'),
                    }),
                }
            )

            const attachments = wfa.action(
                action.core.getAttachmentsOnRecord,
                { $id: Now.ID['get_email_attachments'] },
                { source_record:
wfa.dataPill(params.trigger.inbound_email.sys_id, 'reference') }
            )

            wfa.flowLogic.forEach(
                wfa.dataPill(attachments.parameter, 'records'),
                { $id: Now.ID['copy_attachments_loop'] },
                () => {
                    wfa.action(
                        action.core.copyAttachment,
                        { $id: Now.ID['copy_attachment'] },
                        {
                            target_record:
wfa.dataPill(p3Incident.record, 'reference'),
                            attachment_record:
wfa.dataPill(attachments.parameter, 'records'),
                            table: 'incident',
                        }
                    )
                }
            )

            wfa.action(
                action.core.sendEmail,
                { $id: Now.ID['confirm_p3_email'] },
                {
                    table_name: 'incident',
                    ah_to:
wfa.dataPill(params.trigger.from_address, 'string'),
                    ah_subject: `Incident Created:
${wfa.dataPill(params.trigger.subject, 'string')}`,
                    ah_body: `Your request has been received
and a P3 incident has been created. Our team will be in touch
shortly.`,
                    record:
wfa.dataPill(p3Incident.record, 'reference'),
                }
            )
        })
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    }
)
```

wfa.subflow function

Run a specific subflow from a parent flow or subflow.

Create a subflow object to run a reusable sequence of actions and flow logic. To create a subflow object, see Subflow object.

Call a specific subflow by adding a *`wfa.subflow`* function to the *`Flow Body`* function of a *`Flow`* or *`Subflow`* object.

Properties

Name	Type	Description
subflow	String	Name of the specific subflow to run. All subflow names use a dot notation that starts with `action.core`. For example, `action.core.lookUpRecord`, `action.core.lookUpRecords`, `action.core.createRecord`, or `action.core.updateRecord`.
configuration	Object	Required. An object containing the metadata configuration properties for the Fluent object or function.
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
annotation	String	Text description of the subflow to display in the editor.
inputs	Object	An object containing any subflow input parameters required by the subflow such as `subject` or `body`.
waitForCompletion	Boolean	Option to pause the parent calling flow or subflow until the subflow completes running. Valid values: true, false Default: false

This example creates a subflow object that can be called by the constant name notifyUserSubflow.

```
// File: fluent/flows/notify-user-subflow.now.ts
import { Subflow, wfa, action } from "@servicenow/sdk/automation";
import { StringColumn, BooleanColumn } from "@servicenow/sdk/core";

export const notifyUserSubflow = Subflow(
  {
    $id: Now.ID["notify_user_subflow"],
    name: "Notify User Subflow",
    runAs: "system",
    inputs: {
      userId: StringColumn({ label: "User Sys ID", mandatory:
 true }),
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
      message: StringColumn({ label: "Message", mandatory: true })
    },
    outputs: {
      notified: BooleanColumn({ label: "Notified", mandatory:
 true })
    }
  },
  params => {
    wfa.action(
      action.core.log,
      { $id: Now.ID["log_notify"], annotation: "Log notification" },
      {
        log_level: "info",
        log_message: `Notifying user:
 ${wfa.dataPill(params.inputs.userId, "string")} —
 ${wfa.dataPill(params.inputs.message, "string")}`
      }
    );

    wfa.flowLogic.assignSubflowOutputs(
      { $id: Now.ID["assign_notify_outputs"] },
      params.outputs,
      { notified: true }
    );
  }
);
```

This example calls the notifyUserSubflow subflow that was created by the previous Subflow object.

```
// File: fluent/flows/incident-created-flow.now.ts
import { Flow, wfa, trigger, action } from "@servicenow/sdk/automation";
import { notifyUserSubflow } from "./notify-user-subflow.now";

Flow(
  {
    $id: Now.ID["incident_created_flow"],
    name: "Incident Created Flow",
    runAs: "system"
  },
  wfa.trigger(
    trigger.record.created,
    { $id: Now.ID["trg_incident_created"] },
    {
      table: "incident",
      condition: "priority=1",
      run_flow_in: "background",
      run_on_extended: "false",
      run_when_user_list: [],
      run_when_setting: "both",
      run_when_user_setting: "any"
    }
  ),
  _params => {
    const result = wfa.subflow(
      notifyUserSubflow,
      {
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
        $id: Now.ID["notify_user_instance"],
        annotation: "Notify assigned user"
      },
      {
        userId: wfa.dataPill(
          _params.trigger.current.assigned_to.sys_id,
          "string"
        ),
        message: "A P1 incident has been assigned to you.",
        waitForCompletion: true
      }
    );

    wfa.action(
      action.core.log,
      { $id: Now.ID["log_notify_result"] },
      {
        log_level: "info",
        log_message: `Subflow notified:
 ${wfa.dataPill(result.notified, "boolean")}`
      }
    );
  }
);
```

Import Sets API - ServiceNow Fluent

The Import Sets API defines transform maps [sys_transform_map] that specify how to transform and map data from the import set staging table to target tables.

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

Every import operation to a production table requires at least one transform map associated with an import set. The transform map specifies the data relationships between the import set and the target table.

To create an import set in ServiceNow Fluent code, you must define the required metadata in the following order:

1. Define the staging table [sys_db_object] using the Table API. The table must extend the Import Set Row [sys_import_set_row] table. For more information, see Table API - ServiceNow Fluent.

   The staging table defines all columns that receive imported data.

2. Define the data source [sys_data_source] using the Record API. The data source must reference the staging table from its *`import_set_table_name`* property. For more information, see Record API - ServiceNow Fluent.

   The data source defines the connection to external systems (files, databases, APIs) and how to load data into import staging tables. For more information, see Data sources.

3. Define the transform map [sys_transform_map] using the Import Sets API. The transform map must reference the staging table from its *`sourceTable`* property.

Important: The string `NULL` is a reserved word. It shouldn't be used as a field value in import set transform maps or anywhere in the **First name** or **Last name** fields. The reserved word is `NULL` in all capital letters. A field with the value `Null` or `null`, for example, is acceptable. `NULL` should be used only to clear out a particular field.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For general information about import sets, see Import sets.

Related topics

ServiceNow Fluent

ImportSet object

Create a transform map [sys_transform_map] to define the relationships between fields in an import set table and fields in an existing table.

For general information about creating transform maps, see Create a transform map.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
name	String	Required. An internal name for the transform map.
targetTable	String	Required. The name of the table in which you want the transformed data to be inserted. You can specify only tables within the application scope or the global scope, or tables that grant write access to other applications.
sourceTable	String	Required. The name of the table containing the raw import set data to transform. An import table is any table that extends the Import Set Row [sys_import_set_row] table. You can specify only tables within the application scope. **Note:** The value of this property must match the value of the *`name`* property in the staging table definition and the *`import_set_table_name`* property in the data source definition.
order	Number	The execution order in which to apply transform maps if more than one map fits the conditions. Default: 100
active	Boolean	Flag that indicates whether the transform map is active. Default: false
runBusinessRules	Boolean	Flag that indicates whether to run business rules, workflows, approval engines, auditing, and field normalization while the transformation inserts or updates data into the target table. If false, *`GlideRecord.setWorkflow()`* runs with a value of false. Default: true
enforceMandatoryFields	String	An option to enforce mandatory fields on the target table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Valid values: • no: Don't enforce mandatory fields. • onlyMappedFields: Enforce mandatory mapped fields only. • allFields: Enforce all mandatory fields. Default: no
copyEmptyFields	Boolean	Flag that indicates whether to copy empty fields from source and override existing target field values. Default: false
createOnEmptyCoalesce	Boolean	Flag that indicates whether to create a record when coalesce fields are empty, instead of ignoring the record or overwriting an existing record. If `createOnEmptyCoalesce` is true for any field in the record, the record is coalesced. Default: false
runScript	Boolean	Flag that indicates whether to execute the transform map script. The system runs the transform map script in addition to any field maps. Default: false
script	Script	A script that transforms field values in the source table to the target table. This property supports a function from a JavaScript module, a reference to another file in the application that contains a script, or inline JavaScript. Expects a function of type `(source, target, map, log, isUpdate) => void`. Format: • For functions, use the name of a function, function expression, or default function exported from a JavaScript module and import it into the `.now.ts` file. For information about JavaScript modules, see JavaScript modules and third-party libraries. • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide an inline script, use string literals or template literals for multiple lines of code: `'Script' or` `` `Script` ``.
fields	Object	Key and value pairs of field mappings where the key is target field name and the value is the source field name or the fields

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		object. Each target field name (key) must be unique within the fields object to avoid mapping conflicts. Format: • To map a target field name to a source field object, see fields object. • To map a target field name to a source field name, use the following format: `fields: { targetFieldName: 'sourceFieldName', ... }`
scripts	Array	A list of transform scripts for additional processing. For more information, see scripts array.

```
import '@servicenow/sdk/global'
import { Table, Record, ImportSet } from '@servicenow/sdk/core'

// STEP 1: Create Staging Table Definition (REQUIRED - MUST BE FIRST)
// This creates the actual table structure to hold imported data
export const userStagingTable = Table({
    $id: Now.ID['user-staging-table'],
    name: 'u_user_import_staging',
    label: 'User Import Staging',
    extends: 'sys_import_set_row', // All staging tables extend this
    columns: [
        {
            name: 'u_email_address',
            type: 'email',
            max_length: 100,
            label: 'Email Address'
        },
        {
            name: 'u_full_name',
            type: 'string',
            max_length: 100,
            label: 'Full Name'
        },
        {
            name: 'u_username',
            type: 'string',
            max_length: 40,
            label: 'Username'
        },
    ]
})

// STEP 2: Create Data Source (REQUIRED - MUST BE SECOND)
// The data source defines HOW to get data from external systems
export const userDataSource = Record({
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    $id: Now.ID['user-csv-datasource'],
    table: 'sys_data_source',
    data: {
        name: 'User CSV Data Source',
        type: 'File',
        format: 'CSV',
        file_retrieval_method: 'Attachment',
        csv_delimiter: ',',
        header_row: 1,
        // CRITICAL: This must match the table name from STEP 1
        import_set_table_name: 'u_user_import_staging',
        import_set_table_label: 'User Import Staging',
        batch_size: 500,
        active: true,
    },
})

// STEP 3: Create Import Set (Transform Map) (REQUIRED - MUST BE THIRD)
// The import set defines HOW to transform data from staging to target table
export const userImportSet = ImportSet({
    $id: Now.ID['user-import-transform'],
    name: 'User Import Transform',
    targetTable: 'sys_user',
    // CRITICAL: This must match import_set_table_name in Data Source
    sourceTable: 'u_user_import_staging',
    active: true,
    runBusinessRules: true,
    fields: {
        email: {
            sourceField: 'u_email_address',
            coalesce: true,
        },
        name: 'u_full_name',
        user_name: 'u_username',
    }
})
```

fields object

Define field mappings [sys_transform_entry] from the source fields of an import set to the fields of the target table.

Use the *`fields`* object within the *`ImportSet`* object. In the *`fields`* object, each key for the target field name must be unique to avoid mapping conflicts.

For general information about creating field maps, see Create a field map.

Properties

Name	Type	Description
sourceField	String	The name of the source field from the import table to be transformed. This property is required unless you use the `sourceScript` property or a coalesce-only configuration. Field on the source table to be transformed. If the *`sourceTable`* contains only raw data, the value can be empty.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
choiceAction	String	The action to take if the import set contains a reference or choice value other than those available. This property applies if the target field is a choice list or reference field. Valid values: • create: Create a choice or record in the reference table. • ignore: Ignore the new value from the source table. • reject: Skip the entire row (record) containing the new value and continue to the next row.
sourceScript	Script	A script that transforms field values in the source table to the target table. This property supports a function from a JavaScript module, a reference to another file in the application that contains a script, or inline JavaScript. Expects a function of type `(source) => any`. Format: • For functions, use the name of a function, function expression, or default function exported from a JavaScript module and import it into the `.now.ts` file. For information about JavaScript modules, see JavaScript modules and third-party libraries. • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide an inline script, use string literals or template literals for multiple lines of code: `'Script' or` `` `Script` ``.
useSourceScript	Boolean	Flag that indicates whether to use the source script instead of the source field. Default: false
dateFormat	String	The format for transformations when the target field is a Date or Date/Time field. **Note:** To learn more about allowable date formats, see Global date and time field format. Format: `'dd-MM-yyyy'`, `'yyyy-MM-dd'`, `'yyyy-dd-MM'`, `'MM-dd-yyyy HH:mm:ss z'`, `'yyyy-MM-dd HH:mm:ss'`, `'HH:mm:ss'`, `'MM-dd-yyyy HH:mm:ss'`, `'dd-MM-yyyy HH:mm:ss z'`, `'MM-dd-yyyy'`, or `'dd-MM-yyyy HH:mm:ss'`
referenceValueField	String	The reference value field when the target field is a reference field. The transform map needs a way to match incoming source values to existing records in the reference field's source table. Because most imports don't provide a 32-

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		character sys_id value, you must specify a column from the reference field's source table that contains values that match the incoming source values. For more information, see Create a field map.
coalesce	Boolean	Flag that indicates whether the field is used for record matching. Configuring a target field to coalesce causes the import set to treat the field as a unique key. For more information, see Create a field map. Default: false
coalesceCaseSensitive	Boolean	Flag that indicates whether to make case-sensitive coalesce values result in the creation of new records. By default, coalesce fields are used in a case-insensitive lookup for existing records. Case-insensitive records update existing records only and don't cause the creation of new records. Default: false
coalesceEmptyFields	Boolean	Flag that indicates whether to match an empty source field value to an empty target field value. The `coalesce` property must be set to true for this property to apply. For example, the User transform map coalesces on the email field. If this property is set to true, a source record containing an empty email address coalesces to a target record containing an empty email address. Default: false

```
fields: {
        email: {
            sourceField: 'email_address',
            coalesce: true,
            useSourceScript: true,
            sourceScript: `answer = (function
 transformEntry(source) {
                return source.email_address.toLowerCase().trim();
            })(source);`
        },
        department: {
            sourceField: 'dept_code',
            choiceAction: 'create'
        }
}
```

scripts array

Define transform scripts [sys_transform_script] that run at different stages of the import process.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
active	Boolean	Flag that indicates whether the script is active and executes. Default: true
order	Number	The execution order in which scripts run if more than one script fits the conditions. Default: 100
when	String	The stage of the import process in which to execute the script. For more information, see Map with transformation event scripts. Valid values: onBefore, onAfter, onReject, onStart, onForeignInsert, onComplete, onChoiceCreate Default: onAfter
script	Script	A script that modifies the transformation behavior at the stage specified with the *when* property. This property supports a function from a JavaScript module, a reference to another file in the application that contains a script, or inline JavaScript. Expects a function of type `(source, map, log, target) => void`. Format: • For functions, use the name of a function, function expression, or default function exported from a JavaScript module and import it into the `.now.ts` file. For information about JavaScript modules, see JavaScript modules and third-party libraries. • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide an inline script, use string literals or template literals for multiple lines of code: `'Script' or` `` `Script` ``.

```
scripts: [
        {
            $id: Now.ID['validate-email'],
            active: true,
            order: 100,
            when: 'onBefore',
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
            script: `(function runTransformScript(source, map,
log, target) {
                if (!source.email_address ||
source.email_address.indexOf('@') === -1) {
                    log.error('Invalid email address: ' +
source.email_address);
                    return;
                }
            })(source, map, log, target);`
        },
        {
            $id: Now.ID['validate-user-script'],
            active: true,
            order: 100,
            when: 'onBefore',
            // Using imported function for lifecycle script
            script: validateUserData
        }
    ]
```

List API - ServiceNow Fluent

The List API defines list views [sys_ui_list] for tables.

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

For general information about lists, see ServiceNow AI Platform® list administration.

Related topics

ServiceNow Fluent

List object

Configure lists [sys_ui_list] and their views.

Properties

Name	Type	Description
table	String	Required. The name of the table to which the list applies.
view	Reference or String	Required. The variable identifier or name of the UI view [sys_ui_view] which applies, or the default view. To define a UI view, use the Record API - ServiceNow Fluent. To use the default view (`default_view`), you must import it: `import { default_view } from '@servicenow/sdk/core'`
columns	Array	Required. A list of columns in the table to display in the list, specified using the column name and position in the list.
$meta	Object	Metadata for the application metadata.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		With the *installMethod* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }` Valid values for *installMethod*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
import { List } from "@servicenow/sdk/core";

List({
    $id: Now.ID["app_task_view_list"],
    table: "cmdb_ci_server",
    view: app_task_view,
    columns: [
        { element: "name", position: 0 },
        { element: "business_unit", position: 1 },
        { element: "vendor", position: 2 },
        { element: "cpu_type", position: 3 },
    ],
});
```

The UI view definition referenced is defined using the *Record* object:

```
import { Record } from "@servicenow/sdk/core";

const app_task_view = Record({
   $id: Now.ID['app_task_view'],
   table: 'sys_ui_view',
   data: {
        name: 'app_task_view',
        title: 'app_task_view'
   }
})
```

Property API - ServiceNow Fluent

The Property API defines system properties [sys_properties] that control instance behavior.

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

For general information about system properties, see Add a system property.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Related topics

ServiceNow Fluent

Property object

Add a system property [sys_properties] for configuring an aspect of an application.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
name	String	Required. The name of the property beginning with the application scope in the following format: `<scope>.<name>`.
value	Any	A value for the property. The value must be the correct data type. All property values are stored as strings. When retrieving properties via the *`gs.getProperty()`* method, treat the results as strings. For example, a true|false property returns 'true' or 'false' (strings), not the Boolean equivalent.
type	String	A data type for the property value. Valid values: string, integer, boolean, choicelist, color, date_format, image, password, password2, short_string, time_format, timezone, uploaded_image
description	String	A description of what the property does.
choices	Array	A comma-separated list of choice values. This property only applies if the *`type`* property is set to `choicelist`. If you need a different choice label and value, use an equal sign (=) to separate the label from the value. For example, `['Blue=0000FF', 'Red=FF0000', 'Green=00FF00']` displays **Blue**, **Red**, and **Green** in the list, and saves the corresponding hex value in the property value field.
roles	Object	The variable identifiers of *`Role`* objects or names of roles that have read or write access to the property. For example: `roles: {` `  read: [activity_admin, 'app_user'],` `  write: [admin]` `}`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		For more information, see Role API - ServiceNow Fluent.
ignoreCache	Boolean	Flag that indicates whether to cache flush when the value of the property is set. The system stores system property values in server-side caches to avoid querying the database for configuration settings. When you change a system property value, the system flushes the cache for the System Properties [sys_properties] table. Use this field to determine whether to flush this property's value from all other server-side caches. Valid values: • true: The system ignores flushing some server-side caches, thus flushing only the cache for the System Properties [sys_properties] table and preserving the prior property value in all other caches. This option avoids the performance cost of flushing all caches and retrieving new property values. Generally, you should only set this property to `true` when you have a system property that changes more frequently than once a month, and the property value is only stored in the System Properties [sys_properties] table table. • false: The system flushes all server-side caches and retrieves the current property value from the database. Set this property to `false` for all caches to have the current property value. Default: false
isPrivate	Boolean	Flag that indicates whether to exclude the property from being imported via update sets. Keeping system properties private helps prevent settings in one instance from overwriting values in another instance. For example, you might not want a system property in a development instance to use the same value as a production instance. Valid values: • true: The property isn't included in update sets. • false: The property is included in update sets. Default: false
$meta	Object	Metadata for the application metadata. With the *`installMethod`* property, you can map the application metadata to an output directory that loads only in specific circumstances.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		`$meta: { installMethod: 'String' }` Valid values for *`installMethod`*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
import { Property } from '@servicenow/sdk/core'

Property({
   $id: Now.ID['1234'],
   name: 'x_snc_app.some.new.prop',
   type: 'string',
   value: 'hello',
   description: 'A new property',
   roles: {
      read: ['admin'],
      write: [adminRole, managerRole],
   },
   ignoreCache: false,
   isPrivate: false,
})
```

The roles referenced are defined using the *`Role`* object:

```
import { Role } from "@servicenow/sdk/core";

const managerRole = Role({
   $id: Now.ID['manager_role'],
   name: 'x_snc_example.manager'
})

const adminRole = Role({
   $id: Now.ID['admin_role'],
   name: 'x_snc_example.admin',
   containsRoles: [managerRole]
})
```

Record API - ServiceNow Fluent

The Record API defines records in any table. Use the Record API to define application metadata that doesn't have a dedicated ServiceNow Fluent API.

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Related topics

ServiceNow Fluent

Record object

Add data to any table with a record.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
table	String	Required. The name of the table to which the record belongs.
data	Object	Fields and their values in the table. For example: `data: { state: 'Ready', task: 'Add demo data' }` To use text content from another file, refer to a file in the application using the `Now.include` syntax. For more information, see ServiceNow Fluent language constructs. `data: { script: Now.include('./script-file.js'), html: Now.include('./html-file.html'), css: Now.include('./css-file.css') }`
$meta	Object	Metadata for the application metadata. With the *`installMethod`* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }` Valid values for *`installMethod`*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

In this example, a record defining a menu category is added to the Menu Category [sys_app_category] table. The menu category style is defined in the `css-file.css` file.

```
import { Record } from "@servicenow/sdk/core";

export const appCategory = Record({
    table: 'sys_app_category',
    $id: Now.ID[9],
    data: {
        name: 'example',
        style: Now.include('./css-file.css'),
    },
})
```

In this example, a record defining an incident is added to the Incident [incident] table.

```
import { Record } from '@servicenow/sdk/core';

export const incident1 = Record({
  $id: Now.ID['incident-1'],
  table: 'incident',
  data: {
    active: 'true',
    approval: 'not requested',
    description: 'Unable to send or receive emails.',
    incidentState: '1',
    shortDescription: 'Email server is down.',
    subcategory: 'email',
    callerId: '77ad81767313130057546660c4cf6a7de',
  }
})
```

In this example, a record defining a server is added to the Server [cmdb_ci_server] table.

```
import { Record } from '@servicenow/sdk/core';

export const ciserver1 = Record({
  $id: Now.ID['cmdb-ci-server-1'],
  table: 'cmdb_ci_server',
  data: {
    assetTag: 'P1000199',
    attested: 'false',
    canPrint: 'false',
    company: 'e7c1f3d53790200044e0bfc8bcbe5deb',
    cost: '2160',
    costCc: 'USD',
    cpuSpeed: '633',
    cpuType: 'GenuineIntel',
    diskSpace: '100',
    manufacturer: 'b7e7d7d8c0a8016900a5d7f291acce5c',
    name: 'DatabaseServer1',
    os: 'Linux Red Hat',
    shortDescription: 'DB Server',
    subcategory: 'Computer',
  }
})
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Role API - ServiceNow Fluent

The Role API defines roles [sys_user_role] that grant specific permissions to users of an application.

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

For general information about user roles, see Managing roles.

Related topics

ServiceNow Fluent

Role object

Create a role [sys_user_role] to control access to applications and their features.

Properties

Name	Type	Description
name	String	A name for the role beginning with the application scope in the following format: `<scope>.<name>`.
assignableBy	String	Other roles that can assign this role to users.
canDelegate	Boolean	Flag that indicates if the role can be delegated to other users. For more information, see Delegating roles. Valid values: • true: The role can be delegated to other users. • false: The role can't be delegated to other users. Default: true
description	String	A description of what the role can access.
elevatedPrivilege	Boolean	Flag that indicates whether manually accepting the responsibility of using the role before you can access the features of the role is required. For more information about elevated privileges, see Elevated privilege roles. Valid values: • true: You must manually accept the responsibility of using the role before you can access its features. • false: You don't need to manually accept the responsibility of using the role to access its features. Default: false
grantable	Boolean	Flag that indicates whether the role can be granted independently. Valid values: • true: The role can be granted independently. • false: The role can't be granted independently.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Default: true
containsRoles	Array	The variable identifiers of other *Role* objects that this role contains.
scopedAdmin	Boolean	Flag that indicates whether the role is an Application Administrator role. For general information about application administration roles, see Application administration. Valid values: • true: The role is an Application Administrator. • false: The role isn't an Application Administrator. Default: false
$meta	Object	Metadata for the application metadata. With the *installMethod* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }` Valid values for *installMethod*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
import { Role } from "@servicenow/sdk/core";

const managerRole = Role({
   $id: Now.ID['manager_role'],
   name: 'x_snc_example.manager'
})

const adminRole = Role({
   $id: Now.ID['admin_role'],
   name: 'x_snc_example.admin',
   containsRoles: [managerRole]
})
```

Script Action API - ServiceNow Fluent

The Script Action API defines script actions [sysevent_script_action] that run when an event occurs.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

For general information about scheduled script executions, see Script actions.

Related topics

ServiceNow Fluent

ScriptAction object

Create a script action [sysevent_script_action] that performs a task when triggered by an event.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
name	String	Required. A unique name for the script action.
script	Script	Required. A server-side script that runs when triggered by an event. This property supports a function from a JavaScript module, a reference to another file in the application that contains a script, or inline JavaScript. Format: • For functions, use the name of a function, function expression, or default function exported from a JavaScript module and import it into the `.now.ts` file. For information about JavaScript modules, see JavaScript modules and third-party libraries. • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide an inline script, use string literals or template literals for multiple lines of code: `'Script' or` `` `Script` ``.
eventName	String	Required. The event that triggers the script action to run. For information about creating events, see Create an event.
active	Boolean	Flag that indicates whether the script action is enabled. Valid values: • true: The script action executes when triggered by the event. • false: The script action doesn't execute. Default: false
description	String	A description of the functionality and purpose of the script action.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
order	Number	A number indicating the sequence in which the script action should run. If there are multiple script actions on a particular event, the script actions run in the order specified, from lowest to highest. Default: 100
conditionScript	String	A JavaScript conditional statement that specifies the fields and values that must be true for the script to run. **Note:** Don't use this property if you include the condition statement with the *`script`* property. Format: • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide an inline script, use string literals or template literals for multiple lines of code: `'Script' or` `` `Script` ``.
$meta	Object	Metadata for the application metadata. With the *`installMethod`* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }` Valid values for *`installMethod`*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
import { ScriptAction } from '@servicenow/sdk/core'
import { insertIncident } from '../server/scripts.js'

ScriptAction({
    $id: Now.ID['sample-script-action'],
    name: 'SampleScriptAction',
    active: true,
    description: 'Insert an incident',
    script: insertIncident,
    eventName: 'sample.event',
    order: 100,
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    conditionScript: "gs.hasRole('my_role')"
})
```

The *script* property refers to a function from the `scripts.js` module. For example:

```
import { GlideRecord } from '@servicenow/glide'

export const insertIncident = () => {
    var gr = new GlideRecord('incident')
    gr.initialize()
    gr.setValue('short_description', 'New incident from event')
    gr.insert()
}
```

Script Include API - ServiceNow Fluent

The Script Include API defines script includes [sys_script_include] that store JavaScript functions and classes for use by server-side scripts.

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

For new scripts, use JavaScript modules instead of script includes when possible to support code reuse and using third-party libraries within an application scope. For more information about JavaScript module support and limitations, see JavaScript modules and third-party libraries.

For general information about script includes, see Script includes.

Related topics

ServiceNow Fluent

ScriptInclude object

Create a script include [sys_script_include] to define a server-side script that runs when called from other scripts.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
name	String	Required. The name of the script include. If you define a class, the name must match the name of the class, prototype, and type. If you use a classless (on demand) script include, the name must match the function name.
script	Script	Required. A server-side script to call from other scripts. The script must define a single JavaScript class or a global function. The class or function name must match the *name* property. This property supports inline JavaScript or a reference to another file in the application that contains a script.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Format: • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide an inline script, use string literals or template literals for multiple lines of code: `'Script' or` `` `Script` ``.
apiName	String	An internal name for the script include, which is used to call the script include from out-of-scope applications. Default: <scope>.<name>
description	String	A description of the purpose and function of the script include.
clientCallable	Boolean	Flag that indicates whether client-side scripts can call the script include using GlideAjax. The script include is available to client scripts, list/report filters, reference qualifiers, or if specified as part of the URL. Client callable script includes are invoked from *`GlideAjax`* and require users to satisfy an ACL associated with the script include. Valid values: • true: The script include is available to client-side scripts. • false: The script include isn't available to client-side scripts. Default: false
mobileCallable	Boolean	Flag that indicates whether the script include is available to client scripts called from mobile devices. Valid values: • true: The script include is available to client scripts called from mobile devices. • false: The script include isn't available to client scripts called from mobile devices. Default: false
sandboxCallable	Boolean	Flag that indicates whether the script include is available to scripts invoked from the script sandbox, such as a query condition. **Important:** Script includes should only be made available to the script sandbox if necessary. Valid values:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		• true: The script include is available to scripts invoked from the script sandbox. • false: The script include isn't available to scripts invoked from the script sandbox. Default: false
callerAccess	String	An option for how cross-scope access to the script include is permitted. For more information, see Restricted caller access privilege settings. Valid values: • restriction: Calls to the script include must be manually approved. Access requests are tracked in the Restricted Caller Access table with a status of Requested. • tracking: Calls to the script include are automatically approved. Calls are tracked in the Restricted Caller Access table with a status of Allowed.
accessibleFrom	String	Specifies which applications can access the script include. Valid values: • public: All application scopes can call the script include. • package_private: The script include can only be called from the application scope that it's within. Default: package_private
active	Boolean	Flag that indicates whether the script include is enabled. Valid values: • true: The script include is callable. • false: The script include isn't callable. Default: true
protectionPolicy	String	A policy that determines whether someone can view or edit the script include after the application is installed on their instance. If undefined, other application developers can customize the script include. Valid values: • read: Allows anyone to read values from this downloaded or installed script include. No one can change script values on the instance on which they download or install the script include. • protected: Provides intellectual property protection for application developers. Customers who download the script

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		include cannot see the contents of the script field. The script is encrypted in memory to prevent unauthorized users from seeing it in plain text.
$meta	Object	Metadata for the application metadata. With the *installMethod* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }` Valid values for *installMethod*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

In the following example, the script include uses a script from the `SampleClass.server.js` file.

```
import { ScriptInclude } from '@servicenow/sdk/core';

ScriptInclude({
    $id: Now.ID['sample-script-include'],
    name: 'SampleScriptInclude',
    script: Now.include("./SampleClass.server.js"),
    description: 'some description',
    apiName: 'x_scope.SampleScriptInclude',
    callerAccess: 'tracking',
    clientCallable: true,
    mobileCallable: true,
    sandboxCallable: true,
    accessibleFrom: 'public',
    active: true,
})
```

Scripted REST API - ServiceNow Fluent

The Scripted REST API defines the endpoints, query parameters, and headers for a scripted REST service [sys_ws_definition].

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

For general information about scripted REST services, see Scripted REST APIs.

Related topics

ServiceNow Fluent

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

RestApi object

Create a scripted REST API [sys_ws_definition] to define web service endpoints.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
name	String	Required. The name of the API, which is used in the API documentation.
serviceId	String	Required. The API identifier used to distinguish this API in URI paths. It must be unique within the API namespace.
active	Boolean	Flag that indicates whether the API can serve requests. Valid values: • true: The API can serve requests. • false: The API can't serve requests. Default: true
shortDescription	String	A brief description of the API, which is used in the API documentation.
consumes	String	A list of media types that resources of the API can consume. Default: application/json,application/xml,text/xml
docLink	String	A URL that links to static documentation about the API.
enforceAcl	Array	A list of variable identifiers of *ACL* objects or sys_ids of ACLs to enforce when accessing resources [sys_security_acl]. For more information, see Access Control List API - ServiceNow Fluent. To not enforce ACLs, set this property to an empty array ([ ]). Default: Scripted REST External Default
produces	String	A list of media types that resources of the API can produce. Default: application/json,application/xml,text/xml
routes	Array	The resources [sys_ws_operation] for the API. For more information, see routes object.
policy	String	The policy for how application files are protected when downloaded or installed.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Valid values: • read: Files are viewable only. • protected: Users with password permissions can edit the files.
versions	Array	A list of versions [sys_ws_version] for the API. For more information, see versions object. Specifying versions allows you to manage different versions of an API and their statuses, such as whether they are active, the default version, or deprecated.
$meta	Object	Metadata for the application metadata. With the *installMethod* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }` Valid values for *installMethod*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
import { RestApi } from '@servicenow/sdk/core'
import { process } from '../server/handler.js'

RestApi({
    $id: Now.ID['rest1'],
    name: 'customAPI',
    serviceId: 'custom_api',
    consumes: 'application/json',
    routes: [
        {
            $id: Now.ID['route1'],
            path: '/home/{id}',
            script: process,
            parameters: [{ $id: Now.ID['param1'],
 name: 'n_param' }],
            headers: [{ $id: Now.ID['header1'],  name: 'n_token' }],
            enforceAcl: [acl],
            version: 1,
        },
    ],
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    enforceAcl: [acl],
    versions: [
        {
            $id: Now.ID['v1'],
            version: 1,
        },
    ],
})
```

The ACL referenced is defined using the *ACL* object:

```
import { Acl } from "@servicenow/sdk/core";

const acl = Acl({
    name: 'My random ACL',
    type: 'rest_endpoint',
    script: `answer = (Math.random() > 0.5)`,
    active: true,
    adminOverrides: false,
    operations: ['execute'],
})
```

routes object

Create a scripted REST resource [sys_ws_operation] to define the HTTP method, the processing script, and to override settings from the parent service.

Use the *routes* object within the *RestApi* object.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
name	String	The name of the API resource, which is used in the API documentation. Default: the value of the *path* property
script	Script	Required. The custom script defines how the operation parses and responds to requests. This property supports a function from a JavaScript module, a reference to another file in the application that contains a script, or inline JavaScript. Format: • For functions, use the name of a function, function expression, or default function exported from a JavaScript module and import it into the `.now.ts` file. For information about JavaScript modules, see JavaScript modules and third-party libraries.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		• To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide an inline script, use string literals or template literals for multiple lines of code: `'Script' or `Script``.
parameters	Array	A list of query parameters [sys_ws_query_parameter] for the route. For more information, see parameters and headers objects.
headers	Array	A list of headers [sys_ws_header] for the route. For more information, see parameters and headers objects.
active	Boolean	Flag that indicates whether the resource is used. Valid values: • true: The resource is used. • false: The resource isn't used. Default: true
path	String	The path of the resource relative to the base API path. The relative URI can contain path parameters such as `'/abc/{id}'`. Default: /
shortDescription	String	A brief description of the resource, which is used in the API documentation.
consumes	String	A list of media types that the resource can consume. This property can be overridden with the PUT, PATCH, or POST methods. Default: The value of the *`consumes`* property in the *`RestApi`* object
enforceAcl	Array	A list of variable identifiers of *ACL* objects or sys_ids of ACLs to enforce when accessing resources [sys_security_acl]. For more information, see Access Control List API - ServiceNow Fluent. To not enforce ACLs, set this property to an empty array ([ ]). Default: Scripted REST External Default
produces	String	A list of media types that the resource can produce.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Default: The value of the *produces* property in the *RestApi* object
requestExample	String	A valid sample request body payload for the resource, which is used in the API documentation.
method	String	The HTTP method that the resource implements. Valid values: GET, POST, PUT, PATCH, DELETE Default: GET
authentication	Boolean	Flag that indicates whether users must be authenticated to access the resource. Valid values: • true: Users must be authenticated to access the resource. • false: Authentication isn't required to access the resource. Default: true
authorization	Boolean	Flag that indicates whether ACLs are enforced when accessing the resource. Valid values: • true: ACLs are enforced when accessing the resource. • false: ACLs aren't enforced when accessing the resource. Default: true
internalRole	Boolean	Flag that indicates whether the route requires the snc_internal role. This property is supported only if the Explicit Roles plugin (com.glide.explicit_roles) is enabled. Valid values: • true: The route requires the snc_internal role. • false: The route doesn't require the snc_internal role. Default: true
policy	String	The policy for how application files are protected when downloaded or installed. Valid values:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		• read: Files are viewable only. • protected: Users with password permissions can edit the files.
version	Number	The version of the API. This property is required if the *versions* property is used in the *RestApi* object. The version specified with this property is used to automatically generate a URI with a version, such as /api/management/v1/table/{tableName}. Version numbers identify the endpoint version that a URI accesses. By specifying a version number, you can test and deploy changes without impacting existing integrations.
$meta	Object	Metadata for the application metadata. With the *installMethod* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }` Valid values for *installMethod*: • demo: Outputs the application metadata to the metadata/unload.demo directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the metadata/unload directory to be installed only the first time an application is installed on an instance.

```
routes: [
   {
      $id: Now.ID['route1'],
      path: '/home/{id}',
      script: process,
      parameters: [{ $id: Now.ID['param1'],  name: 'n_param' }],
      headers: [{ $id: Now.ID['header1'],  name: 'n_token' }],
      enforceAcl: [acl],
      version: 1,
   },
],
```

parameters and headers objects

Create query parameters [sys_ws_query_parameter] and headers [sys_ws_header] for routes in a scripted REST API. Query parameters control what values a requesting user can pass in the request URI. Headers specify what the API accepts and can respond with.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Use the *parameters* and *headers* objects within the *routes* object.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
name	String	Required. The name of the parameter or header, which is used in the API documentation.
required	Boolean	Flag that indicates whether the parameter or header is required. Valid values: • true: The parameter or header is required. • false: The parameter or header isn't required. Default: false
exampleValue	String	An example of a valid value for the parameter or header, which is used in the API documentation.
shortDescription	String	A brief description of the parameter or header, which is used in the API documentation.
$meta	Object	Metadata for the application metadata. With the *installMethod* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }` Valid values for *installMethod*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
parameters: [{ $id: Now.ID['param1'],  name: 'n_param' }],
headers: [{ $id: Now.ID['header1'],  name: 'n_token' }],
```

versions object

Create versions for a scripted REST API [sys_ws_version] to define web service endpoints.

Use the *versions* object within the *RestApi* object.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
version	Number	Required. A version of the REST API.
active	Boolean	Flag that indicates whether the version of the REST API can serve requests. Valid values: • true: The version of the API can serve requests. • false: The version of the API can't serve requests. Default: true
deprecated	Boolean	Flag that indicates whether the version of the REST API is deprecated. Resources belonging to deprecated versions can serve requests, but are identified as deprecated in documentation. Valid values: • true: The version of the API is identified as deprecated. • false: The version of the API isn't identified as deprecated. Default: false
shortDescription	String	A brief description of the version of the REST API, which appears in the API documentation.
isDefault	Boolean	Flag that indicates whether the version of the REST API is the default version. Clients can access the default version using either the versioned or non-versioned URI path. Valid values: • true: The version of the API is the default version. • false: The version of the API isn't the default version. Default: false
$meta	Object	Metadata for the application metadata. With the *`installMethod`* property, you can map the application metadata to an output directory that loads only in specific circumstances.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		`$meta: { installMethod: 'String' }` Valid values for *`installMethod`*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
versions: [
 {
   $id: Now.ID['v1'],
   version: 1,
 },
],
```

Service Catalog API - ServiceNow Fluent

The Service Catalog API defines catalog items [sc_cat_item], record producers [sc_cat_item_producer], and related aspects of service catalogs.

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

For general information about service catalogs, see Service Catalog.

Related topics

ServiceNow Fluent

CatalogItem object

Create a catalog item [sc_cat_item] that users can request from a service catalog.

A catalog item must reference a flow, workflow, or execution plan that defines how the item request is fulfilled.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
name	String	Required. A name for the item that appears in the catalog.
active	Boolean	Flag that indicates whether the item is active and available to be ordered.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Default: true
availability	String	The type of device on which the item displays. Valid values: • desktopOnly: The item displays only on desktop devices. • mobileOnly: The item displays only on mobile devices. • both: The item displays on both desktop and mobile devices. Default: desktopOnly
checkedOut	Boolean	Flag that indicates whether the item is checked out for editing. Default: false
description	String	A detailed description of the item that displays in the catalog when a user selects the item or the associated **Preview** link. You can embed videos, images, links to internal knowledge base (KB) articles, and links to external sources of information and instruction documentation.
meta	Array	A list of metadata tags used to aid in searches related to the item. **Note:** The *meta* property is used only for Zing text indexing and search engine and not for AI Search.
model	Reference or String	The variable identifier or sys_id of a product model [cmdb_model] associated with the item. To define a product model, use the Record API - ServiceNow Fluent.
order	Number	The order in which the item appears within its category. Default: 0
owner	Reference or String	The variable identifier or sys_id of a user [sys_user] who owns the item and has edit access to it. To define a user, use the Record API - ServiceNow Fluent.
roles	Array	A list of variable identifiers of *Role* objects or sys_ids of roles [sys_user_role] that can access the item. For more information, see Role API - ServiceNow Fluent.
shortDescription	String	A short description of the item that appears on the catalog home page, search results, and the title bar of the order form.
showVariableHelpOnLoad	Boolean	Flag that indicates whether the item displays variable help text by default. Default: false

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
startClosed	Boolean	Flag that indicates whether to start the item in a collapsed state. Default: false
state	String	The publication state of the item, such as draft or published.
variables	Object	The variable definitions for the item that provide options for requesting it. Each variable type has a specific function. For example: `variables: { laptopType: SelectBoxVariable({ question: "Laptop Type", choices: { standard: { label: "Standard Laptop", sequence: 1 }, developer: { label: "Developer Workstation", sequence: 2 } }, mandatory: true, order: 100 }), justification: MultiLineTextVariable({ question: "Business Justification", mandatory: true, order: 200 }) }` For general information about catalog variables, see Service catalog variables.
version	Number	A version of the item. Default: 1
image	String	Deprecated. An image for the item.
icon	String	An image file that displays as an icon beside the item name in the catalog. Use a 27x27 pixel image. If no image is uploaded, the default icon appears.
picture	String	An image file to display as a picture of the item.
mobilePicture	String	An image to display as a picture of the item on mobile devices. This property applies only if the value of the *mobilePictureType* is `mobilePicture`.
mobilePictureType	String	The type of picture to display for the item on mobile devices.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Valid values: • desktopPicture: Use the image from the *picture* property. • mobilePicture: Use the image from the *mobilePicture* property. • noPicture: The item doesn't display a picture on mobile devices. Default: desktopPicture
hideAddToCart	Boolean	Flag that indicates whether to hide the **Add to Cart** button. Default: false
hideAddToWishlist	Boolean	Flag that indicates whether to hide the **Add to Wishlist** button. Default: false
hideDeliveryTime	Boolean	Flag that indicates whether to hide the delivery time. Default: false
hideQuantitySelector	Boolean	Flag that indicates whether to hide the **Quantity** field. Default: false
hideSaveAsDraft	Boolean	Flag that indicates whether to hide the **Save as Draft** button. Default: false
hideSP	Boolean	Flag that indicates whether to hide an item on Service Portal. Default: false
mandatoryAttachment	Boolean	Flag that indicates whether to require adding an attachment to submit a request. Default: false
hideAttachment	Boolean	Flag that indicates whether to hide the attachment section and not support adding attachments. Default: false
assignedTopics	Array	A list of sys_ids of existing taxonomy topics that control the visibility of the item in the Employee Center portal. For more information, see Associate a catalog item with a taxonomy topic in Employee Center. **Note:** This property is available only when the Employee Experience Taxonomy plugin (sn_ect) is active.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
availableFor	Array	A list of sys_ids of user criteria [user_criteria] that define who can access the item.
notAvailableFor	Array	A list of sys_ids of user criteria [user_criteria] that define who can't access the item. This property overrides the *availableFor* property.
variableSets	Array	A list of variable identifiers of *VariableSet* objects or sys_ids of variable sets [item_option_new_set] to attach to the item. For more information, see VariableSet object. Format: `{ variableSet, order }`
flow	Reference or String	The variable identifier of a *Flow* object or sys_id of a flow [sys_hub_flow] that defines how the request is fulfilled. To define a flow, use the Flow API - ServiceNow Fluent. **Note:** You should use flows as the fulfillment method for catalog items. When *flow*, *workflow*, and *executionPlan* are all specified, the system uses the flow.
executionPlan	Reference or String	The variable identifier or sys_id of an execution plan [sc_cat_item_delivery_plan] that defines how the request is fulfilled. To define an execution plan, use the Record API - ServiceNow Fluent.
workflow	String	The sys_id of a legacy workflow [wf_workflow] that defines how the item request is fulfilled. Use the *flow* property for new implementations.
accessType	String	The user access required to request the item. Valid values: • restricted: Only users who have access to the item can request the item. • delegated: Users who don't have access to the item can request the item on behalf of someone else using the delegated request experience. For more information, see Delegated request experience. **Note:** This functionality is only applicable when the item has a Requested For variable. Default: restricted
location	Reference or String	The variable identifier or sys_id of a location [cmn_location] where the item is provided. To define a location, use the Record API - ServiceNow Fluent.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
vendor	Reference or String	The variable identifier or sys_id of a vendor associated with the item. To define a vendor, use the Record API - ServiceNow Fluent.
deliveryPlanScript	Script	Deprecated. Use the *executionPlan* property instead.
deliveryTime	Object	The estimated time to deliver the item. Format: `deliveryTime: { days: Number, hours: Number })`
entitlementScript	Script	A script that defines entitlement for the item.
makeItemNonConversational	Boolean	Flag that indicates whether to prevent the ability to request the item from a conversational experience such as Virtual Agent. If true, the item can be requested from a conversational experience. Default: false
visibleBundle	Boolean	Flag that indicates whether item is visible in saved bundles. Default: true
visibleGuide	Boolean	Flag that indicates whether item is visible in order guides. Default: true
visibleStandalone	Boolean	Flag that indicates whether the standalone view is visible. Default: true
fulfillmentAutomationLevel	String	The level of fulfillment automation for requests. Valid values: • unspecified: Automation level of item isn't specified. • manual: Catalog item isn't automated. • semiAutomated: Catalog item is semi-automated. • fullyAutomated: Catalog item is automated completely.
fulfillmentGroup	Reference or String	The variable identifier or sys_id of the group [sys_user_group] responsible for delivering the item. To define a user group, use the Record API - ServiceNow Fluent.
catalogs	Array	A list of sys_ids of the catalogs [sc_catalog] in which the item appears.
categories	Array	A list of sys_ids of the categories [sc_category] to which the item belongs. A catalog must be specified in the *catalogs* property before categories can be assigned. In the Service Portal, catalog searches find only items that are assigned to a category. In Employee Center, catalog searches find only items that are associated with a taxonomy topic.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		For more information, see Associate a catalog item with a taxonomy topic in Employee Center.
cost	Number	A number indicating the one-time cost of the item. Default: 0
displayPriceProperty	String	The system property that controls how the item price is displayed.
ignorePrice	Boolean	Flag that indicates whether to hide the item price in the cart and in the catalog listing. Default: true
mobileHidePrice	Boolean	Flag that indicates whether to hide the item price on mobile devices. Default: false
omitPrice	Boolean	Flag that indicates whether to omit the price entirely from all views of the item. Default: false
billable	Boolean	Flag that indicates whether the item is billable. Default: false
pricingDetails	Array	A list of pricing entries for the item. Valid values: • price: A one-time price for the item. • recurring_price: A price that recurs at a regular interval. When using this value, the *`recurringFrequency`* property is required. Format: `{ amount, currencyType, field }`
recurringFrequency	String	The time interval at which the recurring price repeats, such as monthly or yearly. This property is required when *`pricingDetails`* contains a `recurring_price` field entry.
requestMethod	String	The label for the submission button and the order submission experience. Valid values: • order: Displays an **Order Now** button. A confirmation dialog is shown and delivery information is editable. • request: Displays a **Request** button. A confirmation dialog is shown but delivery information isn't displayed. Use for

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		scenarios where delivery information is pre-determined, such as a code access request. • submit: Displays a **Submit** button. No confirmation dialog or delivery information is displayed. Use for scenarios where no further request information is required, such as a password reset. Default: order
customCart	Reference or String	The variable identifier or sys_id of a custom UI macro to use for cart rendering. To define a custom cart, use the Record API - ServiceNow Fluent.
useScLayout	Boolean	Flag that indicates whether the item uses the Service Catalog layout for display. Default: true
noCart	Boolean	Deprecated. Flag that indicates whether to hide the shopping cart. Default: false
noOrder	Boolean	Deprecated. Flag that indicates whether to hide the **Order** option. Default: false
noOrderNow	Boolean	Deprecated. Flag that indicates whether to hide the **Order Now** option. Default: false
noProceedCheckout	Boolean	Deprecated. Flag that indicates whether to hide the checkout process. Default: false
noQuantity	Boolean	Deprecated. Flag that indicates whether to hide the **Quantity** field. Default: false
noSearch	Boolean	Deprecated. Flag that indicates whether to hide search. Default: false
$meta	Object	Metadata for the application metadata. With the *installMethod* property, you can map the application metadata to an output directory that loads only in specific circumstances.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		`$meta: { installMethod: 'String' }` Valid values for *`installMethod`*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
import { CatalogItem } from "@servicenow/sdk/core";

export const softwareLicenseRequest = CatalogItem({
  $id: Now.ID["software_license_request"],
  name: "Software License Request",
  shortDescription: "Request a software license",

  catalogs: [serviceCatalog],
  categories: [softwareCategory],

  // Attach reusable variable sets
  variableSets: [
    { variableSet: contactInfoSet, order: 100 },
    { variableSet: approvalInfoSet, order: 200 }
  ],

  // Item-specific variables
  variables: {
    software_name: SingleLineTextVariable({
      question: "Software Name",
      mandatory: true,
      order: 100
    }),
    license_type: SelectBoxVariable({
      question: "License Type",
      choices: {
        individual: { label: "Individual", sequence: 1 },
        team: { label: "Team (5 seats)", sequence: 2 },
        enterprise: { label: "Enterprise (unlimited)", sequence: 3 }
      },
      mandatory: true,
      order: 200
    }),
    number_of_licenses: SingleLineTextVariable({
      question: "Number of Licenses",
      defaultValue: "1",
      order: 300
    }),
    justification: MultiLineTextVariable({
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
      question: "Business Justification",
      mandatory: true,
      order: 400
    })
  },

  // Pricing with recurring charges
  pricingDetails: [
    { amount: 0, currencyType: "USD", field: "price" },
    { amount: 99, currencyType: "USD", field: "recurring_price" }
  ],
  recurringFrequency: "monthly",

  flow: "523da512c611228900811a37c97c2014",
  deliveryTime: { days: 3, hours: 0 }
})
```

CatalogItemRecordProducer object

Create a record producer [sc_cat_item_producer] for users to create task-based records, such as incident or change request records, from the service catalog.

You can create a record producer for tables and database views that are in the same scope as the record producer. You can also create a record producer for tables that allow create access from applications in other scopes.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
table	Reference or String	Required. The variable identifier of a *`Table`* object or name of a table in which the record producer creates records, such as incident or change_request.
name	String	Required. A name for the item that appears in the catalog.
active	Boolean	Flag that indicates whether the item is active and available to be ordered. Default: true
availability	String	The type of device on which the item displays. Valid values: • desktopOnly: The item displays only on desktop devices. • mobileOnly: The item displays only on mobile devices. • both: The item displays on both desktop and mobile devices. Default: desktopOnly
checkedOut	Boolean	Flag that indicates whether the item is checked out for editing.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Default: false
description	String	A detailed description of the item that displays in the catalog when a user selects the item or the associated **Preview** link. You can embed videos, images, links to internal knowledge base (KB) articles, and links to external sources of information and instruction documentation.
meta	Array	A list of metadata tags used to aid in searches related to the item. **Note:** The `meta` property is used only for Zing text indexing and search engine and not for AI Search.
model	Reference or String	The variable identifier or sys_id of a product model [cmdb_model] associated with the item. To define a product model, use the Record API - ServiceNow Fluent.
order	Number	The order in which the item appears within its category. Default: 0
owner	Reference or String	The variable identifier or sys_id of a user [sys_user] who owns the item and has edit access to it. To define a user, use the Record API - ServiceNow Fluent.
roles	Array	A list of variable identifiers of `Role` objects or sys_ids of roles [sys_user_role] that can access the item. For more information, see Role API - ServiceNow Fluent.
shortDescription	String	A short description of the item that appears on the catalog home page, search results, and the title bar of the order form.
showVariableHelpOnLoad	Boolean	Flag that indicates whether the item displays variable help text by default. Default: false
startClosed	Boolean	Flag that indicates whether to start the item in a collapsed state. Default: false
state	String	The publication state of the item, such as draft or published.
variables	Object	The variable definitions for the item that provide options for requesting it. Each variable type has a specific function. For example: `variables: {` `  laptopType: SelectBoxVariable({` `  question: "Laptop Type",` `  choices: {`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		<pre>standard: { label: "Standard Laptop", sequence: 1 }, developer: { label: "Developer Workstation", sequence: 2 } }, mandatory: true, order: 100 }), justification: MultiLineTextVariable({ question: "Business Justification", mandatory: true, order: 200 }) }</pre> For general information about catalog variables, see Service catalog variables.
version	Number	A version of the item. Default: 1
view	Reference or String	Required. The variable identifier or name of the UI view [sys_ui_view] which applies, or the default view. To define a UI view, use the Record API - ServiceNow Fluent. To use the default view (`default_view`), you must import it: <pre>import { default_view } from '@servicenow/sdk/core'</pre>
icon	String	An image file that displays as an icon beside the item name in the catalog. Use a 27x27 pixel image. If no image is uploaded, the default icon appears.
picture	String	An image file to display as a picture of the item.
mobilePicture	String	An image to display as a picture of the item on mobile devices. This property applies only if the value of the *`mobilePictureType`* is `mobilePicture`.
mobilePictureType	String	The type of picture to display for the item on mobile devices. Valid values: • desktopPicture: Use the image from the *`picture`* property. • mobilePicture: Use the image from the *`mobilePicture`* property. • noPicture: The item doesn't display a picture on mobile devices.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Default: desktopPicture
assignedTopics	Array	A list of sys_ids of existing taxonomy topics that control the visibility of the item in the Employee Center portal. For more information, see Associate a catalog item with a taxonomy topic in Employee Center. **Note:** This property is available only when the Employee Experience Taxonomy plugin (sn_ect) is active.
availableFor	Array	A list of sys_ids of user criteria [user_criteria] that define who can access the item.
notAvailableFor	Array	A list of sys_ids of user criteria [user_criteria] that define who can't access the item. This property overrides the *availableFor* property.
variableSets	Array	A list of variable identifiers of *VariableSet* objects or sys_ids of variable sets [item_option_new_set] to attach to the item. For more information, see VariableSet object. Format: `{ variableSet, order }`
hideAddToCart	Boolean	Flag that indicates whether to hide the **Add to Cart** button. Default: false
hideAddToWishlist	Boolean	Flag that indicates whether to hide the **Add to Wishlist** button. Default: false
hideDeliveryTime	Boolean	Flag that indicates whether to hide the delivery time. Default: false
hideQuantitySelector	Boolean	Flag that indicates whether to hide the **Quantity** field. Default: false
hideSaveAsDraft	Boolean	Flag that indicates whether to hide the **Save as Draft** button. Default: false
hideSP	Boolean	Flag that indicates whether to hide an item on Service Portal. Default: false
mandatoryAttachment	Boolean	Flag that indicates whether to require adding an attachment to submit a request. Default: false
hideAttachment	Boolean	Flag that indicates whether to hide the attachment section and not support adding attachments.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Default: false
allowEdit	Boolean	Flag that indicates whether users can edit the created record after submission. Default: false
canCancel	Boolean	Flag that indicates whether to display a Cancel button on the record producer form. Users can select Cancel to cancel the record producer and return to the last-viewed screen. Default: false
postInsertScript	Script	A server-side script that runs after the record is inserted in the associated table. You can call `current.update()` in this script. **Note:** This script overrides the target record values and record producer template values. This property supports inline JavaScript or a reference to another file in the application that contains a script. Format: • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide an inline script, use string literals or template literals for multiple lines of code: `'Script' or` `` `Script` ``. Default: `/**` `* This script is executed after the record is generated.` `* `current` Is the GlideRecord produced by Record Producer. Use `current.update()` to update the record` `* To access the variables, use `producer.var1` where var1 is the name of the variable` `* To access the Record Producer use `cat_item`` `*/`
redirectUrl	String	The redirect destination after the record is generated. Valid values: • generatedRecord: Redirects to the task record created by the record producer. • catalogHomePage: Redirects to the service catalog. Default: generatedRecord

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
saveOptions	String	Advanced configuration options for saving the record producer.
saveScript	Script	A script that runs at every step save in Catalog Builder. This property supports inline JavaScript or a reference to another file in the application that contains a script. Format: • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide an inline script, use string literals or template literals for multiple lines of code: `'Script' or` `` `Script` ``. Default: `/**` `* This script is executed at every step save in Catalog Builder.` `* This script is executed before `Script` is executed.` `* `current` Is the GlideRecord produced by Record Producer.` `* To access the variables, use `producer.var1` where var1 is the name of the variable` `* To access the Record Producer use `cat_item`` `*/`
script	Script	A server-side script that runs before the record is created. Use this script to assign values to fields on the record dynamically. Don't call `current.update()` or `current.insert()` in this script. This property supports inline JavaScript or a reference to another file in the application that contains a script. Format: • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide an inline script, use string literals or template literals for multiple lines of code: `'Script' or` `` `Script` ``. Default: `/** This script is executed before the Record is generated` `* `current`- GlideRecord produced by Record Producer` `* Don't use `current.update()` or `current.insert()` as the record is generated by Record Producer` `* Don't use `current.setValue('sys_class_name', 'xxx')` as this will trigger reparent flow and can cause data loss`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		* Avoid `current.setAbortAction()` and generate a separate record * Use `producer.var1` to access variables */
image	String	Deprecated. An image for the item.
$meta	Object	Metadata for the application metadata. With the *installMethod* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }` Valid values for *installMethod*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
import { CatalogItemRecordProducer } from "@servicenow/sdk/core";

const serviceCatalog = "e0d08b13c3330100c8b837659bba8fb4";
const itServicesCategory = "d258b953c611227a0146101fb1be7c31";

export const comprehensiveIncidentProducer =
 CatalogItemRecordProducer({
  $id: Now.ID["comprehensive_incident_producer"],
  name: "Report Incident with Full Configuration",
  shortDescription: "Complete incident producer with variables and scripts",
  table: "incident",

  catalogs: [serviceCatalog],
  categories: [itServicesCategory],

  variables: {
    short_description: SingleLineTextVariable({
      question: "Brief Summary",
      mandatory: true,
      mapToField: true,
      field: "short_description",
      order: 100
    }),
    urgency: SelectBoxVariable({
      question: "Urgency",
      mandatory: true,
      mapToField: true,
      field: "urgency",
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
      choices: {
        "1": { label: "High", sequence: 1 },
        "2": { label: "Medium", sequence: 2 },
        "3": { label: "Low", sequence: 3 }
      },
      order: 200
    }),
    assignment_group: ReferenceVariable({
      question: "Assignment Group",
      mapToField: true,
      field: "assignment_group",
      referenceTable: "sys_user_group",
      order: 300
    })
  },

  script: Now.include("../../scripts/rp-pre-insert.js"),
  postInsertScript: Now.include("../../scripts/rp-post-insert.js"),

  redirectUrl: "generatedRecord",
  view: "ess",
  allowEdit: true
})
```

CatalogUiPolicy object

Configure a catalog UI policy [catalog_ui_policy] to control variable behavior on catalog item forms based on conditions.

Catalog UI policies can make variables mandatory, read-only, visible, or hidden when specified conditions are met. For validation, calculations, or asynchronous calls, use catalog client scripts instead.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
shortDescription	String	Required. A short description of what the catalog UI policy does.
catalogItem	Reference or String	Required only if the *`variableSet`* property isn't used. The variable identifier of a *`CatalogItem`* or *`CatalogItemRecordProducer`* object or sys_id of the catalog item [sc_cat_item] or record producer [sc_cat_item_producer] to which the UI policy applies. For more information, see CatalogItem object or CatalogItemRecordProducer object.
variableSet	Reference or String	Required only if the *`catalogItem`* property isn't used. The variable identifier of a *`VariableSet`* object or sys_id of the variable set [item_option_new_set] to which the UI policy applies. For more information, see VariableSet object.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
appliesTo	String	The scope to which the UI policy applies. Valid values: • item: The policy applies to a catalog item. • set: The policy applies to a variable set. Default: item
active	Boolean	Flag that indicates whether the catalog UI policy is active. Default: true
global	Boolean	Flag that indicates on which views of the table the Catalog UI Policy runs. Default: true
onLoad	Boolean	Flag that indicates whether the catalog UI policy runs when the form loads. If false, the policy applies only when variable values change. Default: true
reverseIfFalse	Boolean	Flag that indicates whether to reverse the UI policy actions when the catalog condition evaluates to false. Default: true
inherit	Boolean	Flag that indicates whether the Catalog UI Policy is inherited. Default: false
isolateScript	Boolean	Flag that indicates whether the policy scripts run in an isolated scope. Default: true
catalogCondition	String	The encoded query conditions based on catalog item variable values that trigger the UI policy. For example: `catalogCondition: ` ` ${catalogItem.variables.singleLineText}=catalogitem^${catalogItem.variables.reference}ISNOTEMPTY^EQ` ` `
appliesOnCatalogTaskView	Boolean	Flag that indicates whether the UI policy applies to catalog items displayed in the order screen. This view is available to requesters. Default: true
appliesOnTargetRecord	Boolean	Flag that indicates whether the UI policy applies to records created for task-extended tables via record producers.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Default: false
appliesOnCatalogTasks	Boolean	Flag that indicates whether the UI policy applies to catalog task forms. This view is available to fulfillers. Default: false
appliesOnRequestedItems	Boolean	Flag that indicates whether the UI policy applies to requested item forms. This view is available to fulfillers. Default: false
runScripts	Boolean	Flag that indicates whether to run the *`executeIfTrue`* and *`executeIfFalse`* scripts for this UI policy. Use scripts to apply behaviors beyond read-only, mandatory, or visible, such as targeting a specific role. Default: false
executeIfTrue	String	A client-side script that runs when the catalog condition evaluates to true. The script must be wrapped in a `function onCondition() {}` function.
executeIfFalse	String	A client-side script that runs when the catalog condition evaluates to false. The script must be wrapped in a `function onCondition() {}` function.
runScriptsInUiType	String	The UI type on which the policy scripts run. Valid values: • desktop: Scripts run on the desktop interface. • mobileOrServicePortal: Scripts run on the mobile and Service Portal interfaces. • all: Scripts run on all interfaces. Default: desktop
vaSupported	Boolean	Flag that indicates whether the UI policy is supported in Virtual Agent conversations. Default: false
actions	Array	A list of variable actions to perform when the catalog condition is met. For more information, see actions array.
$meta	Object	Metadata for the application metadata. With the *`installMethod`* property, you can map the application metadata to an output directory that loads only in specific circumstances.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		`$meta: { installMethod: 'String' }` Valid values for *installMethod*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
import { CatalogUiPolicy } from "@servicenow/sdk/core";
import { hardwareRequestItem } from "./catalog-items/HardwareRequest.now";

export const managerApprovalPolicy = CatalogUiPolicy({
  $id: Now.ID["manager_approval_policy"],
  shortDescription: "Show manager approval when high priority selected",
  catalogItem: hardwareRequestItem,
  catalogCondition:
 `${hardwareRequestItem.variables.priority}=high^EQ`,
  actions: [
    {
      variableName:
 hardwareRequestItem.variables.manager_approval,
      visible: true,
      mandatory: true
    }
  ]
})
```

actions array

Configure the variable actions [catalog_ui_policy_action] that a catalog UI policy performs on variables when its conditions are met.

Each action in the array specifies a variable and the property changes to apply to that variable when the UI policy condition evaluates to true.

Properties

Name	Type	Description
variableName	String	Required. The variable to which the action applies.
visible	Boolean	Flag that indicates whether to make the variable visible. Default: false
disabled	Boolean	Flag that indicates whether to turn off the variable. Default: false

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
mandatory	Boolean	Flag that indicates whether to make the variable required. Default: false
cleared	Boolean	Flag that indicates whether to clear the variable value when the condition is met. Default: false
variableMessage	String	A message to display on the variable when the condition is met. This property applies only if the *variableMessageType* property has a value.
variableMessageType	String	The type of field message. Valid values: • info • warning • error
value	String	The value to set on the variable when the condition is met. This property applies only if the *valueAction* property is set to `setValue`.
valueAction	String	The action to take on the variable value when the condition is met. Valid values: • setValue: Sets the variable to the value specified in the *value* property. • clearValue: Clears the variable value.
order	Number	The order in which the action is evaluated relative to other actions. Default: 100
readOnly	Boolean	Flag that indicates whether to make the variable read only. Default: false

```
actions: [
    {
      variableName: laptopRequest.variables.justification,
      mandatory: true,
      variableMessage: "Justification required for urgent requests",
      variableMessageType: "info",
      order: 100
    },
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    {
      variableName: laptopRequest.variables.manager_approval,
      visible: true,
      mandatory: true,
      order: 200
    },
    {
      variableName: laptopRequest.variables.delivery_date,
      visible: true,
      order: 300
    }
]
```

CatalogClientScript object

Configure a catalog client script [catalog_script_client] that runs on the client side to control the behavior of a catalog item form.

Use catalog client scripts to validate user input, auto-populate fields, or display alerts on catalog item forms. For simple show/hide, mandatory, and read-only logic, use catalog UI policies instead.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
name	String	Required. A unique name for the catalog client script.
script	Script	The client-side script to run on the catalog item form. This property supports inline JavaScript or a reference to another file in the application that contains a script. Format: • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide an inline script, use string literals or template literals for multiple lines of code: `'Script' or` `` `Script` ``.
type	String	The event that triggers the client script to run. Valid values:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		• onLoad: Runs when the form loads. Use for initial setup such as setting field states, default values, and visibility. • onChange: Runs when a specific variable changes. Include an `if (isLoading) return;` guard. • onSubmit: Runs when the form is submitted. Return `false` to block submission. Avoid using *`GlideAjax`* in this type because of asynchronous issues.
uiType	String	The UI type on which the client script runs. Valid values: • desktop: Applies to the desktop interface. • mobileOrServicePortal: Applies to the mobile and Service Portal interfaces. • all: Applies to all interfaces. Default: desktop
active	Boolean	Flag that indicates whether the client script is enabled. Default: true
appliesTo	String	Required if using the *`variableSet`* property. The scope to which the catalog client script applies. Valid values: • item: The script applies to a catalog item. • set: The script applies to a variable set. Default: item
catalogItem	Reference or String	Required only if the *`variableSet`* property isn't used. The variable identifier of a *`CatalogItem`* or *`CatalogItemRecordProducer`* object or sys_id of the catalog item [sc_cat_item] or record producer [sc_cat_item_producer] to which the client script applies. For more information, see CatalogItem object or CatalogItemRecordProducer object.
variableSet	Reference or String	Required only if the *`catalogItem`* property isn't used. The variable identifier of a *`VariableSet`* object or sys_id of the variable set [item_option_new_set] to which the UI policy applies. For more information, see VariableSet object.
variableName	String	Required if the value of the *`type`* property is `onChange`. The name of the catalog variable that triggers the script when its value changes.
appliesOnCatalogTaskView	Boolean	Flag that indicates whether the client script applies to catalog items displayed in the order screen of the service catalog. This view is available to requesters.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Default: true
appliesOnRequestedItems	Boolean	Flag that indicates whether the client script applies to requested item forms after the item is requested. This view is available to fulfillers. Default: false
appliesOnCatalogTasks	Boolean	Flag that indicates whether the client script applies to catalog task forms for the item. This view is available to fulfillers. Default: false
appliesOnTargetRecord	Boolean	Flag that indicates whether the client script applies to records created for task-extended tables via record producers. Default: false
global	Boolean	Flag that indicates whether the client script runs in the global scope. Default: true
vaSupported	Boolean	Flag that indicates whether the client script is supported in Virtual Agent conversations. Default: false
publishedRef	String	The sys_id of a published catalog item [sc_cat_item] or record producer [sc_cat_item_producer] that this client script references.
$meta	Object	Metadata for the application metadata. With the *`installMethod`* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }` Valid values for *`installMethod`*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
import { CatalogClientScript } from "@servicenow/sdk/core";
import { laptopRequest } from "../catalog-items/laptop-request.now";
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
CatalogClientScript({
  $id: Now.ID["laptop_onload"],
  name: "Laptop Request - OnLoad",
  script: Now.include("../../client/laptop-onload.js"),
  type: "onLoad",
  catalogItem: laptopRequest,
  active: true,
  appliesOnCatalogItemView: true
});
```

The client script is defined in the `laptop-onload.js` file referenced from the *`script`* property. For example:

```
function onLoad() {
   // Set initial field states
   g_form.setReadOnly("estimated_cost", true);
   g_form.setValue("estimated_cost", "$0");
   g_form.setMandatory("justification", true);
}
```

VariableSet object

Create a variable set [item_option_new_set] that groups reusable variables for use across multiple catalog items and record producers.

Variable sets are reusable collections of variables that can be attached to catalog items and record producers. Catalog UI policies and client scripts can be scoped to a variable set by setting the `appliesTo` property to `set`.

Note:

- Variable sets within a catalog item can't have the same internal name.
- Within a catalog item, the name of a variable can't be the same as the title or internal name of a variable set.
- Catalog client scripts and catalog UI policy scripts must refer to the internal name of a variable set, not the title or display name.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
title	String	Required. The display title of the variable set. This title appears only if the the *`displayTitle`* property is set to true.
internalName	String	The internal name of the variable set used for programmatic access through the *`g_form`* API and server-side scripts. If not provided, the internal name is automatically generated from the *`title`* property.
description	String	A description of the variable set and its intended use.
type	String	The type of the variable set.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Valid values: • singleRow: Variables are arranged in a single row. • multiRow: Variables are arranged in multiple rows that users can add or remove. Default: singleRow
layout	String	The column layout for the variable set. Valid values: • normal: Variables are displayed in a single column. • 2down: Variables are displayed in two columns, one side then the other. • 2across: Variables are displayed in two columns with alternating sides. Default: normal
order	Number	The order in which a variable set displays relative to other variable sets on the form. Default: 100
displayTitle	Boolean	Flag that indicates whether to display a collapsible section header for the variable set. If true, the value of the *title* property is displayed as a collapsible header. **Note:** Check box variables are grouped under a default title of Options. To use a custom title, add a Label variable with an *order* value that places it directly above the check box variables. Default: false
setAttributes	String	Additional configuration attributes for the variable set as a comma-separated string, such as `max_rows=10,collapsible=true`. Use the `max_rows` attribute to set the maximum number of rows for a multi-row variable set. **Note:** *AttachmentVariable*, *ContainerVariable*, *HtmlVariable*, and *CustomVariable* types aren't supported in multi-row variable sets.
readRoles	Array	A list of variable identifiers of *Role* objects or sys_ids of roles [sys_user_role] that can view the variable set. For more information, see Role API - ServiceNow Fluent.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
writeRoles	Array	A list of variable identifiers of *`Role`* objects or sys_ids of roles [sys_user_role] that can modify variable values in the set. For more information, see Role API - ServiceNow Fluent.
createRoles	Array	A list of variable identifiers of *`Role`* objects or sys_ids of roles [sys_user_role] that can create row instances. For more information, see Role API - ServiceNow Fluent. This property applies only if the value of the *`type`* property is `multiRow`.
variables	Object	The variable definitions for the item that provide options for requesting it. Each variable type has a specific function. For example: `variables: { laptopType: SelectBoxVariable({ question: "Laptop Type", choices: { standard: { label: "Standard Laptop", sequence: 1 }, developer: { label: "Developer Workstation", sequence: 2 } }, mandatory: true, order: 100 }), justification: MultiLineTextVariable({ question: "Business Justification", mandatory: true, order: 200 }) }` For general information about catalog variables, see Service catalog variables.
name	String	An optional name for additional identification of the variable set.
version	Number	The version of the variable set. Default: 0
$meta	Object	Metadata for the application metadata. With the *`installMethod`* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Valid values for *installMethod*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
import {
  VariableSet,
  EmailVariable,
  SingleLineTextVariable,
  ReferenceVariable
} from "@servicenow/sdk/core";

export const contactInfoSet = VariableSet({
  $id: Now.ID["contact_info_set"],
  title: "Contact Information",
  description: "Standard contact information fields",
  type: "singleRow",
  layout: "2across",
  order: 100,
  displayTitle: true,
  variables: {
    email: EmailVariable({
      question: "Email Address",
      mandatory: true,
      order: 100
    }),
    phone: SingleLineTextVariable({
      question: "Phone Number",
      mandatory: true,
      order: 200
    }),
    department: ReferenceVariable({
      question: "Department",
      referenceTable: "cmn_department",
      referenceQualCondition: "active=true",
      order: 300
    })
  }
})
```

Service Level Agreement API - ServiceNow Fluent

The Service Level Agreement API defines service level agreements [contract_sla] that set the amount of time for a task to reach a specified condition, ensuring that tasks are resolved according to the service levels agreed between a service provider and customer.

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For general information about SLAs, see Service Level Management.

Related topics

ServiceNow Fluent

Sla object

Create an SLA definition [contract_sla] that controls the timing, conditions, workflows, and other information required to create and progress task SLAs.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
name	String	Required. A name for the SLA definition.
table	String	The name of the table to which the SLA applies. Default: incident
type	String	The type of service level agreement. The type is used for reporting purposes only. Valid values: • SLA: A service level agreement between a service provider and an external customer. • OLA: An operational level agreement between internal teams within the same organization. • Underpinning contract: A contract with an external vendor that underpins the service delivered to the customer. Default: SLA
active	Boolean	Flag that indicates whether the SLA definition is active and can be matched against task records. Valid values: • true: The SLA definition is active. • false: The SLA definition is inactive and isn't matched against task records. Default: true
target	String	The stage of the task that the SLA measures. The target is used for filtering, searching, and reporting purposes only. Valid values:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		• response: The SLA measures the time to first respond to a task. • resolution: The SLA measures the time to resolve a task.
duration	Object	The time duration within which the task must reach the target condition. Use the `Duration()` function to specify the duration. This property is required if the value of the *`durationType`* property is empty. Format: `duration: Duration({ days: Number, hours: Number, minutes: Number, seconds: Number })`
durationType	Reference or String	The sys_id of a relative duration [cmn_relative_duration], such as Breach on Due Date or End of next business day, to use instead of a user-specified duration. To define a relative duration, use the Record API - ServiceNow Fluent.
relativeDurationWorks	String	The record type from which the relative duration is calculated. This property only applies if the value of the *`durationType`* property is set. Valid values: • Task record: The relative duration is calculated from the task record. • SLA record: The relative duration is calculated from the SLA record. Default: Task record
schedule	Reference or String	The sys_id of a schedule [cmn_schedule] for the time periods during which the SLAs accumulate business time. To define a schedule, use the Record API - ServiceNow Fluent. This property is required if the value of the *`scheduleSource`* property is `sla_definition`.
scheduleSource	String	The source from which the schedule is obtained. Valid values: • sla_definition: The schedule is specified in the SLA definition using the *`schedule`* property. • task_field: The schedule is obtained from a field on the task record specified with the *`scheduleSourceField`* property. • no_schedule: No schedule is used, and the SLA runs continuously (24x7).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Default: sla_definition
scheduleSourceField	String	The field from the task that provides the schedule. This property is required if the value of the *`scheduleSource`* property is `task_field`.
conditions	Object	Encoded query conditions that control the timing of an SLA. For example, `'priority=1^state!=6'` matches tasks with a priority of 1 that aren't in a closed state. For more information about SLA conditions, see Create an SLA definition. For information about filter queries, see Operators available for filters and queries. Format: `conditions: { start: 'String', stop: 'String', pause: 'String', resume: 'String', reset: 'String', cancel: 'String', }`
advancedConditionType	String	The type of advanced condition logic to apply. Valid values: • none: No advanced condition logic is applied. • advanced: A custom advanced condition script is used. • advanced_journal: Advanced condition logic based on journal entries is used. • advanced_system: Advanced condition logic based on system events is used. • advanced_journal_and_system: Advanced condition logic based on both journal entries and system events is used. Default: none
conditionType	String	The sys_id of an SLA condition [sla_condition_class] that determines when transitions between different stages of each task SLA.
resetAction	String	The action to take on the current task SLA record when the reset condition is met. Valid values: • cancel: The current task SLA record is canceled and a new one is created. • complete: The current task SLA record is marked as complete and a new one is created.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Default: cancel
whenTo	Object	Settings that control when a paused SLA resumes and when an SLA cancels. • *resume*: The behavior that controls when a paused SLA resumes timing. Valid values: ○ on_condition: The SLA resumes when the task record matches the condition defined with the *resume* property of the *conditions* object. ○ no_match: The SLA resumes when the task record no longer matches the pause condition defined with the *pause* property of the *conditions* object. Default: on_condition • *cancel*: The behavior that controls when an active SLA is canceled. Valid values: ○ on_condition: The SLA is canceled when the task record matches the condition defined with the *cancel* property of the *conditions* object. ○ no_match: The SLA is canceled when the task record no longer matches the start condition defined with the *start* property of the *conditions* object. ○ never: The SLA is never canceled regardless of task record state. Default: on_condition Format: `whenTo: { resume: 'String', cancel: 'String', }`
retroactive	Object	Settings that control whether the SLA start time is set to a point in the past based on a specified field value. • *start*: Flag that indicates whether retroactive start is enabled. When enabled, the SLA start time is set to the value of the field specified in *setStartTo* rather than the time when the start condition was first met. Default: false • *setStartTo*: The field on the task record whose value is used as the SLA start time when retroactive start is enabled.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		This property is required if the value of the *start* property is true. • *pause*: Flag that indicates whether retroactive pause is enabled when retroactive start is active. When enabled, any time that the task was in a paused state before the SLA was attached is subtracted from the elapsed time. Default: true Format: `retroactive: { start: Boolean, setStartTo: 'String', pause: Boolean, }`
timezoneSource	String	The source from which the time zone for SLA calculations is obtained. Valid values: • task.caller_id.time_zone: The time zone of the caller on the task record. • task.caller_id.location.time_zone: The time zone of the caller's location on the task record. • task.cmdb_ci.location.time_zone: The time zone of the configuration item's location on the task record. • task.location.time_zone: The time zone of the location on the task record. • sla.timezone: The time zone specified with the *timezone* property. Default: task.caller_id.time_zone
timezone	String	The time zone to use for SLA calculations, such as `US/Pacific`. This property is required if the value of the *timezoneSource* property is `sla.timezone`.
overrides	Reference or String	The variable identifier or sys_id of another SLA definition [contract_sla] that this SLA definition overrides.
workflow	Reference or String	The variable identifier or sys_id of a workflow [wf_workflow] to run when the SLA reaches a milestone or breaches. To define a workflow, use the Record API - ServiceNow Fluent.
flow	Reference or String	The variable identifier or sys_id of a flow [sys_hub_flow] to run when the SLA reaches a milestone or breaches. To define a flow, use the Flow API - ServiceNow Fluent.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Default: Default SLA flow (828f267973333300e289235f04f6a7a3)
vendor	Reference or String	The variable identifier or sys_id of a company [core_company] that is the vendor for an underpinning contract SLA. To define a company, use the Record API - ServiceNow Fluent.
domainPath	String	The domain path that determines which domain owns the SLA in a multi-domain environment. Default: The global domain (/)
enableLogging	Boolean	Flag that indicates whether debug logging is enabled for the SLA definition. Valid values: • true: Debug logging is enabled. • false: Debug logging is turned off. Default: false
$meta	Object	Metadata for the application metadata. With the *installMethod* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }` Valid values for *installMethod*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
import { Sla, Duration } from '@servicenow/sdk/core'

Sla({
    $id: Now.ID['incident-p1-resolution'],
    name: 'P1 Incident Resolution',
    table: 'incident',
    target: 'resolution',
    duration: Duration({ hours: 4 }),
    schedule: 'b1992362eb601100fcfb858ad106fe16',
    conditions: {
        start: 'priority=1',
        stop: 'state=6',
        pause: 'state=3',
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
        resume: 'state!=3',
    },
    whenTo: {
        resume: 'on_condition',
    },
    resetAction: 'cancel',
})
```

Service Portal API - ServiceNow Fluent

The Service Portal API defines custom widgets [sp_widget] for portal pages.

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

For general information about portals, see Service Portal.

Related topics

ServiceNow Fluent

SPWidget object

Create a custom widget [sp_widget] to include on a portal page.

For general information about creating custom widgets, see Developing custom widgets.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
name	String	Required. A name for the widget.
category	String	The type of widget. Valid values: • standard • otherApplications • custom • sample • knowledgeBase • servicePortal • serviceCatalog Default: custom
clientScript	Script	A client-side script that defines the AngularJS controller. This property supports inline JavaScript or a reference to another file in the application that contains a script. Format:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		• To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide an inline script, use string literals or template literals for multiple lines of code: `'Script' or` `` `Script` ``. Default: `api.controller=function() { /* widget controller */ var c = this;};`
serverScript	Script	A server-side script that sets the initial widget state, sends data to the widget's client script using the `data` object, or runs server-side queries. This property supports inline JavaScript or a reference to another file in the application that contains a script. Format: • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide an inline script, use string literals or template literals for multiple lines of code: `'Script' or` `` `Script` ``. Default: `(function () { /* populate the 'data' object */ /* e.g., data.table = $sp.getValue('table'); */ })();`
controllerAs	String	A variable for a reference to the controller in the directive's scope. The client script accesses the server data object using the `c.data` variable by default. Default: c
htmlTemplate	String	The body HTML code that defines what is rendered when the page is shown. It can contain either static XHTML, dynamically generated content defined as Jelly, or call script includes and UI Macros. This property supports a reference to another file in the application that contains HTML or inline HTML. Format: • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide inline HTML, use string literals or template literals for multiple lines of code: `'HTML' or` `` `HTML` ``.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Default: `<div><!-- your widget template --></div>`
customCss	String	The CSS or SCSS that defines the widget style. This property supports a reference to another file in the application that contains CSS or inline CSS. Format: • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide inline CSS, use string literals or template literals for multiple lines of code: `'CSS' or `CSS``.
dataTable	String	The table in which to store widget instance options. To define a custom option schema, you can add fields to a table that extends the Widget Instance [sp_instance] table and set your widget to use the extension table as a data source. For more information, see Store instance options in a table. Default: sp_instance
demoData	String or Object	Data that demonstrates the widget functionality. This property supports inline strings, a reference to another file in the application that contains JSON, or inline JSON serializable objects. Format: • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide inline JSON, use an object with key and value pairs: `{ key: value }`.
description	String	A description of the widget and its purpose.
docs	Reference	The variable identifier of the Service Portal documentation [sp_documentation] that provides additional information about the widget and its purpose. To define Service Portal documentation, see Record API - ServiceNow Fluent.
fields	Array	A list of column names from the data table to use in the widget option schema.
hasPreview	Boolean	Flag that indicates whether you can preview the widget from the Widget Editor. Valid values:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		• true: You can preview the widget from the Widget Editor. • false: You can't preview the widget from the Widget Editor. Default: false
id	String	A unique ID for the widget. The ID can't contain spaces.
linkScript	Script	A link function that uses AngularJS to directly manipulate the DOM. This property supports inline JavaScript or a reference to another file in the application that contains a script. Format: • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide an inline script, use string literals or template literals for multiple lines of code: `'Script' or` `` `Script` ``. Default: `function link(scope, element, attrs, controller) { }`
roles	Array	A list of variable identifiers of *Role* objects or names of roles that can access the widget. For more information, see Role API - ServiceNow Fluent.
optionSchema	Array	A list of parameters that a Service Portal administrator (sp_admin) can configure for a widget instance. The widget option schema supports reusing a widget and uniquely configuring instances of the widget on different pages. For more information, see Widget option schema. `optionSchema: [ { name: 'String', label: 'String', section: 'String', type: 'String', defaultValue: 'String', hint: 'String' }, ... ]` • name: Required. The name of the parameter. • label: Required. A label for the parameter that appears in the widget instance options. • section: Required. The section of the widget instance options in which the parameter appears.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Valid values: ○ data ○ behavior ○ documentation ○ presentation ○ other • type: Required. The data type of the parameter. Valid values: ○ string ○ boolean ○ integer ○ reference ○ choice ○ fieldList ○ fieldName ○ glideList ○ glyphIcon • defaultValue: The default value of the parameter. • hint: A short description of the parameter that displays as tooltip when hovering over it.
public	Boolean	Flag that indicates whether the widget is available to unauthenticated users. Valid values: • true: Unauthenticated users can access the widget. • false: Only authenticated users can access the widget. Default: false
dependencies	Array	A list of variable identifiers of *SPWidgetDependency* objects or names or sys_ids of dependencies for the widget. For more information, see SPWidgetDependency object.
angularProviders	Array	A list of variable identifiers of *SPAngularProvider* objects or names or sys_ids of Angular providers for the widget. For more information, see SPAngularProvider object.
templates	Array	A list of Angular ng-templates [sp_ng_template] to associate with the widget. Angular ng-templates contain content that is rendered only when you instruct it to render.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		`` templates: [{ $id: Now.ID['String' or Number] id: 'String' htmlTemplate: 'HTML' or `HTML` }, ...] ``` • $id: Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. Format: `Now.ID['String' or Number]` • id: Required. The ID of the Angular ng-template. • htmlTemplate: Required. The HTML template content. This property supports a reference to another file in the application that contains HTML or inline HTML.
$meta	Object	Metadata for the application metadata. With the *`installMethod`* property, you can map the application metadata to an output directory that loads only in specific circumstances. ``` $meta: { installMethod: 'String' } ``` Valid values for *`installMethod`*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
import { SPWidget } from '@servicenow/sdk/core'

SPWidget({
    $id: 'my_simple_widget',
    name: 'My Simple Widget',
    category: 'knowledgeBase',
    clientScript: Now.include('./client.js'),
    serverScript: Now.include('./server.js'),
    controllerAs: '$ctrl',
    customCss: Now.include('./custom_css.css'),
    dataTable: 'sp_instance',
    demoData: { message: 'Hello, World!' },
    description: 'This is a test widget',
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    docs: widgetDoc,
    htmlTemplate: Now.include('./template.html'),
    fields: ['color', 'class_name'],
    hasPreview: true,
    id: 'my-simple-widget',
    linkScript: Now.include('./link.client.js'),
    optionSchema: [
        {
            name: 'my_option',
            label: 'My Option',
            type: 'string',
            section: 'behavior'
        }
    ],
    roles: [manager, 'admin'],
})
```

SPAngularProvider object

Create an Angular Provider [sp_angular_provider] to reuse components in multiple widgets and improve portal performance.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
name	String	Required. A name for the Angular provider.
clientScript	Script	A client-side script to reuse in widgets.This property supports inline JavaScript or a reference to another file in the application that contains a script. Format: • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide an inline script, use string literals or template literals for multiple lines of code: `'Script' or `Script``.
type	String	The type of Angular provider. Valid values: • directive • factory • service

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Default: directive
requires	Array	A list of variable identifiers of other *SPAngularProvider* objects or names or sys_ids of Angular providers.
$meta	Object	Metadata for the application metadata. With the *installMethod* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }` Valid values for *installMethod*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
import { SPAngularProvider } from '@servicenow/sdk/core'
const OTHER_ANGULAR_PROVIDER = 'd11f285fe069e1f119b44bd05c0770aa'

SPAngularProvider({
    $id: 'my_angular_provider',
    name: 'my_angular_provider',
    clientScript: Now.include('my_angular_provider.client.js'),
    type: 'directive',
    requires: [OTHER_ANGULAR_PROVIDER]
})
```

SPWidgetDependency object

Create a widget dependency [sp_dependency] to link JavaScript and CSS files to widgets and use third-party libraries, external style sheets, or Angular modules.

Dependencies are loaded asynchronously from the server when needed. Widgets can have as many or as few dependencies as needed. However, the more you add, the more content a widget must download to render on the page. Keep dependencies as small as possible for more efficient load times. For more information, see Create a widget dependency.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
name	String	Required. The name of the widget dependency.
angularModuleName	String	The name of the Angular module to load if the JS include is an Angular module.
includeOnPageLoad	Boolean	Flag that indicates when the dependency loads on a page. Valid values: • true: The dependency loads with the initial page load. • false: The dependency loads only when the linked widget is loaded on a page. Default: false
cssIncludes	Array	A list of variable identifiers of *`CssInclude`* objects or sys_ids of CSS includes and their order. For more information, see CssInclude object. `cssIncludes: [` `  {` `  order: Number,` `  include: 'String' or Reference,` `  },` `  ...` `]`
jsIncludes	Array	A list variable identifiers of *`JsInclude`* objects or sys_ids of JS includes and their order. For more information, see JsInclude object. `jsIncludes: [` `  {` `  order: Number,` `  include: 'String' or Reference,` `  },` `  ...` `]`
portalsForPageLoad	Array	A list of sys_ids of portals [sp_portal] that load the widget dependency. If empty, the dependency is included on the page load for all portals.
$meta	Object	Metadata for the application metadata. With the *`installMethod`* property, you can map the application metadata to an output directory that loads only in specific circumstances.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		`$meta: { installMethod: 'String' }` Valid values for *`installMethod`*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
import { SPWidgetDependency } from '@servicenow/sdk/core'

SPWidgetDependency({
    $id: 'samplejs',
    name: 'Sample',
    angularModuleName: 'samplejs',
    includeOnPageLoad: true,
    portalsForPageLoad:
 ['b4572a48262a16df3032b48cef75a853', 'fe12dbbed14bd3f712f0787141c2f656'],
    cssIncludes: [
        {
            order: 100,
            include: localCss,
        },
        {
            order: 200,
            include: '94112ccb0fb3c2ed072b01d3cb401196',
        },
    ],
    jsIncludes: [
        {
            order: 100,
            include: localJs,
        },
        {
            order: 200,
            include: 'f8af18a5e6c71a3702c4f2038b43cf62',
        },
    ],
})
```

CssInclude object

Create a CSS include [sp_css_include] to reference a style sheet or external CSS in a widget dependency.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
name	String	Required. The name of the CSS include.
url	String	The URL to an external CSS file. This property is required if a style sheet isn't provided with the *`spCss`* property.
spCss	String	The sys_id of a style sheet [sp_css]. This property is required if an external CSS file isn't provided with the *`url`* property.
rtlCssUrl	String	The URL to an external right-to-left (RTL) CSS file for mirroring the direction of a widget when the session language is a right-to-left language, such as Hebrew. For more information, see Styling for right-to-left languages in portals.
lazyLoad	Boolean	Flag that indicates how to load the CSS Include. This property applies only if you use the *`spCSS`* property to specify a style sheet. Valid values: • true: The CSS include loads asynchronously to improve page load time. • false: The CSS include doesn't load asynchronously. Default: false
$meta	Object	Metadata for the application metadata. With the *`installMethod`* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }` Valid values for *`installMethod`*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
import { CssInclude } from '@servicenow/sdk/core'

const localCss = CssInclude({
    $id: '22bcf16da81e2bc0340c53d50d531adf',
    name: 'Sample Styles',
    spCss: '50e3e32aa321b1c7d1945c5f423228bd',
})
```

JsInclude object

Create a JS include [sp_js_include] to reference a UI script or external JavaScript code in a widget dependency.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
name	String	Required. The name of the JS include.
url	String	The URL to an external JavaScript file. The URL should be an absolute path. This property is required if a UI script isn't provided with the *`sysUiScript`* property.
sysUiScript	String	The sys_id of a UI script [sys_ui_script]. This property is required if an external JavaScript file isn't provided with the *`url`* property.
$meta	Object	Metadata for the application metadata. With the *`installMethod`* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }` Valid values for *`installMethod`*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
import { JsInclude } from '@servicenow/sdk/core'

const localJs = JsInclude({
    $id: '98239e4eadfac88b01cce7daa23b6fc3',
    name: 'Sample Framework',
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    sysUiScript: 'b67af05645f738df1f286bb3e9ecd55f',
})
```

Table API - ServiceNow Fluent

The Table API defines tables [sys_db_object] to store data in an application.

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

Create a table using the *Table* object. From the *schema* property, add Column objects, such as *StringColumn* or *IntegerColumn*, to define the columns.

For general information about tables, see Table administration.

Related topics

ServiceNow Fluent

Table object

Create a table [sys_db_object] in an application.

Properties

Name	Type	Description
name	String	Required. A name for the table beginning with the application scope and in all lowercase letters in the following format: <scope>_<name>. The name should match the variable identifier of the *Table* object. **Note:** To add columns to an existing table in a different application scope, you can provide the name of the table without the application scope followed by `as any`. The column names must begin with the application scope instead. Maximum length: 80
schema	Array	Required. A list of *Column* objects. For more information, see Column object.
extends	String	The name of any other table on which the table is based. Extending a base table incorporates all the fields of the original table and creates system fields for the new table. If they are in the same scope or if they can be configured from other scopes, you can extend tables that are marked as extensible.
label	String or Array	A unique label for the table in list and form views. Field labels can be provided as a string or an array of *label* objects. For more information, see label object. Maximum length: 80 Default: the value of the *name* property

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
licensingConfig	Object	The licensing configuration [ua_table_licensing_config] for a table. For more information, see licensingConfig object.
display	String	The default display column. Use a column name from the `schema` property.
extensible	Boolean	Flag that indicates whether other tables can extend the table. Valid values: • true: Other tables can extend the table. • false: Other tables can't extend the table. Changing this property from true to false prevents the creation of additional child tables but existing child tables remain unchanged. Default: false
liveFeed	Boolean	Flag that indicates if live feeds are available for records in the table. Valid values: • true: Live feeds are provided for records in the table. This option adds the **Show Live Feed** option () in the form header. • false: Live feeds aren't provided for records in the table. Default: false
accessibleFrom	String	The application scopes that can access the table. Valid values: • public: The table is accessible from all application scopes. • package_private: The table is accessible from only the application scope it's in. Default: public
callerAccess	String	The access level for cross-scope requests. Valid values: • restricted: Calls to the resource must be manually approved. Access requests are tracked in the Restricted Caller Access table with a status of Requested.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		• tracking: Calls to the resource are automatically approved. Calls are tracked in the Restricted Caller Access table with a status of Allowed. • none: Cross-scope calls to the resource are approved or denied based on the value of the *accessibleFrom* property. For more information, see Restricted caller access privilege settings. Default: none
actions	Array	A list of access options. Valid values: • read: Allow script objects from other application scopes to read records stored in this table. For example, a script in another application can query data on this table. Read access is required to grant any other API record operations. • create: Allow script objects from other application scopes to create records in this table. For example, a script in another application can insert a new record in this table. • update: Allow script objects from other application scopes to modify records stored in this table. For example, a script in another application can modify a field value on this table. • delete: Allow script objects from other application scopes to delete records from this table. For example, a script in another application can remove a record from this table. Default: read
allowWebServiceAccess	Boolean	Flag that indicates whether web services can make calls to the table. Valid values: • true: Web services can make calls to the table. • false: Web services can't make calls to the table. Default: false
allowNewFields	Boolean	Flag that indicates whether to allow design time configuration of new fields on the table from other application scopes. Valid values:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		• true: Allow design time configuration of new fields on the table from other application scopes. • false: Don't allow design time configuration of new fields on the table from other application scopes. Default: false
allowUiActions	Boolean	Flag that indicates whether to allow design time configuration of UI actions on the table from other application scopes. Valid values: • true: Allow design time configuration of UI actions on the table from other application scopes. • false: Don't allow design time configuration of UI actions on the table from other application scopes. Default: false
allowClientScripts	Boolean	Flag that indicates whether to allow design time configuration of client scripts on the table from other application scopes. Valid values: • true: Allow design time configuration of client scripts on the table from other application scopes. • false: Don't allow design time configuration of client scripts on the table from other application scopes. Default: false
audit	Boolean	Flag that indicates whether to track the creation, update, and deletion of all records in the table. Valid values: • true: Track the creation, update, and deletion of all records in the table • false: Don't track the creation, update, and deletion of all records in the table. Default: false
readOnly	Boolean	Flag that indicates whether users can edit fields in the table. Valid values: • true: Users can't edit fields in the table. • false: Users can edit fields in the table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Default: false
textIndex	Boolean	Flag that indicates whether search engines index the text in a table. Valid values: • true: The table's text is indexed. • false: The table's text isn't indexed. Default: false
attributes	Object	Key and value pairs of any supported dictionary attributes [sys_schema_attribute]. For example: `attributes: { updateSyncCustom: Boolean, nativeRecordLock: Boolean }` For more information, see Dictionary Attributes.
index	Array	A list of column references to generate indexes in the metadata XML of the table. The value of the *element* property should match the object key used with the *Column* object. A database index increases the speed of accessing data from the table with the expense of using additional storage. `index: [ { name: 'String', element: 'String', unique: Boolean }, ... ]`
autoNumber	Object	The auto-numbering configuration [sys_number] for a table. For more information, see autoNumber object.
scriptableTable	Boolean	Flag that indicates whether the table is a remote table that uses data retrieved from an external source. For more information, see Remote tables. Valid values: • true: The table is a remote table. • false: The table isn't a remote table. Default: false

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For typeahead support for columns, assign the *Table* object to an exported variable with the same name as the *name* property.

```
import { Table, StringColumn } from "@servicenow/sdk/core";
import { myFunction } from "../server/myFunction.js"

export const x_snc_example_to_do = Table({
    name: 'x_snc_example_to_do',
    label: 'My To Do Table',
    extends: 'task',
    schema: {
        status: StringColumn({ label: 'Status' }),
        deadline: StringColumn({
            label: 'Deadline',
            active: true,
            mandatory: false,
            readOnly: false,
            maxLength: 40,
            dropdown: 'none',
            attributes: {
                updateSync: false,
            },
            default: 'today',
            dynamicValueDefinitions: {
                type: 'calculated_value',
                calculatedValue: '',
            },
            choices: {
                choice1: {
                    label: 'Choice1 Label',
                    sequence: 0,
                    inactiveOnUpdate: false,
                    dependentValue: '5',
                    hint: 'hint',
                    inactive: false,
                    language: 'en',
                },
                choice2: { label: 'Choice2 Label', sequence: 1 },
            },
        }),
        dynamic1: StringColumn({
            dynamicValueDefinitions: {
                type: 'calculated_value',
                calculatedValue: myFunction,
            },
        }),
        dynamic2: StringColumn({
            dynamicValueDefinitions: {
                type: 'dynamic_default',
                dynamicDefault: `gs.info()`,
            },
        }),
        dynamic3: StringColumn({
            dynamicValueDefinitions: {
                type: 'dependent_field',
                columnName: 'status',
            },
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
        }),
        dynamic4: StringColumn({
            dynamicValueDefinitions: {
                type: 'choices_from_other_table',
                table: 'sc_cat_item',
                field: 'display',
            },
        }),
    },
    actions: ['create', 'read'],
    display: 'deadline',
    accessibleFrom: 'package_private',
    allowClientScripts: true,
    allowNewFields: true,
    allowUiActions: true,
    allowWebServiceAccess: true,
    extensible: true,
    liveFeed: true,
    callerAccess: 'none',
    autoNumber: {
        number: 10,
        numberOfDigits: 2,
        prefix: 'abc',
    },
    audit: true,
    readOnly: true,
    textIndex: true,
    attributes: {
        updateSync: true,
    },
    index: [
        {
            name: 'idx',
            element: 'status',
            unique: true,
        },
    ],
})
```

Column object

Add a column [sys_dictionary] to a table.

Add *Column* objects in the *schema* property of the *Table* object.

There are many types of columns based on the field type. Column objects use the format *<Type>Column* where *<Type>* is the field type. For information about field types, see Field types reference.

The following types of columns are supported: *ListColumn*, *RadioColumn*, *StringColumn*, *ChoiceColumn*, *ScriptColumn*, *BooleanColumn*, *ConditionsColumn*, *DecimalColumn*, *IntegerColumn*, *VersionColumn*, *DomainIdColumn*, *FieldNameColumn*, *ReferenceColumn*, *TableNameColumn*, *UserRolesColumn*, *BasicImageColumn*, *DocumentIdColumn*, *DomainPathColumn*, *TranslatedTextColumn*, *SystemClassNameColumn*, *TranslatedFieldColumn*, *GenericColumn*, *DateColumn*, *DateTimeColumn*, *CalendarDateTime*, *BasicDateTimeColumn*, *DueDateColumn*, *CalendarDateTime*, *IntegerDateColumn*, *ScheduleDateTimeColumn*, *OtherDateColumn*,

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Password2Column, *GuidColumn*, *JsonColumn*, *NameValuePairsColumn*, *UrlColumn*, *EmailColumn*, *HTMLColumn*, *FloatColumn*, *MultiLineTextColumn*, *DurationColumn*, *TimeColumn*, *FieldListColumn*, *SlushBucketColumn*, *TemplateValueColumn*, and *ApprovalRulesColumn*.

Properties

Name	Type	Description
label	String or Array	A unique label for the column that appears on list headers and form fields. Field labels can be provided as a string or an array of *label* objects. For more information, see label object. Default: the key used for the column object
maxLength	Number	The maximum length of values in the column. A length of under 254 appears as a single-line text field. Anything 255 characters or over appears as a multi-line text box. **Note:** To avoid data loss, only decrease the length of a string field when you're developing a new application and not when a field contains data. Default: varies depending on the column type
active	Boolean	Flag that indicates whether to display the field in lists and forms. Valid values: • true: Displays the field. • false: Hides the field. Default: true
mandatory	Boolean	Flag that indicates whether the field must contain a value to save a record. Valid values: • true: The field must contain a value. • false: The field isn't required. Default: false
readOnly	Boolean	Flag that indicates whether you can edit the field value. Valid values: • true: You can't change the value, and the system calculates and displays the data for the field. • false: You can change the field value.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Default: false
readOnlyOption	String	Control the ability to edit read-only fields by configuring read-only options. Valid values: • instance_configured: Maintains backwards compatibility and testing read-only behavior in non-production instances. • display_read_only: Displays the field as read-only in the UI, but allows changes to the read-only field through client scripts and server-side operations such as *`TableAPI`*, *`GraphQL`*, and *`GlideRecordSecure()`*. • client_script_modifiable: Displays the field as read-only in the UI, and allows changes to read-only fields through client scripts but not through background scripts or server-side APIs like *`TableAPI`*, *`GraphQL`*, and *`GlideRecordSecure()`*. • strict_read_only: Displays the field as read-only in the UI, but prevents any changes from both client scripts and server-side APIs. For more information, see Configuring read-only security options.
default	Any	The default value of the field when creating a record. The value must use the correct type based on the column type.
choices	Object	A list of choices [sys_choice] for a column. For more information, see choices object. This property only applies to *`ChoiceColumn`* objects and column types that extend choice columns. It can include either an array of primitive values or a series of choice objects.
attributes	Object	Key and value pairs of any supported dictionary attributes [sys_schema_attribute]. For example: `attributes: { updateSyncCustom: Boolean, nativeRecordLock: Boolean }` For more information, see Dictionary Attributes.
functionDefinition	String	The definition of a function that the field performs, such as a mathematical operation, field length computation, or day of the week calculation.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Each definition begins with `glidefunction:`, followed by the operation to be performed (such as, `concat`), followed by function parameters. Constants must be enclosed in single quotes. For example, the following function definition creates a field that shows the short description, followed by a space, followed by the caller name: `functionDefinition: 'glidefunction:concat(short_description, ' ', caller_id.name)'` For more information about function definitions, see Function field.
dynamicValueDefinitions	Object	Default values that are generated dynamically based on dynamic filters. Provide a combination of a type and a related behavior key to specify dynamic defaults. The following types are supported: • dynamic_default: Provide a function from the Dynamic Filter Options [sys_filter_option_dynamic] table. For more information, see Create a dynamic filter option. For example: `dynamicValueDefinitions: { type: 'dynamic_default', dynamicDefault: `gs.info()`, },` • dependent_field: Provide another column name from the same table. For example: `dynamicValueDefinitions: { type: 'dependent_field', columnName: 'status', },` • calculated_value: Provide a function for calculating the value. The function can be imported from a JavaScript module or be defined inline. For example: `dynamicValueDefinitions: { type: 'calculated_value', calculatedValue: function, },` • choices_from_other_table: Provide choices from a column on another table. For example: `dynamicValueDefinitions: { type: 'choices_from_other_table', table: 'sc_cat_item',`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		`field: 'display',` `},`
dropdown	String	An option for how a list of choices displays for list and form views of the table. This property only applies to *`ChoiceColumn`* objects and column types that extend choice columns. Valid values: • none: The choices aren't enforced. • dropdown_without_none: A menu without the **-- None --** option. If you select this option, you must configure the *`default`* property for the column. • dropdown_with_none: A menu with the **-- None --** option. The default value is **-- None --**. • suggestion: Choices are displayed in a list of suggested values. Default: none

Column names are provided as object keys paired with the column definitions.

```
schema: {
   deadline: DateColumn({ label: 'Deadline' }),
   state: StringColumn({
      label: 'State',
      choices: {
         ready: { label: 'Ready' },
         completed: { label: 'Completed' },
         inProgress: { label: 'In Progress' },
      }
   }),
   task: StringColumn({ label: 'Task', maxLength: 120 }),
}
```

If the table name doesn't include the application scope, column names must be prefixed with the application scope instead.

```
schema: {
   x_scope_myColumn: StringColumn({...})
}
```

choices object

Configure choices [sys_choice] for a column in a table.

The *`choices`* object is a property within the *`Column`* object. Use the *`choices`* object with supported column types in the *`schema`* property of a *`Table`* object. Only certain column types extend the choice column type (*`ChoiceColumn`*) and can include choices.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties

Name	Type	Description
label	String	Required. The text to display for the choice in the list.
dependentValue	String	A value that you map to the *dependentField* in the *dynamicValueDefinitions* property of the *Column* object.
hint	String	A short description of the choice that displays as tooltip when hovering over it.
language	String	The BCP 47 code of the language for the translated choice. Default: en
sequence	Integer	The order in the list of choices that a choice occurs.
inactive	Boolean	Flag that indicates whether to show the choice in the list. Valid values: • true: The choice is hidden from the list. • false: The choice appears in the list. Default: false

The `choices` object includes a series of choice objects, where the names of the choices are provided as object keys paired with the choices definitions.

```
choices: {
   choice1: {
      label: 'Choice1 Label',
      sequence: 0,
      inactiveOnUpdate: false,
      dependentValue: '5',
      hint: 'hint',
      inactive: false,
      language: 'en',
   },
   choice2: { label: 'Choice2 Label', sequence: 1 },
}
```

label object

Configure a field label [sys_documentation] for a table or column.

The *label* object is a property within the *Table* and *Column* objects.

Properties

Name	Type	Description
language	String	The BCP 47 code of the language for the field label. A language can have only one label, so each language must be unique within an array of *label* objects.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
label	String	The text of the field label in the specified language.
hint	String	A short description that displays as a tooltip when hovering over the field label.
help	String	Additional information about the field. Help text isn't displayed in form or list views of the table.
plural	String	The plural form of the field label.
url	String	A URL for a web page that provides information about the field. When a URL is provided, the label displays as a hyperlink.
urlTarget	String	Not used (deprecated).

```
label: [
   {
      label: 'English description',
      language: 'en',
      hint: 'Provide a short description'
   },
   {
      label: 'Description de español',
      language: 'es'
   },
]
```

licensingConfig object

Create a licensing configuration [ua_table_licensing_config] to track subscription counts for a table.

The *licensingConfig* object is a property within the *Table* object. If this property isn't specified, a default licensing configuration with *licenseModel* set to none is generated for the table on the instance.

Note: Specifying a licensing model is not applicable for ServiceNow customers who build custom applications for their own use. Licensing models are used only by partners who sell and monitor the usage of resellable applications on the ServiceNow Store.

Properties

Name	Type	Description
licenseModel	String	The model for tracking subscription usage. Valid values: • none: Licensing isn't used for the table. • fulfiller: Fulfiller/requester operations are tracked. This model applies to applications in which users open requests and fulfillers address them. Fulfillment is determined by insert, update, and delete operations on records in one or more key tables in the application

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		under a set of specified conditions. For more information, see Fulfillment tables. • producer: Producer operations are tracked. This model applies to applications in which users can perform insert, update, and delete operations on a table without identifying requesters and fulfillers. Default: none
licenseRoles	Array	A list of roles for which any operations on records in the table count toward the subscription.
opDelete	Boolean	Flag that indicates whether a subscription is required to delete records for tables with the producer model. Valid values: • true: A subscription is required to delete records in the table. • false: A subscription isn't required to delete records in the table. Default: true
opInsert	Boolean	Flag that indicates whether a subscription is required to insert records for tables with the producer model. Valid values: • true: A subscription is required to insert records in the table. • false: A subscription isn't required to insert records in the table. Default: true
opUpdate	Boolean	Flag that indicates whether a subscription is required to update records for tables with the producer model. • true: A subscription is required to update records in the table. • false: A subscription isn't required to update records in the table. Default: true
licenseCondition	String	A filter query that determines conditions for counting operations toward a subscription. For the fulfiller model, specify the set of conditions that determine whether the logged-in user is the fulfiller of the record.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		For the producer model, specify the set of conditions that determine whether records count toward the subscription.
ownerCondition	String	A filter query that determines whether a user owns a record for the fulfiller model.
isFulfillment	Boolean	Not used (deprecated). Flag that indicates whether to disallow updates by users who aren't subscribed to the application. Valid values: • true: Users who aren't subscribed to the application can't make updates to the table. • false: Users who aren't subscribed to the application can make updates to the table. Default: false

```
licensingConfig: {
  licenseModel: 'fulfiller',
  opInsert: false,
  licenseRoles: ['admin'],
}
```

autoNumber object

Configure auto-numbering [sys_number] for a table.

The `autoNumber` object is a property within the `Table` object. To use the number in a table, you must create a number column, such as `IntegerColumn`, that uses the number as the default value.

Properties

Name	Type	Description
prefix	String	A prefix for every record number in the table. For example, INC for Incident. Default: pre
number	Integer	The base record number for this table. Record numbers are automatically incremented, and the next number is maintained in the Counter [sys_number_counter] table. If you set the base number to a value higher than the current counter, the next record number uses the new base number. Otherwise the next record number uses the current counter. The counter doesn't reset to a base number lower than itself. Default: 1000

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
numberOfDigits	Integer	The minimum number of digits to use after the prefix. Leading zeros are added to auto-numbers, if necessary. For example, INC0001001 contains three leading zeros. The number of digits can exceed the minimum length. For example, if *numberOfDigits* is 2 and more than 99 records are created on the table, the numbers continue past 100 (such as INC101). **Warning:** Changing this field can update all number values for existing records on a table. Take care when changing this field on a production instance. Default: 7

```
autoNumber: {
   prefix: 'TODO',
   number: 2000,
   numberOfDigits: 9,
}
```

In the following example, a number column uses the auto-numbering as the default value. For example:

```
number: IntegerColumn({
   default: 'javascript:getNextObjNumberPadded();'
})
```

UI Action API - ServiceNow Fluent

The UI Action API defines custom user interface (UI) actions [sys_ui_action], such as buttons, links, and context menu items on forms and lists.

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

For general information about UI actions, see Create a UI action.

Related topics

ServiceNow Fluent

UiAction object

Create a UI action [sys_ui_action] to display on a form.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Format: `Now.ID['String' or Number]`
table	String	Required. The name of the table on which the UI action is available. By default, the UI action also appears on tables that extend the selected table. For example, Task actions appear on the Incident table. Set the value to `global` to make the action available on all tables.
name	String	Required. The text that appears on the button, link, or context menu item. The name must be unique within the table specified.
actionName	String	A unique name to use when referencing the UI action in scripts.
active	Boolean	Flag that indicates whether the UI action is enabled. Valid values: • true: The UI action appears and executes. • false: The UI action is hidden. Default: true
form	Object	Options for how UI actions appear on forms. For more information, see form object.
list	Object	Options for how UI actions appear on the list view. For more information, see list object.
client	Object	Options to execute the script in the browser. For more information, see client object.
workspace	Object	Options for how UI actions function and appear in workspaces. For more information, see workspace object.
overrides	Reference or String	The name or variable identifier of another UI action that the UI action overrides.
showInsert	Boolean	Flag that indicates whether to show a button on new records before they're inserted. Valid values: • true: The button appears on records before they're inserted. • false: The button doesn't appear on records that haven't been inserted. Default: false
showUpdate	Boolean	Flag that indicates whether to show a button on existing records.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Valid values: • true: The button appears on existing records. • false: The button doesn't appear on existing records. Default: true
showQuery	Boolean	Flag that indicates whether the UI action is visible on a list when a filter query is applied. Valid values: • true: The UI action appears when a filter query is applied. • false: The UI action doesn't appear a filter query is applied. Default: false
showMultipleUpdate	Boolean	Flag that indicates whether to show a button when multiple records are selected. Valid values: • true: The button appears when multiple records are selected. • false: The button doesn't appear when multiple records are selected. Default: false
condition	String	A JavaScript conditional statement that specifies the fields and values that must be true for the script to run. **Note:** • Don't use this property if you include the condition statement with the *`script`* property. • The current object isn't available for conditions on a list context menu. If the list's *`showContextMenu`* property is true, any use of `current` on these actions is ignored. • You can reference the parent record for the UI action conditions on a related list button. For example, to disable the **New** and **Edit** buttons on the Affected CIs related list for closed changes, copy the global m2m UI actions to the task_ci table and add a condition of *`parent.active`*. • If you leave one of the fields that you specify in your condition statement empty, that condition defaults to true.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Format: • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide an inline script, use string literals or template literals for multiple lines of code: `'Script'` or `` `Script` ``.
script	Script	A client-side or server-side script that runs when the UI action is executed. Function names must be unique. This property supports inline JavaScript or a reference to another file in the application that contains a script. Format: • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide an inline script, use string literals or template literals for multiple lines of code: `'Script'` or `` `Script` ``.
comments	String	Internal notes about the UI action.
messages	String	Text strings that the UI action can use as a key to look up a localized message alternative from the Message [sys_ui_message] table. Each message key is on a separate line in the Messages field. The instance looks for a localized message string anytime the UI action makes a *`getmessage('[message]')`* call where the message string matches a key in the Messages field. For more information, see Translate a client script message.
hint	String	A short description of the UI action that displays as tooltip when hovering over it.
order	Number	The order in which the UI action appears. The order applies to buttons from left to right and to menu actions from top to bottom. Default: 100
isolateScript	Boolean	Flag that indicates whether the script runs in strict mode, with access to direct DOM, jQuery, prototype, and the window object turned off. Valid values:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		• true: Isolate the script and don't run it in strict mode. • false: Run the script in strict mode. Default: false
roles	Array	A list of variable identifiers of *`Role`* objects or names of roles required for the UI action to apply. For more information, see Role API - ServiceNow Fluent.
includeInViews	Array	A list of names of views in which the UI action is included.
excludeFromViews	Array	A list of names of views from which the UI action is excluded.
$meta	Object	Metadata for the application metadata. With the *`installMethod`* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }` Valid values for *`installMethod`*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
import { UiAction } from '@servicenow/sdk/core'

UiAction({
    $id: Now.ID['car_info'],
    table: 'x_snc_ts_custom_cars',
    actionName: 'Car Information',
    name: 'View car info',
    active: true,
    showInsert: true,
    showUpdate: true,
    hint: 'View car info',
    condition: "current.type == 'SUV'",
    form: {
        showButton: true,
        showLink: true,
        showContextMenu: false,
        style: 'destructive',
    },
    list: {
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
        showLink: true,
        style: 'primary',
        showButton: true,
        showContextMenu: false,
        showListChoice: false,
        showBannerButton: true,
        showSaveWithFormButton: true,
    },
    workspace: {
        isConfigurableWorkspace: true,
        showFormButtonV2: true,
        showFormMenuButtonV2: true,
        clientScriptV2: `function onClick(g_form) {
                        }`,
    },
    script: `current.name =  "updated by script";
                current.update();`,
    roles: ['u_requestor'],
    client: {
        isClient: true,
        isUi11Compatible: true,
        isUi16Compatible: true,
        },
    order: 100,
    showQuery: false,
    showMultipleUpdate: false,
    isolateScript: false,
    includeInViews: ['specialView'],
    excludeFromViews: [],
})
```

form object

Configure how a UI action appears on a form.

The *form* object is a property within the *UiAction* object.

Properties

Name	Type	Description
showButton	Boolean	Flag that indicates whether to include a button on a form. Valid values: • true: A button appears on forms. • false: A button doesn't appear on forms. Default: false
showLink	Boolean	Flag that indicates whether to include a link in the Related Links section of a form. Valid values:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		• true: A link appears in the Related Links section. • false: A link doesn't appear in the Related Links section. Default: false
showContextMenu	Boolean	Flag that indicates whether to include an item in the context menu of a form. Valid values: • true: An item appears in the context menu. • false: An item doesn't appear in the context menu. Default: false
style	String	A style that defines how UI action buttons appear on a form. Valid values: • primary: Colors the UI action blue. • destructive: Colors the UI action red. • unstyled: The UI action isn't styled.
$meta	Object	Metadata for the application metadata. With the *installMethod* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }` Valid values for *installMethod*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
form: {
  showButton: true,
  showLink: true,
  showContextMenu: false,
  style: 'destructive',
}
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

list object

Configure how a UI action appears on the list view.

The `list` object is a property within the `UiAction` object.

Properties

Name	Type	Description
showButton	Boolean	Flag that indicates whether to include a button at the bottom of a list. **Note:** Buttons at the bottom of a list appear regardless of condition and are evaluated per record on execution. Valid values: • true: A button appears at the bottom of lists. • false: A button doesn't appear at the bottom of lists. Default: false
showLink	Boolean	Flag that indicates whether to include a link in the Related Links section of a list. Valid values: • true: A link appears in the Related Links section. • false: A link doesn't appear in the Related Links section. Default: false
showContextMenu	Boolean	Flag that indicates whether to include an item in the context menu of a list. Valid values: • true: An item appears in the context menu. • false: An item doesn't appear in the context menu. Default: false
style	String	A style that defines how UI action buttons appear on the list view. Valid values: • primary: Colors the UI action blue. • destructive: Colors the UI action red. • unstyled: The UI action isn't styled.
showListChoice	Boolean	Flag that indicates whether to include a choice in the Actions list of a list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Note: Choices in the Actions list appear regardless of condition and are evaluated per record on execution. Valid values: • true: A choice appears in the Actions list. • false: A choice doesn't appear in the Actions list. Default: false
showBannerButton	Boolean	Flag that indicates whether to include a button on the banner of a list. **Note:** Buttons on the banner of a list aren't intended to support record-specific conditions. Only the first row is considered when the condition is evaluated to determine whether the button appears for the list. Don't use record-specific conditions, such as `current.getValue('state') === 'closed'`. Valid values: • true: A button appears on the banner of lists. • false: A button doesn't appear on the banner of lists. Default: false
showSaveWithFormButton	Boolean	Flag that indicates whether the form is saved when accessed from a list before executing the UI action button. Valid values: • true: The form must be saved before the UI action executes. • false: The form doesn't have to be saved before the UI action executes. Default: false
$meta	Object	Metadata for the application metadata. With the *`installMethod`* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Valid values for *installMethod*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
list: {
  showButton: true,
  showLink: true,
  showContextMenu: false,
  style: 'primary',
  showListChoice: false,
  showBannerButton: true,
  showSaveWithFormButton: true,
}
```

client object

Configure options to execute the UI action script in the browser.

The *client* object is a property within the *UiAction* object.

Properties

Name	Type	Description
isClient	Boolean	Flag that indicates where the UI action script runs. Valid values: • true: The script runs on the client (the user's browser). • false: The script runs on the server. Default: false
isUi11Compatible	Boolean	Flag that indicates whether the UI action is supported in the legacy UI 11. Valid values: • true: The UI action is supported in the legacy UI 11. • false: The UI action isn't supported in the legacy UI 11. Default: false
isUi16Compatible	Boolean	Flag that indicates whether the UI action is supported in the Core UI. Valid values:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		• true: The UI action is supported in the Core UI. • false: The UI action isn't supported in the Core UI. Default: false
onClick	String	The name of the JavaScript function to run when the UI action is executed. The function is defined with the *`script`* property.
$meta	Object	Metadata for the application metadata. With the *`installMethod`* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }` Valid values for *`installMethod`*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
client: {
  isClient: true,
  isUi11Compatible: true,
  isUi16Compatible: true,
  onClick: 'reopenIncident()'
}
```

workspace object

Configure how a UI action functions and appears in workspaces.

The *`workspace`* object is a property within the *`UiAction`* object.

Properties

Name	Type	Description
isConfigurableWorkspace	Boolean	Flag that indicates the type of workspace in which a UI action applies. Valid values: • true: The UI action applies to Configurable Workspaces. • false: The UI action applies to Legacy Workspaces.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Default: false
showFormButtonV2	Boolean	Flag that indicates whether to include a button on forms in a workspace. Valid values: • true: A button appears on forms. • false: A button doesn't appear on forms. Default: false
showFormMenuButtonV2	Boolean	Flag that indicates whether to include an item in the More Actions menu in a workspace. Valid values: • true: An item appears in the More Actions menu. • false: An item doesn't appear in the More Actions menu. Default: false
clientScriptV2	String	A script that runs when the UI action is executed in workspaces. This property supports inline JavaScript or a reference to another file in the application that contains a script. Format: • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide an inline script, use string literals or template literals for multiple lines of code: `'Script'` or `` `Script` ``.
$meta	Object	Metadata for the application metadata. With the *`installMethod`* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }` Valid values for *`installMethod`*:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		• demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

```
workspace: {
  isConfigurableWorkspace: true,
  showFormButtonV2: true,
  showFormMenuButtonV2: true,
  clientScriptV2: `function onClick(g_form) {
  }`,
}
```

UI Page API - ServiceNow Fluent

The UI Page API defines custom user interface (UI) pages [sys_ui_page] that display forms, dialogs, lists, and other UI components.

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

A UI page displays as a web page and can be added to a widget for use in dashboards. For general information about UI pages, see UI pages.

You can develop a simple React application with the UI Page API. In the `src/client` directory, add static content files that define the HTML, client script, and styling of the page. From the *`UiPage`* object, refer to the page's HTML entry point (`index.html`). For more information, see User interface development with React.

Related topics

ServiceNow Fluent

UiPage object

Create a UI page [sys_ui_page] for a custom user interface.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
endpoint	String	Required. The endpoint to access the web page. The endpoint value can't contain spaces.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Format: `<scope>_<ui_page_name>.do`
description	String	A description of the user interface and its purpose.
direct	Boolean	Flag that indicates whether to omit the standard HTML, CSS, and JavaScript for a UI page. For React UI pages, set this property to `true`. Valid values: • true: Omit the standard HTML, CSS, and JavaScript and provide custom CSS and JavaScript for the page. • false: Include the standard HTML, CSS, and JavaScript. Default: false
html	String	The body HTML code that defines what is rendered when the page is shown. It can contain either static XHTML, dynamically generated content defined as Jelly, or call script includes and UI Macros. This property supports an alias to import an `index.html` file for React development, a reference to another file in the application that contains HTML, or inline JavaScript. **Note:** With React development and imported HTML, the *`html`* property of the *`UiPage`* object supports only one-way synchronization. After defining the HTML of a UI page in source code, if the HTML is modified outside of the source code, those changes aren't synchronized and reflected in the source code. Format: • An alias to import the `index.html` file from the previous build for UI development with React. The alias can be any value and must also be imported in the `now.ts` file that contains the *`UiPage`* object: `import <alias> from 'path/to/index.html'`. • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide inline HTML, use string literals or template literals for multiple lines of code: `'HTML' or` `` `HTML` ``.
category	String	The type of UI page. Valid values:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		• general: The page is general purpose. • homepages: The page is used as a home page. • htmleditor: The page is used to insert HTML content. • kb: The page is used with a Knowledge Base. • cms: The page is used with the Content Management System (CMS). • catalog: The page is used with Service Catalog.
clientScript	Script	A script that runs in the browser, such as functions called by buttons. This script handles any required client-side processing needed, like setting focus to a field or other interactive DHTML features after a page is loaded. This property supports inline JavaScript or a reference to another file in the application that contains a script. Client scripts for the UI page are deployed to the browser within a `<script>` tag, so the content can similarly be defined within the HTML field. The *`clientScript`* property can be used instead to define these scripts concisely to maintain the Jelly and HTML manageability. Format: • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more information, see ServiceNow Fluent language constructs. • To provide an inline script, use string literals or template literals for multiple lines of code: `'Script'` or `` `Script` ``.
processingScript	Script	A script that runs on the server when the page is submitted, which is useful if your page has a form defined with the `<g:ui_form/>` or `<g:form/>` tags. This property supports a function from a JavaScript module, a reference to another file in the application that contains a script, or inline JavaScript. Format: • For functions, use the name of a function, function expression, or default function exported from a JavaScript module and import it into the `.now.ts` file. For information about JavaScript modules, see JavaScript modules and third-party libraries. • To use text content from another file, refer to a file in the application using the following format: `Now.include('path/to/file')`. For more

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		information, see ServiceNow Fluent language constructs. • To provide an inline script, use string literals or template literals for multiple lines of code: `'Script'` or `` `Script` ``.
$meta	Object	Metadata for the application metadata. With the *`installMethod`* property, you can map the application metadata to an output directory that loads only in specific circumstances. `$meta: { installMethod: 'String' }` Valid values for *`installMethod`*: • demo: Outputs the application metadata to the `metadata/unload.demo` directory to be installed with the application when the **Load demo data** option is selected. • first install: Outputs the application metadata to the `metadata/unload` directory to be installed only the first time an application is installed on an instance.

In this example of UI development with React, the HTML of the page is imported from the `index.html` file in the `src/client` directory.

```
//incident-manager.now.ts
import '@servicenow/sdk/global'
import { UiPage } from '@servicenow/sdk/core'
import incidentPage from '../../client/index.html'

UiPage({
    $id: Now.ID['incident-manager-page'],
    endpoint: '<scope>_incident_manager.do',
    description: 'Incident Response Manager UI Page',
    category: 'general',
    html: incidentPage,
    direct: true,
})
```

In the `index.html` file, the `<script>` tag refers to the `main.jsx` file in the `src/client` directory, which contains the component code.

```
//index.html
<html>
<head>
  <title>Incident Response Manager</title>

  <!-- Initialize globals and Include ServiceNow's required scripts -->
  <sdk:now-ux-globals></sdk:now-ux-globals>
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    <!-- Include your React entry point -->
    <script  src="./main.jsx"  type="module"></script>
</head>
<body>
    <div  id="root"></div>
</body>
</html>
```

UI Policy API - ServiceNow Fluent

The UI Policy API defines user interface (UI) policies [sys_ui_policy] that dynamically change the behavior of information on a form and control custom process flows for tasks.

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

UI Policies can make fields mandatory, read-only, visible, hidden, or cleared when certain conditions are met. You can also use client scripts to perform all of these actions, but for faster load times use UI policies when possible.

For general information about UI policies, see Using UI policies.

Related topics

ServiceNow Fluent

UiPolicy object

Create a UI policy sys_ui_policy to configure form behavior.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
table	String	Required. The table of the form to modify.
shortDescription	String	Required. A description of the policy and its purpose.
active	Boolean	Flag that indicates whether the policy is applied to the form. Default: true
global	Boolean	Flag that indicates whether to which form views the policy applies. If true, the policy applies to all views of the table. If false, the policy is specific to the form view specified with the *`view`* property. Default: true
onLoad	Boolean	Flag that indicates when the policy executes. If true, the policy runs every time a form is loaded if the conditions are satisfied. Default: true

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
reverseIfFalse	Boolean	Flag that indicates whether to invert the policy behavior when the condition evaluates to false. If true, the policy action is undone when the conditions of its policy evaluate to false. Default: true
inherit	Boolean	Flag that indicates whether tables that extend the current table inherit the policy. If true, extended tables inherit the policy. When a child table has an inherited policy from its parent table, the policy on the child table runs first. This event is true regardless of the order of the policies. Default: false
isolateScript	Boolean	Flag that indicates whether to run scripts in isolated scope. If true, the script runs in isolated scope. This property only applies if the *runScripts* is set to true. Default: false
conditions	String	A filter query that specifies the fields and values that must be true for users to access the object. For more information, see Operators available for filters and queries. To set conditions using a script, use a client script instead. Conditions are only rechecked if a user manually changes a field on a form. If the change is made by a UI action, context menu action, or through the list editor, it isn't evaluated.
runScripts	Boolean	Flag that indicates whether advanced behavior can be scripted for both true and false conditions. If true, scripts defined with the *scriptTrue*, *scriptFalse*, *uiType*, and *isolateScript* properties run when applicable. Default: false
scriptTrue	String	Client-side script that runs if the conditions of the policy are met. This property is required if the *runScripts* property is set to true. Format: `function onCondition() {}` Default: `function onCondition() {\n\n}`
scriptFalse	String	Client-side script that runs if the conditions of the policy aren't met and the *reverseIfFalse* property is set to true. This property is required if the *runScripts* property is set to true. Format: `function onCondition() {}` Default: `function onCondition() {\n\n}`
uiType	String	The type of user interface to which the policy applies. This property is required if the *runScripts* property is set to true.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Valid values: • desktop: The policy applies to desktop interfaces. • mobile-or-service-portal: The policy applies to mobile and Service Portal interfaces. • all: The policy applies to all interfaces. Default: desktop
actions	Array	A list of field actions to apply if the conditions are met. For more information, see actions array.
relatedListActions	Array	A list of visibility controls for related lists. For more information, see relatedListActions array.
description	String	Additional information about the policy.
modelId	String	The sys_id of the parent UI policy to which the policy applies when this policy is inherited. This property works in conjunction with *model_table* property.
modelTable	String	The name of the parent table to which the policy applies when a UI policy is inherited from a parent table. This property works in conjunction with the *model_id* property.
order	Number	The execution order in which to apply policies if more than one policy fits the conditions. For inherited UI policies, the extended table's policies are executed first. Then the base table policies are executed. Default: 100
setValues	String	(Deprecated) The field values to set using an encoded string format. Use UI policy actions [sys_ui_policy_action] instead.
view	Reference or String	Required. The variable identifier or name of the UI view [sys_ui_view] which applies, or the default view. To define a UI view, use the Record API - ServiceNow Fluent. To use the default view (`default_view`), you must import it: `import { default_view } from '@servicenow/sdk/core'`

```
import { UiPolicy } from '@servicenow/sdk/core';

export const securityIncidentPolicy = UiPolicy({
  $id: Now.ID['security_incident_policy'],
  table: 'incident',
  shortDescription: 'Lock critical fields for security incidents',
  active: true,
  onLoad: true,
  conditions: 'category="security"',
  actions: [
    {
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
      field: 'security_notes',
      mandatory: true,
      visible: true
    },
    {
      field: 'caller_id',
      readOnly: true   // Lock caller field when category is security
    },
    {
      field: 'assignment_group',
      readOnly: true   // Lock assignment group when category is security
    },
    {
      field: 'priority',
      readOnly: true   // Lock priority when category is security
    }
  ],
  relatedListActions: [
    {
      // Using plain GUID for system relationships
      list: 'b9edf0ca0a0a0b010035de2d6b579a03',  // Attachments
      visible: false
    },
    {
      // Using table.field format for reference fields
      list: 'x_snc_17sepapp1_expenseitem.expensereport',  // Example reference field
      visible: true
    }
  ]
})
```

actions array

Configure the actions [sys_ui_policy_action] that the UI policy performs on fields.

Use the *`actions`* array within the *`UiPolicy`* object. Actions are processed in the order that they appear in the array. At least one of the `visible`, `readOnly`, `mandatory`, or `cleared` properties must be specified for each action in the array.

Properties

Name	Type	Description
field	String	Required. The name of the field to which the action applies. **Note:** If the specified field isn't found on the form, the UI policy performs the action on the variable with the same name.
visible	Boolean or String	An option to control the visibility of the field. Valid values: • true: The field is visible. • false: The field is hidden. • 'ignore': The field isn't changed.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		Default: ignore
readOnly	Boolean or String	An option to control access to edit the field. Valid values: • true: The field is read-only. • false: The field is editable. • 'ignore': The field isn't changed. Default: ignore
mandatory	Boolean or String	An option to control whether the field is required. Valid values: • true: The field is required. • false: The field is optional. • 'ignore': The field isn't changed. Default: ignore
cleared	Boolean	Flag that indicates whether the field should be cleared if the conditions of the policy are met. Default: false
table	String	The table to which the action applies, which overrides the table specified by the policy. If empty, the table specified by the policy applies.
value	String	The value to set the field to if the policy conditions are met.
fieldMessage	String	A message to display about the field if the policy conditions are met.
fieldMessageType	String	A message type that determines how the field message is presented. Valid values: • error • info • warning • none Default: none
valueAction	String	An action to perform on the field value. Valid values:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		• set_value • clear_value • ignore Default: ignore

```
actions: [
        {
            field: 'assignment_group',
            mandatory: true,
            value: 'Critical Response Team',  // Set default value
            fieldMessage: 'This incident requires immediate attention from the
Critical Response Team',
            fieldMessageType: 'error'  // Show as error message
        },
        {
            field: 'urgency',
            value: '1',  // Set to High urgency
            fieldMessage: 'Urgency has been automatically set to High',
            fieldMessageType: 'info'
        },
        {
            field: 'impact',
            value: '1',  // Set to High impact
            fieldMessage: 'Impact has been automatically set to High',
            fieldMessageType: 'warning'
        }
    ]
```

relatedListActions array

Configure the visibility of related lists [sys_ui_policy_rl_action] on a form for a UI policy.

Use the `relatedListActions` array within the `UiPolicy` object. Related list actions are processed in the order that they appear in the array. Either the `list` or `visible` property must be specified for each related list action in the array.

Properties

Name	Type	Description
list	String	A reference to a related list on the form. If empty, the action applies to all related lists. This property is required if the `visible` property is not set. Format:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties (continued)

Name	Type	Description
		• sys_ID: The sys_ID for the list. Use this format for custom queries, attachments, and system-defined related lists. • table.field: The table and field name associated for the list, such as `'incident.caller_id'`. Use this format for reference field-based relationships.
visible	Boolean or String	An option to control the visibility of the related list. This property is required if the *`list`* property is not set. Valid values: • true: The related list is visible. • false: The related list is hidden. • 'ignore': The related list isn't changed. Default: ignore

```
relatedListActions: [
    {
      // Using plain GUID for system relationships
      list: 'b9edf0ca0a0a0b010035de2d6b579a03',  // Attachments
      visible: false
    },
    {
      // Using table.field format for reference fields
      list: 'x_snc_17sepapp1_expenseitem.expensereport',  // Example reference field
      visible: true
    }
  ]
```

Workspace API - ServiceNow Fluent

The Workspace API defines configurable workspace experiences for organizing and sharing data visually.

Note: For the latest ServiceNow Fluent API documentation and examples, see the ServiceNow Fluent API reference and ServiceNow SDK examples repository on GitHub.

The Workspace API creates application metadata in the following tables depending on the workspace definition: UX Application [sys_ux_page_registry], UX App Configuration [sys_ux_app_config], UX Application Category M2M [sys_ux_registry_m2m_category], UX Page Property [sys_ux_page_property], UX Screen Collection [sys_ux_screen_type], UX App Route [sys_ux_app_route], UX Screen [sys_ux_screen], and UX Macroponent Definition [sys_ux_macroponent].

Dashboards can be used as the home page of a workspace by referring to one or more workspaces from the *`visibilities`* array of the *`Dashboard`* object. To create a dashboard, see Dashboard API - ServiceNow Fluent.

For general information about workspaces, see Configurable Workspace UI.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Related topics

ServiceNow Fluent

Workspace object

Create a workspace for managing business entities in a single focused working area that enables users to complete an entire job.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
title	String	Required. A name for the workspace that appears in navigation and headers.
path	String	Required. The URL path segment of the workspace. Workspace URLs follow the pattern `/now/<path>/<landingPath>` and use kebab case. Workspaces require access control lists (ACLs) to secure the workspace routes. The *`field`* property of an *`Acl`* object must match the value of this property with a wildcard pattern: `{workspace.path}.*`.
tables	Array	Required. A list of table names to manage in the workspace.
listConfig	Reference	Required. The variable identifier of a *`UxListMenuConfig`* object that defines the navigation structure of the workspace. For more information, see UxListMenuConfig object.
landingPath	String	The URL path segment of the landing page. Workspace URLs follow the pattern `/now/<path>/<landingPath>` and use kebab case. Default: home
active	Boolean	Flag that indicates whether the workspace is accessible to users. Default: true

```
import { Workspace } from '@servicenow/sdk/core';

const itsmWorkspace = Workspace({
    $id: Now.ID['itsm_workspace'],
    title: 'IT Service Management',
    path: 'itsm',
    tables: ['incident', 'problem', 'change_request', 'user', 'sys_user_group'],
    listConfig: incidentListConfig
})
```

The UX List Menu Configuration referenced is defined using the *`UxListMenuConfig`* object.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

UxListMenuConfig object

Define a UX list menu configuration [sys_ux_list_menu_config] for the navigation structure and list views of a workspace.

A UX list menu configuration organizes data into categories and lists, enabling users to access different views of business data with filtering, column selection, and role-based visibility. This structure appears in the workspace's navigation panel, providing organized access to different data views.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
name	String	Required. A name for the list configuration.
description	String	A description of the list configuration.
active	Boolean	Flag that indicates whether the list configuration is active. Default: true
categories	Array	A list of top-level groupings in the list configuration. For more information, see categories array.

```
import { UxListMenuConfig } from '@servicenow/sdk/core';

const incidentListConfig = UxListMenuConfig({
    $id: Now.ID['incident_list_config'],
    name: 'Incident List Configuration',
    description: 'Navigation for Incident Workspace',
    categories: [
        {
            $id: Now.ID['incidents_category'],
            title: 'Incidents',
            order: 10,
            lists: [
                {
                    $id: Now.ID['incidents_open'],
                    title: 'Open',
                    order: 10,
                    condition: 'active=true^EQ',
                    table: 'incident',
                    columns: 'number,short_description,priority,state',
                    applicabilities: [
                        {
                            $id: Now.ID['incidents_open_applicability'],
                            applicability: applicability
                        }
                    ]
                }
            ]
        }
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    ]
})
```

categories array

Define categories of related lists [sys_ux_list_category] for a UX list menu configuration.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
title	String	Required. A title for the category to display in the navigation menu.
lists	Array	Required. A list of list views in the category. For more information, see lists array.
order	Number	A number indicating the position of the category in the navigation menu. Categories with lower numbers appear first.
active	Boolean	Flag that indicates whether the category is visible in the navigation menu. Default: true
description	String	A description of the category.

```
categories: [
    {
      $id: Now.ID["incidents_category"],
      title: "Incidents",
      order: 10,
      lists: [
        {
          $id: Now.ID["incidents_open"],
          title: "Open",
          order: 10,
          condition: "active=true^EQ",
          table: "incident",
          columns: "number,short_description,priority,state",
          applicabilities: [
            {
              $id: Now.ID["incidents_open_applicability"],
              applicability: applicability
            }
          ]
        },
        {
          $id: Now.ID["incidents_all"],
          title: "All",
          order: 20,
          condition: "",
          table: "incident",
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
        columns: "number,short_description,priority,state",
        applicabilities: [
          {
            $id: Now.ID["incidents_all_applicability"],
            applicability: applicability
          }
        ]
      }
]
```

lists array

Define list views of table data [sys_ux_list] with filtering and column configurations for a UX list menu configuration.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
title	String	Required. A title for the list to display in the navigation menu.
table	String	Required. The name of a table to use for the list.
columns	String	A comma-separated list of column names to display in the list.
condition	String	An encoded query string to filter the records displayed in the list.
order	Number	A number indicating the position of the list within its category. Lists with lower numbers appear first.
active	Boolean	Flag that indicates whether the list is visible to users. Default: true
applicabilities	Array	A list of variable identifiers of *`Applicability`* objects that control which roles can view the list. For more information, see Applicability object.

```
lists: [
      {
        $id: Now.ID["assets_active"],
        title: "Active",
        order: 10,
        condition: "install_status=1",
        table: "alm_asset",
        columns: "asset_tag,display_name,model_category,assigned_to",
        applicabilities: [
          {
            $id: Now.ID["assets_active_applicability"],
            applicability: assetApplicability
          }
        ]
      },
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
        {
          $id: Now.ID["assets_all"],
          title: "All",
          order: 20,
          condition: "",
          table: "alm_asset",
          columns: "asset_tag,display_name,model_category,assigned_to",
          applicabilities: [
            {
              $id: Now.ID["assets_all_applicability"],
              applicability: assetApplicability
            }
          ]
  }
]
```

Applicability object

Define the audience [sys_ux_applicability] that can view a list in the UX list menu configuration.

Properties

Name	Type	Description
$id	String or Number	Required. A unique ID for the metadata object. When you build the application, this ID is hashed into a unique sys_id. For more information, see ServiceNow Fluent language constructs. Format: `Now.ID['String' or Number]`
name	String	Required. A name for the applicability rule.
description	String	A description of the audience.
active	Boolean	Flag that indicates whether the applicability rule is applied. Default: true
roles	Array	A list of variable identifiers of *`Role`* objects or sys_ids of roles that a user must have to view the list. For more information, see Role API - ServiceNow Fluent.
roleNames	String	A comma-separated list of role names that a user must have to view the list. This property is an alternative to the *`roles`* property.

```
import { Applicability } from '@servicenow/sdk/core';

const managerApplicability = Applicability({
    $id: Now.ID['manager_applicability'],
    name: 'Managers Only',
    roles: [managerRole]
})
```

The role referenced is defined using the *`Role`* object:

```
import { Role } from '@servicenow/sdk/core';

const managerRole = Role({
```

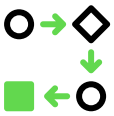
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    $id: Now.ID['manager_user_role'],
    name: 'x_snc_manager.user',
    containsRoles: ['canvas_user']
})
```

Lux Development

Lux is a framework for building modern widget-based UI experiences that run on a ServiceNow instance. Create pages, widgets and reusable components, configure experience behavior, and safely extend applications using Lux development tools and frameworks.

Get started

Explore  Learn about Lux concepts and features.	Configure  Configure experiences with Lux	Extend  Extend experiences with Lux
Build  Build with Lux	Test and Deploy  Test and deploy Lux experiences	Reference  Lux reference material

Key benefits of Lux

- Admin configuration changes survive platform upgrades
- Extend or override another app's experience without forking it or coordinating release schedules
- Users see content instantly with no spinners, and search engines can index the fully rendered markup
- Lux applies design system styling and theming automatically, letting admins override tokens per tenant without a rebuild
- Build components and widgets once and reuse across pages, apps, AI placement, and chat surfaces
- One consistent build-deploy pipeline and architecture across the CLI, Lux Lab, and Maven, regardless of app size

Exploring Lux

Lux is a framework for building widget-based UI experiences on a ServiceNow instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Lux streams the rendered HTML to the browser and hydrates it into a fully interactive single-page app.

Your app is served under `/aiux/<app-basename>/<route>`. The `aiux` prefix is the platform's public URL surface. It also appears in package names such as *`@servicenow/aiux-components-core`* and in API identifiers such as `AIUXElement`. Those are real names you type in code, and they don't change.

Faster page loading

When a user navigates to your Lux app they will see fully rendered content immediately, with no loading spinners or blank screens. Subsequent navigation happens client-side, so the app feels like a single-page app.

Lux provides the following benefits:

- The browser shows the first screen quickly.
- The markup is accessible and search engines can read it.
- Data retrieval is efficient. Loaders run on the server, near your ServiceNow instance. This helps prevent the delay of a round trip that client-side fetches cause.

Built on Lit widgets

You build all Lux UI with Lit widgets. Your widgets extend a platform base class from the package *`@servicenow/aiux-components-core`*. Use `AIUXElement` for a simpler presentational widget or a page. Never extend `LitElement` directly.

The base classes carry the SSR, styling, and data-fetching plumbing:

- Tailwind and DaisyUI styling, applied automatically through adopted stylesheets, so your widgets inherit the design system without manual imports.
- SSR-safe `this.loaderData`, holding the result of your widget's `static async loader(ctx)` method during both server render and client hydration.
- `getLayoutData(key)`, reading layout-level data shared across all pages.
- The `static async loader(ctx)` convention, a framework-managed pattern for fetching data before render.

The fundamental building blocks are described in the following table.

Lux building blocks

Building block	Description
`aiux.json`	The app manifest: app name, URL basename, landing route
Pages	Lit widgets under `pages/`; directory structure maps directly to URLs (file-based routing)
Loaders	A `static async loader(ctx)` method on a page or widget that fetches data on the server before render; the result becomes `this.loaderData`
Widgets	Reusable Lit UI units extending `AIUXWidgetElement` (discoverable, self-fetching) or `AIUXElement` (presentational)

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Lux building blocks (continued)

Building block	Description
Layout	A single `pages/layout.js` that renders the app chrome (header, navigation, footer) around each page
Context API	`createContext()` for sharing data across widgets without prop-drilling

What you build with it

You build a scoped ServiceNow application that depends on Lux. Your application lives in its own repository, separate from Lux itself. It deploys to a ServiceNow instance on its own schedule using the ServiceNow SDK.

Who this is for

Lux is for developers building custom UI experiences on a ServiceNow instance. You should know JavaScript (ES modules) and basic HTML and CSS, plus web components or a component framework. Prior Lit experience helps but isn't required.

Development tooling

Three tools can build a Lux experience: Lux Lab, ServiceNow Lux Lab for VS Code, and the ServiceNow SDK. Lux and ServiceNow Lux Lab for VS Code build on the ServiceNow SDK foundation.

Tool comparison

The tools differ in how much of the ServiceNow SDK CLI they wrap in a UI, and where that UI lives.

Tool comparison

Tool	What it is	Where it runs	Best for
Lux Lab	A ServiceNow application that combines a project explorer, code editor, instance browser, agent chat panel, source control, and build actions in one window	Standalone desktop app, downloaded from Lux Lab.	Working in a purpose-built tool, and agent-assisted page and widget authoring
Lux Lab for VS Code	An extension adding Instance Explorer, Script Runner, Quick Preview, and `Lux:` commands to VS Code	Inside Visual Studio Code	Working in VS Code with Lux tooling alongside an existing setup
ServiceNow SDK	The CLI that scaffolds, builds, and deploys a Lux application (`now-sdk init` / `build` / `install`)	Terminal, in any editor	Full control, scripting, and continuous integration (CI) pipelines

Lux Lab and ServiceNow Lux Lab for VS Code are separate products that ship side by side. Pick whichever fits your workflow. Neither is a track you graduate from.

Both wrap the ServiceNow SDK rather than replacing it. Lux Lab runs `now-sdk` commands on your behalf when you build and deploy. The VS Code extension requires ServiceNow SDK 4.12 or later to be installed.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Next steps

After you pick a tool, see Create your first Lux experience, which uses the Now SDK CLI directly.

Lux Lab

Lux Lab is the standalone desktop app for building Lux applications, not a VS Code extension. It combines a project explorer, code editor, instance browser, agent chat panel, source control, and build actions in one window.

Lux Lab is one of the tools you can use to create or extend Lux experiences. The other is Lux Lab for VS Code. Pick whichever fits your workflow; neither is a track you graduate from.

System requirements

macOS requirements

Requirement	Minimum	Recommended
Operating system	macOS 12 (Monterey) or later	macOS 13 (Ventura) or later
Processor	Apple Silicon (M1 or later)	Apple Silicon M2 or later
RAM	8 GB	16 GB or more
Disk space	2 GB free	10 GB or more free

Windows requirements

Requirement	Minimum	Recommended
Operating system	Windows 10 (64-bit) or later	Windows 11
Processor	Intel Core i5 / AMD Ryzen 5	Intel Core i7 / AMD Ryzen 7 or better
RAM	8 GB	16 GB or more
Disk space	2 GB free	10 GB or more free

You also need Node.js v24 or later, pnpm v10 or later, Git v2.30 or later, and an internet connection for instance connectivity and agent features.

Installation

Download the Lux Lab app using the following link. Lux Lab installation files.

For more information on installing Lux Lab, see Install Lux Lab.

- macOS ships as a `.dmg` you drag into `/Applications`
- Windows ships as an `.exe` installer

First launch

On the first macOS launch, the operating system may block the app. Open it from **System Settings** > **Privacy & Security** by selecting **Open Anyway**.

A guided setup screen walks you through three steps before you reach the home page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Guided setup steps

Step	What it does	Required
Environment	Checks for Node.js v24+, pnpm v10+, and Git v2.30+	Yes
Agents	Detects whether the agents the Agent Chat Harness needs are installed, and offers to install them	No
Instance sign-in	Signs in to a provisioned ServiceNow instance that can run Lux applications, using your instance URL, username, and password (basic authentication)	Yes

A failing environment check gets a **Run Fix** button that installs or resolves the issue and re-validates. All environment checks must pass before you continue. The Agents step completes automatically if the agents are already present, and you can configure agents later from the Agents and Skills panel. The home page is not accessible without signing in to an instance.

Agent support is vendor-neutral. The Agents and Skills panel lists every ACP-compatible agent configured on your system, such as Claude, Devin, or Codex. The panel also lists the skills bundled with Lux Lab and any project-level skills in your project.

Lux Lab home

Home regions

Region	Purpose
Top bar	Workspace tabs, plus the **Run**, **Preview**, and **Deploy** actions
Activity bar	Icon strip that switches between Explorer, Source Control, Search, and other views
Project Explorer	File tree for your local project
Code Editor	Editing area with tabs and a breadcrumb trail; supports split views
Agent Chat panel	Conversational agent interactions
Bottom panel	Tabbed: Terminal, Build, Console, Problems, Tests
Instance / Work Items	The connected instance and current work items, pinned in the sidebar
Status bar	Branch name, sync status, IntelliSense, and other indicators

The editor gives language-server-based, context-aware completions as you type, using the same IntelliSense engine as VS Code.

Project templates

The **Create New Project** dialog scaffolds from one of three templates, or imports an existing local folder or Git repository.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Project templates

Template	What it scaffolds
Base AIUX template	A full starter with nav layout, pages, and a hello-world widget
Extension template	An extension to an existing application
Collections template	A collections-based Lux application

Lux Lab scaffolds the project, installs dependencies, and opens it. For the same project structure explained from the CLI side, see Create your first Lux experience.

Run and preview actions

Run starts the local dev server. The button shows **Stop** while the server is up. **Preview** starts the dev server and opens the preview at the same time. Saving a source file triggers hot module replacement, which compiles only the changed module and pushes it to the preview without a full reload. Changes to configuration files such as `aiux.json` need a dev server restart.

The preview opens as an editor tab showing your application in an embedded browser frame. Lux Lab serves it on localhost and proxies it to the instance you're connected to in the app.

Dev server logs and hot-reload activity stream into the **Console** tab. Linter and type errors land in **Problems**; build failures land in **Build**.

Deployment

Lux Lab's deploy pipeline follows the ServiceNow SDK deployment process. When you trigger a deploy, Lux Lab builds the application, compiling and bundling the source into deployable artifacts. It then runs `now-sdk install` to push the built metadata and artifacts to the target instance. The target is the instance you're currently signed in to. The process is identical to running the ServiceNow SDK deployment from the command line.

The **Deploy** button in the title bar runs a full build and deploy (`Cmd+Shift+D` on macOS, `Ctrl+Shift+D` on Windows). Its drop-down menu adds three options.

Deploy options

Option	What it does
Build	Runs the build step only, without deploying
Custom Deploy	Deploys to a different instance you have previously connected to but aren't currently signed in to
Project scripts	Runs any script defined in the project's `package.json`

Deploy output streams into the **Build** tab. If several projects are open in the workspace, **Deploy** asks which one to deploy.

Before deploying, confirm that the project builds, that you're connected to the target instance, and that your account has permission to deploy applications on it.

Under **App Settings**, the ServiceNow SDK version Lux Lab uses for a project defaults to the latest. For what those commands do if you want the manual workflow instead, see ServiceNow SDK.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Privacy

Lux Lab collects only installation status, a signal used to count active installations. It doesn't collect or transmit code, project files, prompts, agent conversations, or instance credentials. To opt out, open **App Settings** > **Advanced** and turn off the installation status telemetry toggle.

Related topics

- Lux Lab for VS Code
- ServiceNow SDK

Lux Lab for VS Code

ServiceNow Lux Lab for VS Code brings project creation, instance browsing, local preview, build, and deployment into Visual Studio Code. It ships alongside the standalone Lux Lab desktop app as a separate product.

Use the Lux Lab VS Code extension, rather than the standalone Lux Lab app, when you want the same build/preview/deploy workflow inside an editor you already use. The extension adds project creation, instance browsing, local preview, build, and deployment commands to VS Code.

For the desktop app, see Lux Lab.

Requirements

Requirement	Minimum version
Visual Studio Code	1.97
Node.js	24
pnpm	10
now-sdk	4.12
Target instance	Zurich Patch 12 or later, or Australia Patch 5 or later

The *Lux*: **Show Environment Check** command displays the required and detected versions for Node.js, pnpm, and ServiceNow SDK. After fixing a failure, run *Lux*: **Re-check Environment**.

⚠ **Warning:** The Environment Check page offers **Skip checks and proceed anyway**. Skipping leaves the extension's features enabled while the checks are unmet, and commands may fail. The check must pass before you can create a project.

Instance connections

You connect to an instance from **Instance Explorer** in the Lux Lab activity bar view by selecting **Connect to ServiceNow**, or by running *Lux*: **Add ServiceNow Instance**. You enter the instance name or full URL, then select an authentication method.

Basic

You enter your username, password, and a unique profile alias. The extension validates the credentials against the instance, then stores the profile in VS Code SecretStorage. The extension masks passwords and doesn't write them to the project.

OAuth

You enter a profile alias, open the authorization URL the extension displays, authorize in the browser, and paste the returned code into a masked input. The

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

extension exchanges it using Authorization Code with PKCE and stores the token in VS Code SecretStorage. You don't enter a username or password in the extension.

Remote instances must use HTTPS.

What it provides

Instance Explorer

Browses Experiences, Pages, and Widgets on a connected instance. Select an artifact to open its metadata, open its source, or launch a preview.

Script Runner

Creates projects, installs dependencies, and runs dev, build, and deploy. Actions include **Dev**, **Build**, **Deploy**, **Stop**, **Show Command Output**, and **Show Dev Server Terminal**.

Quick Preview

Opens local preview-able pages and widgets without leaving VS Code. Save the file first, then select it in Quick Preview. You can also select and hold (or right-click) a preview-able file and select *Lux*: **Open Preview**.

An AI agent can drive page and widget creation through the project's bundled skills. The extension isn't tied to a specific agent vendor.

Commands

Open the **Command Palette** (`Cmd+Shift+P` on macOS, `Ctrl+Shift+P` on Windows and Linux) and search for *Lux*:.

- *Lux*: Show Welcome Page
- *Lux*: Show Environment Check
- *Lux*: Re-check Environment
- *Lux*: Add ServiceNow Instance
- *Lux*: Sign in to Active Instance
- *Lux*: Refresh Instance Explorer
- *Lux*: Create LUX Project
- *Lux*: Create Page
- *Lux*: Run Script...
- *Lux*: Build
- *Lux*: Deploy
- *Lux*: Open Preview
- *Lux*: Show Command Output

Selecting and holding (or right-clicking) a project, page, widget, or preview-able file in the **Explorer** or Lux Lab views gives the same actions in context. The context menu also opens a terminal and reveals a project in **Finder** or **File Explorer**.

Workflow

The Lux: Create Lux Project command scaffolds a standalone application. You select the instance, the parent folder, the application name, the project folder name, the **Standalone App** template, and the scope suffix. When the extension retrieves the vendor prefix from the

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

instance, the prefix is fixed and only the suffix is editable. The extension then scaffolds, installs dependencies, and opens the project. The Lux: `Create Page` command adds a route and creates `pages/<page-slug>/page.js`.

For extension applications rather than standalone ones, see Create an extension application with Lux Lab.

Related topics

- Lux Lab
- ServiceNow SDK

ServiceNow SDK

ServiceNow SDK is the command-line tool that scaffolds, builds, and deploys a Lux application. It's distributed as the npm package `@servicenow/sdk` and invoked as `now-sdk`.

ServiceNow SDK overview

Use the ServiceNow SDK CLI directly, rather than Lux Lab, when you need full control over your development environment. Lux Lab is a desktop application that bundles an editor, an AI coding agent, a local dev server, and a one-click deploy action into a single window. The CLI exposes those same operations as separate commands (now-sdk build, now-sdk install, etc.). You can run these commands in whatever editor, terminal, and CI/CD pipeline you choose.

Requirements

- Node.js 24 or later
- pnpm 10 or later — the project's package manager
- Git 2.30 or later
- Access to a ServiceNow instance (local or remote) with credentials

ServiceNow SDK Versions

Version	Notable changes
4.12.x (latest)	Lux fixes and enhancements, async app installs by default, expanded `auth` and `cicd` commands. The floor for Lux Lab for VS Code.
4.10.x	The `now-sdk cicd` command, for CI/CD pipeline integration
4.9.x	Maintenance and reliability fixes for the build and transform pipeline. The floor at which `@servicenow/sdk` became the customer-facing CLI for the Lux dev server.

Check which version a project uses:

```
cat package.json | grep @servicenow/sdk
```

Installation

```
npm install -g @servicenow/sdk@latest
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Commands

Now SDK commands

Command	What it does
`now-sdk auth`	Configure authentication to an instance
`now-sdk init` (alias: `create`)	Initialize a new application, apply a template to an existing one, or convert a legacy application
`now-sdk download <directory>`	Download application metadata from the instance
`now-sdk build [source]`	Compile sources into app files and generate an installable package
`now-sdk install` (alias: `deploy`)	Install or update the application on the instance
`now-sdk dependencies [sysIds..]`	Download configured dependencies and TypeScript type definitions
`now-sdk transform`	Convert XML records from the instance or a local path into Fluent source code
`now-sdk clean [source]`	Clean the output directory
`now-sdk pack [source]`	Zip the built app into an installable artifact
`now-sdk explain [topic]`	Display documentation for an SDK topic
`now-sdk query <table>`	Query records from a table on the instance
`now-sdk cicd`	Integrate builds and installs into a CI/CD pipeline
`now-sdk run dev`	Start the local dev server with hot reload

A scaffolded project wraps the common ones as `package.json` scripts — `pnpm dev`, `pnpm build`, `pnpm deploy`, `pnpm deploy:reinstall`, `pnpm lint`. So day to day you run those instead of calling `now-sdk` directly. For the scaffold those scripts come from, see Create your first Lux experience.

Those are the only scripts a scaffolded app ships with. The `build:force`, `storybook`, `test`, and `typecheck` scripts exist only inside the Lux monorepo and won't be in your app's `package.json`.

Reference documentation

The SDK's own documentation covers the commands, `now.config.json`, and the Fluent language in depth.

SDK documentation resources

Resource	URL
SDK documentation home	`https://servicenow.github.io/sdk/`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

SDK documentation resources (continued)

Resource	URL
CLI reference	`https://servicenow.github.io/sdk/cli`
`now.config.json` reference	`https://servicenow.github.io/sdk/config/now-config-reference`
npm package	`https://www.npmjs.com/package/@servicenow/sdk`
Release notes	`https://github.com/servicenow/sdk/releases`

Related topics

ServiceNow SDK

Create your first Lux experience

Install the ServiceNow SDK toolchain, scaffold a new Lux application, and deploy your first app to your ServiceNow instance. This foundational workflow guides you through environment setup, local development, and deployment.

About this task

If you'd rather use Lux Lab or the VS Code extension, which wrap these same steps in a UI, see Development tooling.

Before you begin

- Role required: admin
- Node.js 24 or later and `pnpm` 10 or later
- A ServiceNow instance you can deploy to
- `@servicenow/sdk` installed globally: `npm install -g @servicenow/sdk@latest`

Procedure

1. Check your prerequisites

   Verify that you have the right version of Node.js.

   ```
   node --version
   ```

   Verify that you have the right version of pnpm.

   ```
   pnpm --version
   ```

   Verify that you have the right version of the ServiceNow SDK.

   ```
   now-sdk --version
   ```

   If your version does not match the listed requirements, install or update your packages as necessary.

2. Connect the ServiceNow SDK to your instance

   Run the following command to authenticate the now-sdk with your ServiceNow instance:

   ```
   now-sdk auth --add https://your-instance.service-now.com --type
    basic --alias dev
   ```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Replace `your-instance` with your actual instance name. You will be prompted to enter your ServiceNow username and password.

For more information on instance setup or other authentication methods, see the ServiceNow SDK authentication documentation.

3. Scaffold the application

Run the following command to create a local directory for your first experience and initialize a new ServiceNow application project. This scaffolds the project for Lux development with a boilerplate structure including configuration files, build scripts, and server-side logic.

```
mkdir my-app && cd my-app
npx @servicenow/sdk@latest init --template javascript.aiux
```

You will see prompts for an app name, package name (also used as the URL basename), and scope name. To skip the prompts:

```
npx @servicenow/sdk@latest init --template javascript.aiux
 --appName my-app --scopeName x_aix_myapp --packageName my-app
```

The scope name must start with your vendor prefix and be 18 characters or fewer. Don't change it after the first install. Changing it without uninstalling the old scope from the instance creates a duplicate.

Note: For more information about application scope naming, refer to: application scope naming documentation.

Flag	Purpose
`--instance=<url>`	ServiceNow instance URL
`--scope=<scope>`	App scope for scoped applications
`--template=<name>`	Starter template (default: default)
`--env`	Create .env file interactively

The scaffold generates the following project structure.

```
my-app/
├── aiux.json              # Lux app manifest — app name, URL
 basename, landing route
├── now.config.json        # ServiceNow scope manifest —
 scope, scopeId, install settings
├── package.json           # Depends on the Lux runtime +
 @servicenow/sdk
├── application.js         # App-level lifecycle hooks
 (setup, navigate, teardown)
├── theme.js               # App theme tokens (colors, radii)
├── tailwind.app.css       # Tailwind configuration
├── head.app.css           # App-level head styles
├── eslint.config.mjs      # ESLint flat config
├── pages/                 # File-based routing — one
 directory per route
│   ├── home/
│   │   ├── page.js         # The page (default export)
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
│   │   └── components/    # Components co-located with the
page that uses them
│   └── incidents/
│       └── page.js        # A page with a static async
loader(ctx)
├── widgets/               # Custom widgets (+ server
scripts)
│   └── aiux-hello-world/
│       ├── index.js
│       └── server-script.js
├── utils/                 # Shared utility modules
├── scripts/               # Postinstall helpers
├── dist/                  # Build output (generated — do not
commit)
└── dist-metadata/         # Build metadata (generated — do
not commit)
```

Add `dist/` and `dist-metadata/` to your `.gitignore`. The build generates both.

4. Install dependencies

   Run the following command to install dependencies.

   ```
   pnpm install
   ```

5. Run the example locally

   Use the following command to start at local dev server.

   ```
   pnpm run dev
   ```

   This spins up the full request pipeline on your machine with hot reload, so you can build against a real instance locally.

   To see your experience, Open `http://localhost/aiux/<basename>/<route>` in your browser.

   Example: if your app's `basename` is `my-app` and you want to access the `/home` route: `http://localhost/aiux/my-app/home`

   Note: The path after `/aiux/` is your app's basename (from `aiux.json`) followed by the page route:

   - `/aiux/my-app/home` = basename `my-app` + route `/home`
   - `/aiux/my-app/dashboard` = basename `my-app` + route `/dashboard`

   Editing a file under pages/ or components/ automatically recompiles and reloads the page in your browser. Build artifacts are written to .dev directories during development.

   You can try creating a page to add more to your experience: Create a page.

6. Build

   ```
   pnpm run build
   ```

   This compiles your pages and widgets into an installable package, written to `dist/`. It also confirms that the toolchain and dependencies are in place.

   A clean build means that Node, pnpm, and the @servicenow/karuna package all resolved correctly.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

7. Deploy

```
pnpm run deploy
```

This installs the application on your ServiceNow instance. Open `https://<your-instance>/aiux/<basename>` to see it live.

Result

The starter app ships two routes (Home and Incidents), an example component, and an example widget.

What to do next

- Lux architecture for how these pieces fit together
- Create a page to add more routes
- Lux Widgets to build the widget the scaffold generated

Lux architecture

Lux renders UI as a separate service in front of a ServiceNow instance. It builds pages, sends HTML to the browser, and calls back for data.

This architectural split shapes everything else about how a Lux app behaves.

Request pipeline

A request to a Lux app passes through four layers before a response reaches the browser. Each layer has one job: routing, version selection, rendering, and data fetching.

HTTP proxy, port 80

Entry point for every request, and a plain Node.js HTTP server. It matches the incoming URL against the `/aiux` gateway basename, strips the prefix, injects authentication and forwarding headers, and then forwards the request to the version router. Anything that doesn't start with `/aiux` is proxied straight through to the ServiceNow instance, so Lux intercepts only its own traffic.

Version router, port 3000

Decides which running version of the rendering service handles the request. Requests to `/aiux/*` go to the current version (vN, port 3001), and requests to `/aiux-1/*` go to the previous version (vN-1, port 3002). If a version is unavailable, the router falls back to vN. It adds an `x-aiux-version` response header, so you can tell which version answered.

Rendering service, ports 3001 and 3002

One Fastify service per live version. For a page request it creates a request context and resolves the route to a page widget. It then runs every loader in the tree (the page's, its layout's, and any widgets it contains) in parallel rather than one after another. Finally it server-renders the Lit widgets to an HTML string and streams that string back.

ServiceNow instance, port 8080 or a remote host

Target of the data calls a page makes. Loaders call back into the instance over REST, GraphQL, and Table API endpoints to fetch the data a page needs. Those calls are authenticated as the logged-in user. The rendering service forwards that user's session cookie and cross-site request forgery (CSRF) token rather than using a separate service credential.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note:

Basename matching order. The gateway basenames are a fixed, ordered list: `/aiux-1`, `/aiux-2`, `/aiux`. The first match wins. Bare `/aiux` has to stay last, or it shadows the versioned prefixes. Three basenames are matched; two version tiers run.

Running loaders concurrently instead of sequentially is a deliberate performance choice. A page pulling from several different ServiceNow APIs fires all of those calls at once instead of waiting on each in turn. The results are serialized into the HTML, so the browser doesn't re-fetch them on first paint.

Loaders don't get the raw request to do this, though. They receive a filtered, loader-safe context exposing only what they need.

How Lux renders pages

Lux builds pages on the server first, then adds browser behavior second.

When a user opens a Lux app, the framework does the following:

1. It runs your data-fetching code on the server.
2. It builds the UI as an HTML string.
3. It streams the HTML to the browser.

The user sees the full content immediately. No blank screen or loading spinner shows while JavaScript downloads.

After the HTML arrives, the browser downloads the JavaScript bundles. Then the JavaScript makes the page interactive. It adds event listeners and reactive behavior to the page elements that already exist. It does not build these elements again.

After this step, the page works like a single-page app. The browser does later navigation on the client. It does not reload the full page.

This method gives these measurable benefits:

- Content appears fast. The browser does not wait for JavaScript to show content.
- The first HTML contains real content, not an empty shell. This helps search engines and assistive technology.
- Data fetching happens on the server, close to the ServiceNow instance. The browser does not make extra round trips.

An experience can declare resource hints in `aiux.json`, under `experienceProperties.resourceHints`. The hints apply to every page of the experience. Lux resolves the hints before it builds the head HTML of the response.

The `render()` output of each widget must be the same on the server and on the client. For this reason, the contract is stricter than the contract of a client-only UI framework. The `render()` function must be pure and deterministic. It must not use browser globals, such as `window` or `document`, unless a guard checks for the server environment first.

For more about this contract, and for where side effects are allowed, refer to Core concepts

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Built on Lit widgets

All UI in Lux is built with Lit, which provides web components with Shadow DOM. You don't extend Lit's base class directly. You extend a Lux base class that layers SSR support, styling, and data fetching on top of Lit.

> **⚠ Warning:**
>
> Two copies of Lit break hydration. Lit is externalized into a single `lit-core` bundle loaded through an import map. If your app bundles its own copy, hydration and `instanceof` checks break. In a non-core component package, list `lit` as a `peerDependency`, not a `dependency`.

Lux building blocks

A handful of concepts recur throughout a Lux app. The following table names each one.

Lux building blocks

Concept	What it is
App manifest (`aiux.json`)	Declares an app's name, its URL basename, and its landing route.
Pages	Lit widgets under a `pages/` directory. The directory structure maps directly to URLs, so routing follows file layout rather than a separate route table.
Loaders	`static async loader(ctx)` methods that fetch data on the server before a page or widget renders. A loader's return value is available on the widget as `this.loaderData`.
Widgets	Reusable UI units, either self-fetching and independently discoverable or purely presentational, depending on which base class they extend.
Layout	`pages/layout.js`, the single file that renders the shared app chrome (header, navigation, footer) around every page.
Context API	Shares data between widgets without passing it down through props at every level.

Each of these gets its own treatment in Core concepts.

Subsystem map

Beyond the request pipeline, the framework groups into a handful of subsystems. You rarely touch them directly, but their vocabulary shows up in error messages. The following table describes what each subsystem covers.

Framework subsystems

Subsystem	Responsibility
Rendering and SSR	`@lit-labs/ssr` rendering, hydration, streaming, and the isolate pool that runs untrusted widget code.
Routing and navigation	File-based route resolution, the client-side router, prefetch and hover-prefetch, navigation guards.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Framework subsystems (continued)

Subsystem	Responsibility
Data and state	Loaders, the Context API, batch fetch, caching, and dirty-state tracking.
Local development	The dev proxy, the widget dev server, hot reload, and the command-line interface (CLI).
Build and bundling	Rollup and SWC pipelines that emit separate SSR and browser builds, with Lit externalized into a single `lit-core` bundle.
Observability	Structured backend logging, browser log forwarding, tracing, and server timing.
Advanced UI	Modal teleporting, side panels, the window manager, page alerts, and layout overrides.

Where a Lux app runs

The pipeline you develop against locally is the same pipeline that runs in production: proxy, version router, rendering service, and calls back to a ServiceNow instance. That is what makes local behavior a reliable predictor of production behavior.

In production, that pipeline is a cloud deployment sitting in front of many ServiceNow instances rather than something deployed per instance. When someone navigates to `/aiux/*` on their ServiceNow instance, the request is proxied to the Lux deployment. The deployment identifies the tenant from the hostname, runs loaders against that tenant's instance APIs with the user's session, and streams the HTML back.

This shared deployment model has a consequence worth knowing at the architecture level. Because one deployment serves many tenants at once, it has to stay compatible with every ServiceNow version still in use, not just the newest one.

For how the files behind these concepts are laid out in a project (`aiux.json`, `pages/`, the layout, widgets), see Project structure.

Core concepts

Lux apps use six core building blocks to create consistent, maintainable applications: a manifest, pages, loaders, widgets, a layout, and a context API for sharing data.

This topic introduces each piece and how they relate. For the request lifecycle that ties them together (proxy, routing, server rendering, hydration), see Lux architecture. For step-by-step instructions on building with each concept, see the topics on creating pages, widgets, and components.

Core building blocks

App manifest, `aiux.json`

Declares your app's name, its URL basename, and its landing route. The framework reads it to serve your app under `/aiux/<basename>/<route>` and to determine which page to show first.

Page

A Lit widget that lives under your app's `pages/` directory. The directory structure maps directly to URLs through file-based routing, so `pages/inventory/page.js` serves `/inventory` and `pages/inventory/[id]/page.js` serves `/inventory/:id`. A page is the default export of its `page.js`. Pages

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

extend *AIUXElement*, the same base class reusable components extend, so a page is really just a component that the router can address directly.

Loader

A *static async loader(ctx)* method on a page or widget. It runs on the server before the component renders and fetches whatever data the component needs. Its return value becomes available on the component as *this.loaderData*. Because loaders run on the server, they fetch data close to your ServiceNow instance rather than over a client round trip.

Widget

A reusable Lit UI unit. Widgets extend one of two base classes from `@servicenow/aiux-components-core`. Use *AIUXWidgetElement* for a discoverable widget that owns its own data, or *AIUXElement* for a presentational widget that takes its data through properties. Neither extends *LitElement* directly. The Lux base classes add Tailwind and DaisyUI styling, *this.loaderData*, and *getLayoutData(key)* for reading layout-level data.

Layout

A single file, `pages/layout.js`, that renders the app chrome (header, navigation, footer) around every page. It extends *NavLayout*, itself an *AIUXAppLayoutElement*, and is not itself a page. It's the persistent shell every page renders inside, and it is not re-created when the user navigates from one route to another. The layout is also where an app declares its navigation tree (`l1Nav`, with nested `l2Nav` and `l3Nav`). That tree determines both the sidebar and the tab structure pages appear under.

Context API

A mechanism for sharing data between components without prop drilling. Use *createContext()* to establish a context that parent components can provide and child components can consume, enabling state management across the component tree.

Coordination mechanisms

Beyond this core spine, more mechanisms come up as an app grows. Widgets may need to coordinate with each other, to coordinate with the platform's AI chat, or to sit in a layout an admin can rearrange.

Widget-to-chat protocol

How a page participates in a two-way exchange with the ServiceNow Otto chat backend. Widgets push their current state to the AI through `setAiContext()`, and the AI can invoke actions on a widget in response to what the user asks it.

Region DSL

The `region` tagged template, imported from `@servicenow/aiux-components-core`. It's a constrained, HTML-like domain-specific language (DSL) for arranging widgets inside a page, deliberately restricted to a known set of layout primitives (`row`, `column`, `raw`) and widget tags. Arbitrary HTML, `class` and `style` attributes, event bindings, directives, and text children are all rejected at parse time. The restriction is what lets a visual editor parse, manipulate, and serialize a layout programmatically. A stored region has to be expressible as a value in a database column and a field in a property panel.

The admin-facing region editor lets a configuration admin restructure a rendered page's regions: add and remove rows, columns, and widgets, resize column spans, move containers. The editor saves the result as an override row in `sys_aix_page_region`. A page whose regions have never been overridden

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

has no row at all. The renderer records the code-authored structure so the editor has something to open.

Where a region lives and what it renders

A region is not written inline in a page's *render()*. It lives in its own file under `<projectRoot>/regions/`, and that file's default export is a thunk that takes no arguments and returns the tagged template. The region's name comes from its file path. The page imports the thunk and calls it once per render. That call is what lets the runtime read the current request's stored overrides instead of a snapshot taken when the module loaded. Importing `region` anywhere outside `regions/` fails the build, and so does a function that takes arguments.

Every rendered region is enclosed in exactly one wrapper element, `<div data-aiux-region="{name}" class="flex flex-col gap-layout overflow-hidden">`. That wrapper is part of the rendered-output contract, so anything that needs to find a region in the DOM queries `data-aiux-region` as a literal. The renderer's constants aren't exported. The wrapper is a real layout box. It stacks the region's top-level rows as a flex column on `gap-layout`, the same theme-able gutter a `row` puts between its columns. So two top-level rows sit exactly one gutter apart. The consequence for the page is that a region is one item of the page's own grid or flex parent, rather than each of its rows being one.

A region has one spacing value and no padding. The `gap` and `padding` attributes are deprecated. They are still parsed, so they never reach the rendered element, and they are ignored on any tag with any value. Retune the gutter with the `--layout-gap` custom property at `:root`, or `--spacing-layout` on an ancestor to scope one subtree.

Important:

A widget is a leaf. A widget in a region holds no children. An element inside one throws at parse time, and text children are rejected outright. `<slot>` is a blocked tag, because a slot projects the host element's light DOM and a region doesn't own that. `<raw>` is not the workaround: it emits whatever markup it is given, widget tags included, but its content is unsanitized trusted-input-only markup. A widget nested that way is invisible to both the editor and the structural rules. Composition hosts (`aiux-layout-host`) were removed outright. To compose one widget inside another, write plain Lit `html` with Tailwind classes. For more information about widgets, see Lux Widgets.

How it all renders

All of this UI is built with Lit and styled with Tailwind and DaisyUI, applied automatically through adopted stylesheets, so you don't import styles manually. Every page and widget renders on both the server, for first paint, and the client, during hydration and on later re-renders. So `render()` must be pure: same input, same output, every time. For what drives that server and client split, see Lux architecture.

Project structure

A scaffolded Lux app has a fixed file layout: two manifests, an app-level lifecycle module, a theme file, and directories for pages, widgets, and regions.

```
my-app/
├── aiux.json              # Lux app manifest
├── now.config.json        # ServiceNow scope manifest
 (now-sdk)
├── package.json           # Depends on @servicenow/karuna +
 @servicenow/sdk
├── application.js         # App-level lifecycle hooks (setup,
 navigate, teardown)
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
├── theme.js                # AppTheme tokens (colors, radii)
├── pages/                  # File-based routing (one directory
per route)
│   ├── layout.js           # App chrome (header, nav) wrapped
around every page
│   ├── home/
│   │   ├── page.js         # AIUXElement page (default export)
│   │   └── components/     # Components co-located with the
page that uses them
│   │       └── aiux-feature-card.js
│   └── incidents/
│       └── page.js         # Page with `static async
loader(ctx)`
├── widgets/                # Custom AIUXWidgetElement widgets
(+ server scripts)
│   └── aiux-hello-world/
│       ├── index.js
│       └── server-script.js
├── regions/                # Region DSL files (one thunk per
named region)
├── eslint.config.mjs       # ESLint flat config (aiux plugins)
├── scripts/                # setup-agent-skills.mjs
(postinstall helper)
├── AGENTS.md               # Canonical guidance for AI coding
agents
├── CLAUDE.md / GEMINI.md   # Point back to AGENTS.md
├── .claude/                # Hooks + settings for Claude Code
└── .nvmrc / .gitignore
```

What each entry is for

Entry	Purpose
`aiux.json`	Lux app manifest (the experience's declaration).
`now.config.json`	ServiceNow scope manifest that `now-sdk` reads on build and deploy. Your app has two manifests, not one.
`package.json`	Dependency and script declarations. `@servicenow/karuna` and `lit` are dependencies; `@servicenow/sdk`, the `eslint-plugin-aiux-*` rules, and the agent packs are dev dependencies. Holds the `dev`, `build`, `serve`, `deploy`, and `lint` scripts.
`application.js`	App-level lifecycle hooks: `setup()`, `navigate()`, and `teardown()`.
`pages/layout.js`	The layout every page in the experience renders inside (header, navigation, footer). It extends `NavLayout` and is not re-created when the user navigates between routes.
`theme.js`	The experience's theme tokens: colors, radii.
`pages/`	File-based routing. Each subdirectory is a route, and `page.js` in that directory is an `AIUXElement` page (the file's default export, optionally with `static async loader(ctx)` for server-side data fetching).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Entry	Purpose
`pages/<name>/components/`	Components scoped to the page that uses them, co-located rather than shared globally.
`widgets/`	Custom `AIUXWidgetElement` widgets, each with an `index.js` and an optional `server-script.js`.
`regions/`	Region DSL files. Each one default-exports a nullary thunk returning a `region` template, and its region name comes from its path. `region` can't be imported from anywhere else.
`eslint.config.mjs`	ESLint flat config with the `eslint-plugin-aiux-*` rules.
`scripts/`	Build and setup helpers, such as `setup-agent-skills.mjs`, run through `postinstall`.
AGENTS.md, CLAUDE.md, GEMINI.md	AI coding agent guidance. CLAUDE.md and GEMINI.md point back to AGENTS.md.
`.claude/`	Hooks and settings for Claude Code.
`.nvmrc`, `.gitignore`	Standard Node project files.

Note the split between the two files at the app root that are easy to confuse. `pages/layout.js` is the layout (the persistent chrome your pages render inside, and a widget class in its own right). `application.js` is the app's lifecycle module: three hooks, no markup. They are separate files with separate jobs.

Fields in aiux.json

Field	What it defines
`name`	The human-readable application name.
`basename`	The URL prefix (everything under `/aiux/<basename>` is this experience).
`landing`	The route served when someone opens the app with no path.
`development.token`	A basic-auth token used only for local development.
`build.exclude`	Glob patterns kept out of production builds.
`experienceAcls`	Who is allowed into the experience at all.
`scope` / `scopeId`	The ServiceNow application scope that owns the records. Added at deploy time, not by hand.

What the tree compiles to

Deploying compiles each source file into its corresponding platform record.

Source	What it becomes
`aiux.json`	The experience itself (name, URL prefix, landing page, per-experience settings, and any menus declared in it)
`pages/<route>/page.js`	One page record per page
`pages/layout.js`	The app's layout

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Source	What it becomes
`widgets/<name>/index.js`	Reusable, discoverable widgets
`theme.js`	Theme tokens
Static assets	Attachments in your scope (images, fonts, JSON)
Compiled bundles	JavaScript attachments on the widget records, served by the runtime

Pages, widgets, layouts, and internal helper components all compile to the same kind of widget record. A `category` field is what distinguishes them: `page`, `layout`, `discoverable`, or `internal`. So there's one schema to learn, not three.

Related topics

Core concepts

Build with Lux

Build with Lux

You build a Lux app in code, starting from its pages. A page is a Lit web component that Lux registers as a route based on where you put its file.

Use page collections to let more than one page module claim the same route. Topics in this chapter also cover layouts and editable regions, making a component property configurable inline, and authoring widgets. They close with components: their fundamentals, their lifecycle, and how to make one reusable across pages and apps.

Create a page

Add a page to a Lux app. A page is a Lit web component that Lux registers as a route based on where you put its file.

Before you begin

- Role required: admin
- A scaffolded Lux app (see the getting-started topic for your app; out of scope here). You need an existing `pages/` directory and a working `pnpm build`/dev server setup.
- Familiarity with Lit (`html`, `customElement`, reactive properties).

About this task

Create a working page that renders at a URL in your app, optionally fetching data on the server before it renders.

Procedure

1. Create a directory under `pages/`

   Lux uses file-based routing: the directory path under `pages/` becomes the URL path. Create one directory per route segment.

   ```
   mkdir -p pages/risk
   ```

   A nested path becomes a nested route:

   ```
   mkdir -p pages/settings/preferences
   ```

   `pages/settings/preferences/page.js` maps to `/settings/preferences`.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

2. Add the page file.

   Inside the directory, create `page.js` (or `page.ts`). The build looks for a file named `page.js`/`page.ts` to terminate a route; anything else in that directory is ignored.

   Older applications may name the page file `index.js` instead. Both work: `page.js` is preferred for new code, and an existing app can migrate gradually.

```
// pages/risk/page.js
import {html} from 'lit';
import {customElement} from 'lit/decorators.js';
import {AIUXElement} from '@servicenow/aiux-components-core';

@customElement('my-app-risk-page')
export default class RiskPage extends AIUXElement {
  static async loader(ctx) {
    return {title: 'Risk'};
  }

  render() {
    return html`<h1>${this.loaderData?.title}</h1>`;
  }
}
```

   A page file must meet the following requirements:

   - Extend `AIUXElement`, not `LitElement` directly: it supplies Tailwind/DaisyUI styling, the `loaderData` accessor, and SSR safety.
   - `@customElement(...)` registers the tag. Prefix it with your app name so it doesn't collide with another app's tags (`my-app-risk-page`, not `risk-page`).
   - `export default`: the build's page discovery looks for the default export. Any other component you define in the same directory should use a named export instead.
   - `render()` must be pure and deterministic: it runs during server-side rendering and in the browser, so it can't touch `window`, `document`, or `localStorage`.

3. **Optional:** Fetch data with a loader.

   (Optional) `static async loader(ctx)` runs on the server before the page renders. Its return value is JSON-serialized into the page and becomes available as `this.loaderData`.

```
import {AIUXElement, getHeaders} from
 '@servicenow/aiux-components-core';

@customElement('my-app-risk-page')
export default class RiskPage extends AIUXElement {
  static async loader(ctx) {
    const res = await fetch(

 `${ctx.protocol}://
${ctx.hostname}/api/now/table/sn_risk_risk`,
      {headers: getHeaders(ctx)}
    );
    return {items: (await res.json()).result ?? []};
  }

  render() {
    const {items} = this.loaderData;
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    return html`
      <section class="p-4 flex flex-col gap-2">
        ${items.map(item => html`<div
 class="aiux-card">${item.number}</div>`)}
      </section>
    `;
  }
}
```

Use `getHeaders(ctx)` from `@servicenow/aiux-components-core` when calling ServiceNow table APIs: it forwards the session cookie and CSRF token in one call. `loader` runs on the server, so no browser globals here either.

Note: Route parameters and query parameters on `ctx` are strings: cast them yourself if you need a number or boolean.

4. **Optional:** Add a dynamic route segment.

   (Optional) Wrap a directory or file name in square brackets to declare a path parameter:

   ```
   mkdir -p "pages/risk/[riskId]"
   ```

   `pages/risk/[riskId]/page.js` maps to the route pattern `/risk/:riskId`. The value lands on `ctx.params` in the loader:

   ```
   static async loader(ctx) {
     const {riskId} = ctx.params; // a string
     // fetch the record identified by riskId ...
     return {riskId};
   }
   ```

   You can nest dynamic segments, for example `pages/risk/[riskId]/controls/[controlId]/page.js` for `/risk/:riskId/controls/:controlId`.

5. Build and check the route.

   ```
   pnpm build
   ```

   The build walks `pages/` and writes a route manifest mapping each route pattern to its bundle and tag name. Your app's basename (from `aiux.json`) is prepended to every route to form the production URL: `/aiux/<basename><route>`. For a basename of `my-app`, `pages/risk/page.js` serves at `/aiux/my-app/risk`.

6. Link to the page.

   A page isn't reachable until something links to it. Either use a normal anchor tag:

   ```
   render() {
     return html`<a href="/risk">Go to Risk</a>`;
   }
   ```

   or navigate programmatically from an event handler:

   ```
   import {navigate} from '@servicenow/aiux-client';

   _openRecord(e) {
     const {riskId} = e.currentTarget.dataset;
   ```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
  navigate(`/risk/${riskId}`);
}
```

The router intercepts a same-origin `<a href>` select automatically and swaps in the new page client-side, without a full reload. It intercepts only a plain left click, with no modifier keys, no `target="_blank"`.

Example: Verify

- `pnpm build` succeeds.
- Open the route in a browser (or check the Network tab's response preview during SSR) and confirm your rendered content and loader data appear.
- If the page uses `navigate()` or a sidebar link, select it and confirm client-side navigation works without a full page reload.

What to do next

- #unique_350: serving different page implementations from the same route
- #unique_1217: wiring a page into the app's shell and navigation
- #unique_1218: the full loader contract and `ctx` shape

Lux Widgets

A widget in Lux is a custom UI unit authored for a specific application. The platform ships no widgets; every widget on an instance was authored for an app.

This topic covers what a widget is, how it gets registered, and how the runtime resolves one by tag name. To learn how to build a Lux widget, see #unique_1219.

Everything you render in a Lux app is a Lit widget. A Lux widget is a Lit custom element, registered as a `sys_aix_widget` record, extending one of two platform base classes. Both give it Tailwind/DaisyUI styling, and server-side rendering (SSR) safety; the difference is discoverability and how it gets its data.

Project shape

```
my-widget/
                ├── aiux.json              # { "basename":
 "my-widget", "scope": "x_demo_widget", "scopeId": "…" }
                ├── now.config.json
                ├── package.json
                └── widgets/
                ├── _shared/           # NOT a widget (no
 index.js): discovery skips it
                │   └── chart.js
                ├── weather/
                │   └── index.js       # <x-demo-weather>
                └── air-quality/
                └── index.js       # <x-demo-air-quality>
```

Discovery convention: each sub directory of `widgets/` that contains an `index.{js,ts}` is one widget. Sub directories without an index entry (helpers, fixtures, shared modules like `_shared/`) are ignored silently, with no warning. Widgets and pages coexist: a project can contain only `widgets/`, only `pages/`, or both, and the build discovers whatever is present.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Choosing a base class

`AIUXWidgetElement` extends `AIUXElement`: it's a superset, so the real question is whether the widget needs what the superset adds. The choice affects runtime capabilities, not how the file builds: both go through the same discovery, analysis, and bundling pipeline.

Widget base classes

Base class	Use it for	Adds
`AIUXElement`	A presentational widget that receives its data through `@property` from a parent, or a page; every `pages/<route>/page.js` extends `AIUXElement`	Tailwind + DaisyUI styling, `this.loaderData`, `getLayoutData(key)`, the `static async loader(ctx)` convention, `this.server`, and `trackEvent()` telemetry
`AIUXWidgetElement`	A discoverable, self-contained widget that owns its data; the platform can register it and the AI can place it	Everything `AIUXElement` has, plus a reactive `this.data` populated by `this.server`'s result, dependency injection through `this.deps`, and chat context through `setAiContext()`

Note: `this.server` and `trackEvent()` live on the shared `AIUXElement` base and are available either way. `AIUXWidgetElement` is what makes `this.server`'s result land in a reactive `this.data`. Reach for `AIUXWidgetElement` when you need that reactive property, not to get `this.server` itself.

Start with `AIUXElement`. Reach for `AIUXWidgetElement` only when the widget genuinely needs to be a discoverable, self-fetching unit; because it's a superset, you can upgrade later without rewriting render logic. Pages are always `AIUXElement`. Never extend `LitElement` directly; that loses the automatic styling, `loaderData/data`, and the loader and server-script conventions.

Widget Builder

Widget Builder is where you create, edit, and preview Lux widgets without leaving the browser.

Widgets catalog

The Widgets catalog lists every widget on the instance, except internal widgets.

- Use the search box to filter widgets by name.
- Select **Create widget** to open a new widget in the Editor.
- Select a widget's action menu for **Edit**, **Clone**, or **Delete**.

Use the scope picker in the header to choose which application scope a widget you create lands in.

A widget included with the base system, or one from a store app rather than created by you, shows a locked state. Trying to edit or delete it is blocked. For details, see Customize a base system widget.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Editor

Selecting or creating a widget opens the Editor, a three-panel workspace, at `/aiux/builder/edit/widget/-1` for a new widget, or `/aiux/builder/edit/widget/{widget_sys_id}` for an existing one.

Chat panel

Describe what to build or change in natural language. Collapsible and resizable.

Widget editor

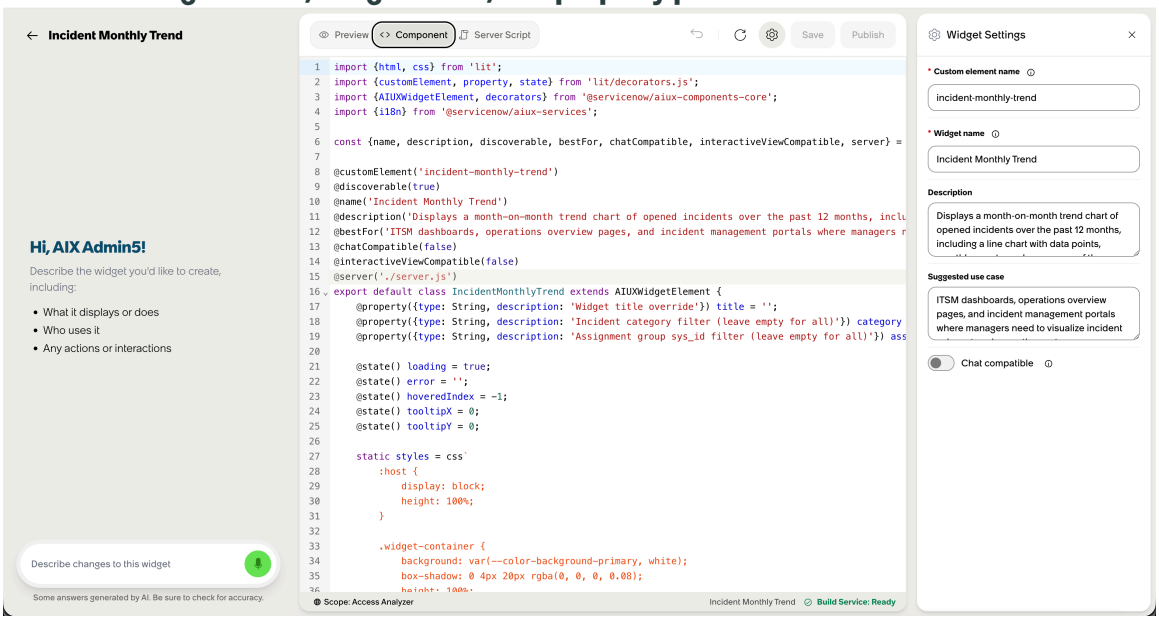
Tabs for **Preview**, **Component**, and **Script**. This is where you read and write the widget's code and see it rendered live.

Property panel

Widget Properties and **Preview Props** modes. Collapsible and resizable.

See Create a widget and Widget editor for what to do in each panel.

Editor showing the chat, widget editor, and property panels



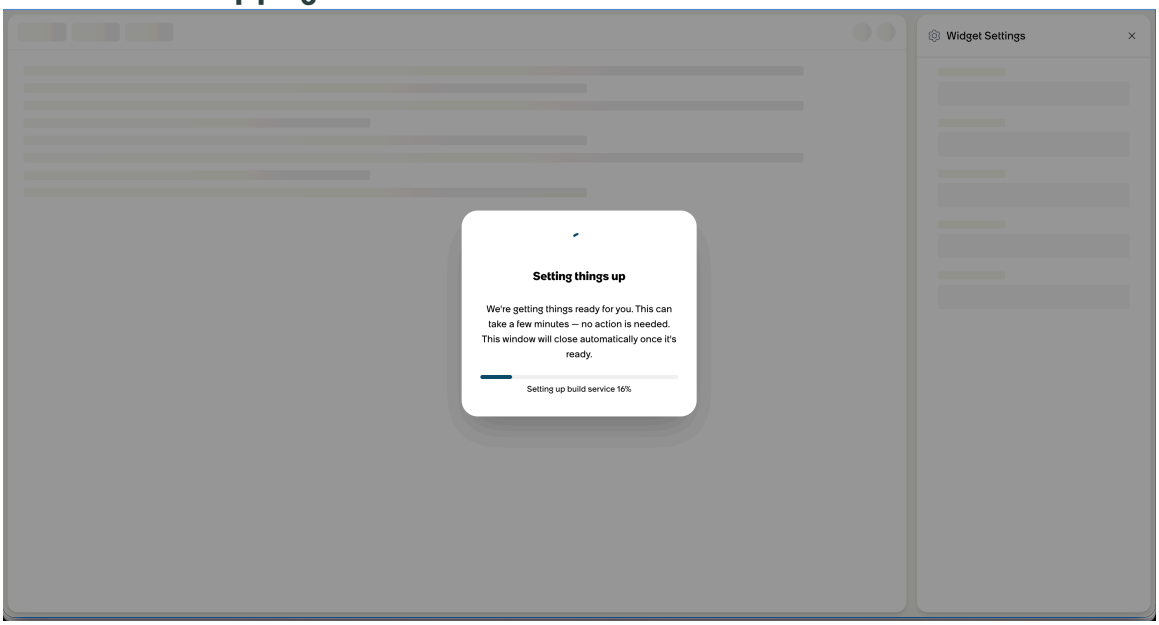
Build Service

Selecting Save or Publish hands the widget to the Build Service, an in-browser build tool powered by `now-sdk`. The Build Service compiles the widget to publish and deploy it to your instance.

Note: The first time you open the Editor, Widget Builder sets up the Build Service. A modal shows progress while this completes, it can take a few minutes.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Build Service setup progress modal



Create a widget

You can create a widget three ways — describe it in chat, build it by hand, or clone an existing widget. Each way produces a new widget ready to edit in the Editor.

Before you begin

Role required: admin

Procedure

- Create with AI chat

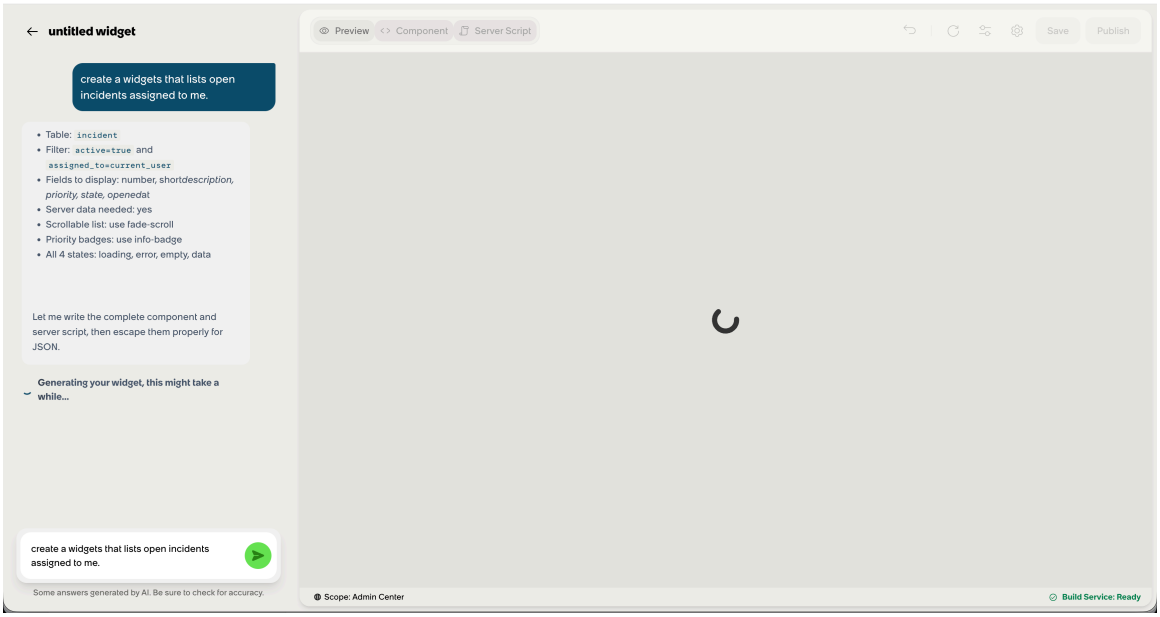
1. From the Widgets catalog, select **Create widget**.
   The Editor opens with a new, unsaved widget.
2. In the chat panel, type a description of the widget you want, for example, "a card that lists open incidents assigned to me" and send it.
3. Chat shows a thinking indicator while the widget is generated, followed by streamed progress text.
4. When generation finishes, Widget Builder builds and deploys the widget automatically. The URL updates to the new widget's ID, and the **Preview** tab shows the result.
5. Chat then prompts for further changes, describe a change, or move on to editing manually.

Every AI response in chat carries the disclaimer "Some answers generated by AI. Be sure to check for accuracy." Review the generated code and behavior before publishing.

Chat history is kept for the current browser tab (it survives a reload but doesn't follow you to another tab or device).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Chat panel generating a widget, with streamed thinking text



- Accept or decline a chat-driven change

  After chat applies a change, it shows **Accept changes** and **Decline changes**.

  - **Accept changes** keeps the update.
  - **Decline changes** reverts the widget to how it was before the change and reloads the editor with the prior code.

- Create manually

  1. From the Widgets catalog, select **Create widget**.
  2. Skip the chat panel and edit the **Component** and **Script** tabs directly in the widget editor.
  3. Select **Save** to build the widget without deploying it — the preview updates immediately from the build output.
  4. When you're ready to make it available on the instance, select **Publish** to build and deploy it.

- Clone an existing widget

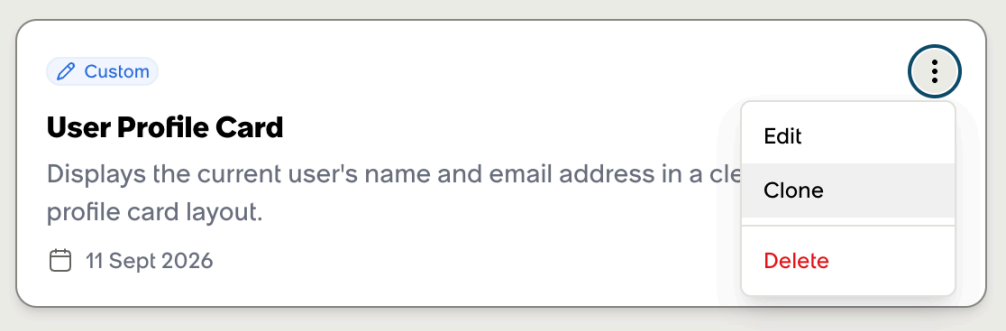
  Cloning gives you your own independent copy of a widget, seeded with its current source.

  1. From the Widgets catalog, select a widget's action menu and choose **Clone** (or clone from the create dialog when creating a widget).
  2. Review and edit the pre-filled name, ID, and description.
  3. **Confirm**.
     Widget Builder opens the new widget in the Editor, already published and unrelated to the original. Editing or deleting the clone never affects the widget you cloned it from.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The clone is created in whichever application scope is currently active.

Create/clone dialog with pre-filled name, ID, and description



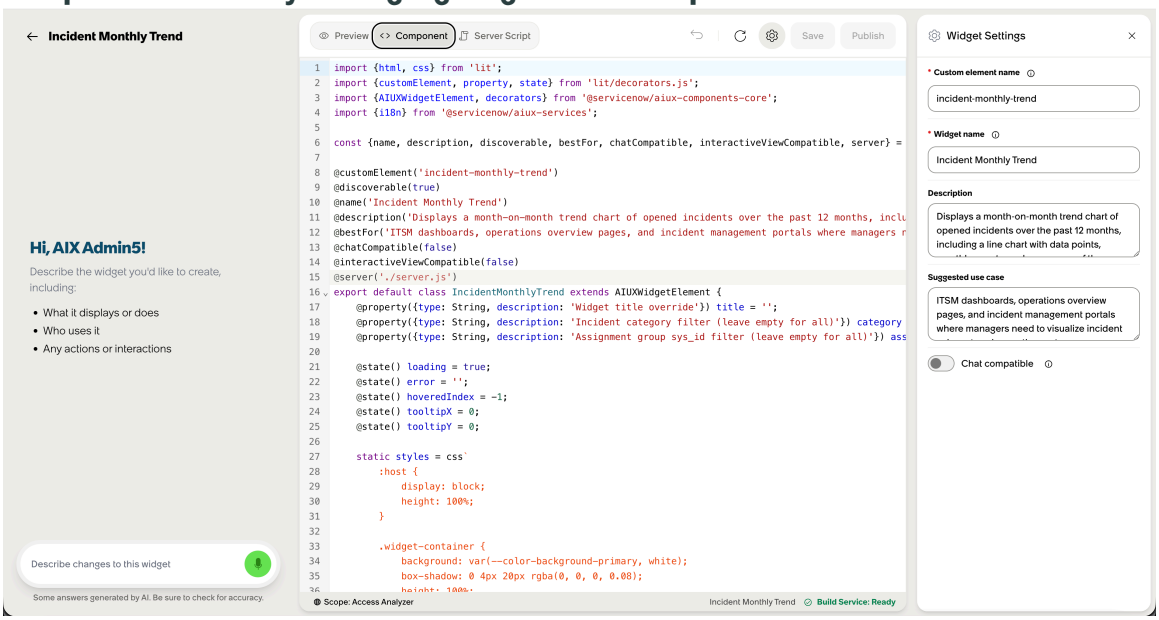
Widget editor

The widget editor is a three-panel workspace for building and previewing a widget. It has a code editor for the component and server script, a live preview, and a panel for widget properties.

Component and Script tabs

Both tabs are code editors with syntax highlighting and autocomplete for the widget component code and its server script. The server script is written as `export default function server(data, options, input) {...}`, with `GlideRecordSecure`, `GlideQuery`, and `gs` available — use `GlideRecordSecure` rather than `GlideRecord` so queries respect ACLs. Call `this.server.get(input)`, `.update()`, or `.refresh()` from the component to run it.

Component tab with syntax highlighting and autocomplete



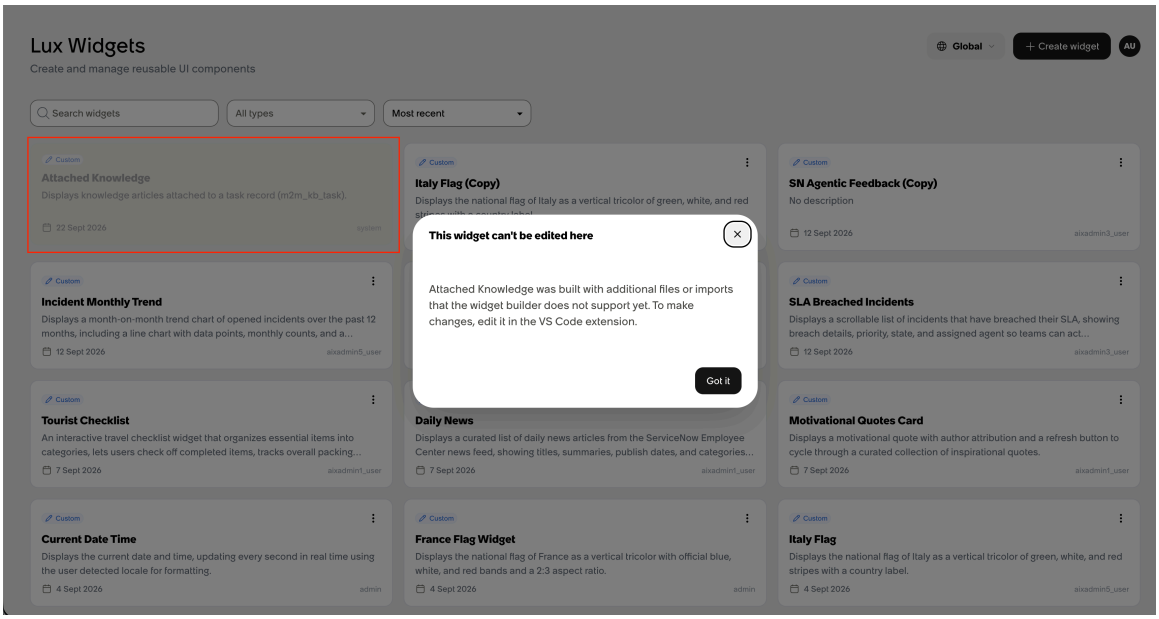
Complex widgets with additional files

Widget Builder only edits the widget's component (`index.js`) and server script (`server.js`). A complex widget that has relative imports to other utility files isn't fully editable here, those additional files aren't shown in the **Component** or **Script** tabs.

Important: If a widget imports from files other than its own component and server script, use the Lux Lab VS Code extension instead.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Complex widget with additional utility files, opened in the Lux Lab VS Code extension



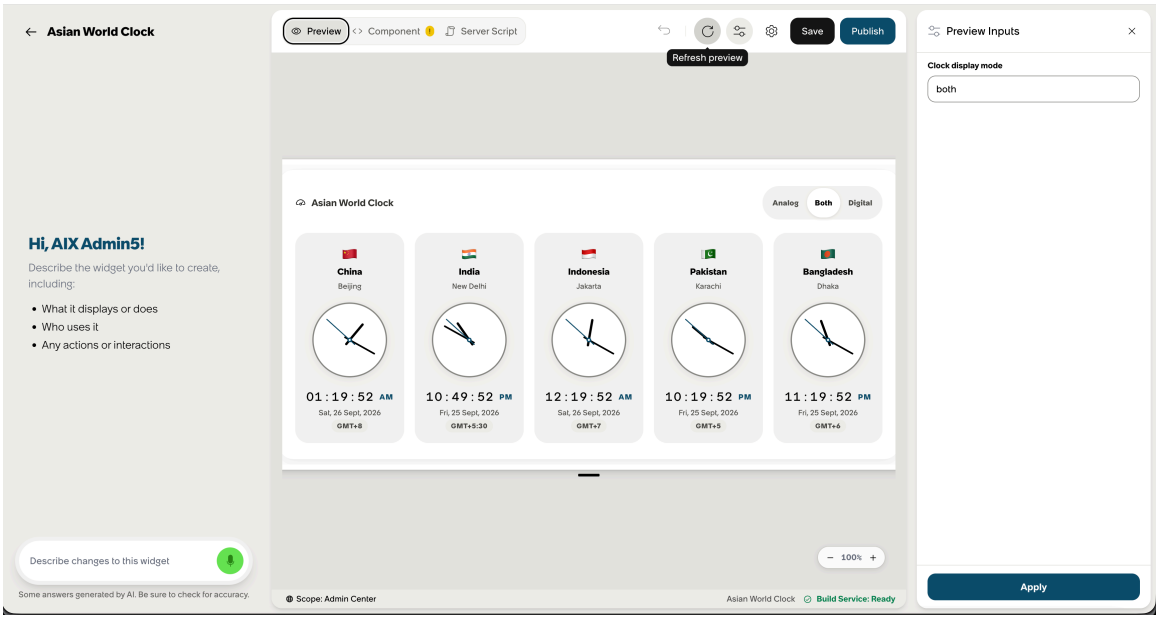
Preview tab

The **Preview** tab renders the widget live, in a resizable and zoomable viewport:

- Drag the handles on the left, right, bottom, or corner to resize the preview.
- Use the zoom controls in the bottom-right corner, or hold Ctrl/Cmd and scroll, to zoom in and out. The zoom controls reflow the layout rather than just scaling it visually.
- Select **Refresh** to reload the preview from the latest build, bypassing the short cache Widget Builder keeps when you switch tabs.

If the preview can't load, it shows the specific problem, such as a missing dependency, a timeout, an import failure, or a registration error with a **Retry** button. A widget that hasn't been built yet shows "Preview not available yet — Click Build to compile" instead of an error.

Preview tab with resize handles and zoom controls



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Property panel

The property panel on the right has two modes:

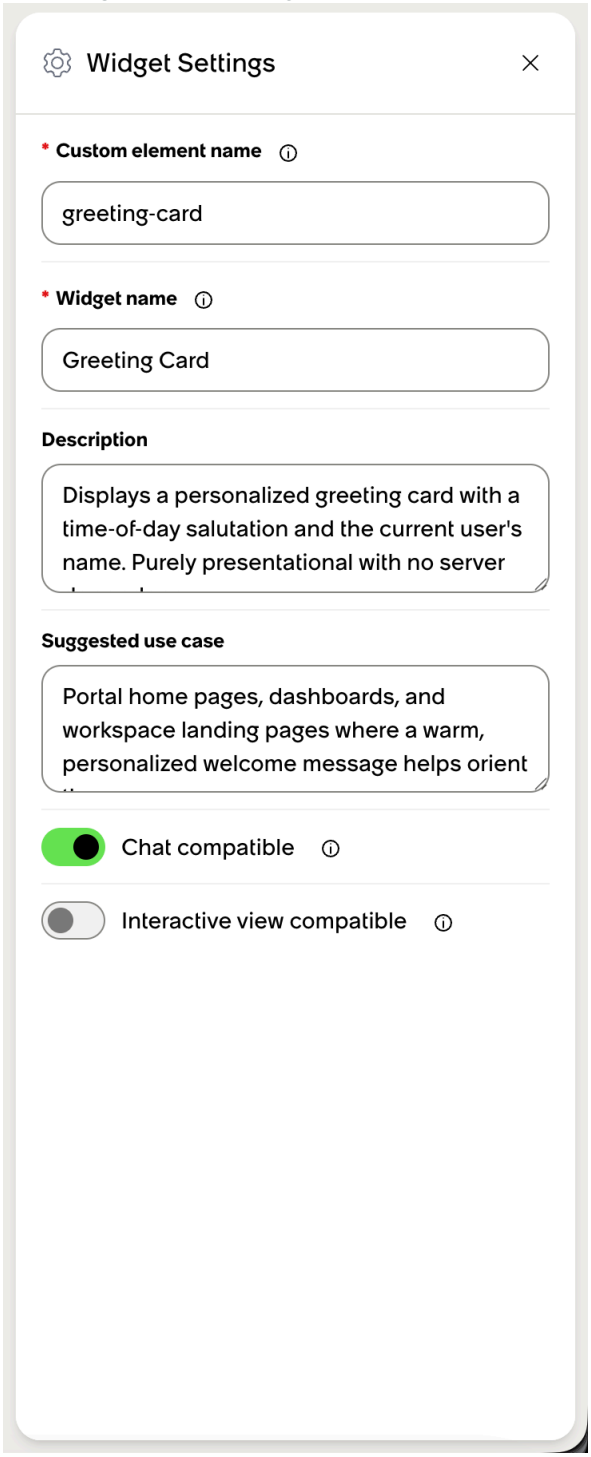
Widget Properties

A form for the widget's metadata: name, ID, description, category, and similar fields.

Preview Props

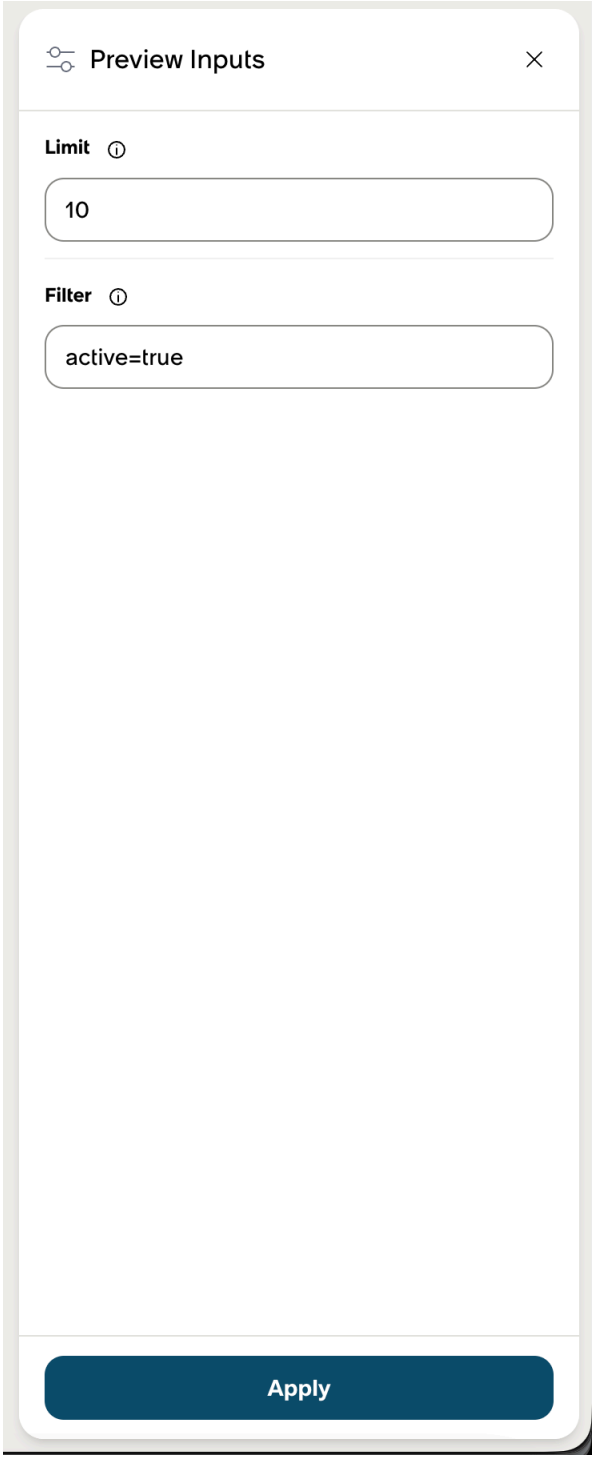
Inject property values into the running preview to see how the widget behaves with different inputs, without changing the widget's saved defaults.

Property panel in Widget Properties mode



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Property panel in Preview Props mode



Save, Publish, and Build

Save

Builds the widget without deploying it. The preview updates right away, so you can iterate quickly. If a build fails, your edits aren't lost — Widget Builder saves a draft and shows the build error. Fix the issue and save or publish again.

Publish

Builds and deploys the widget to the instance. A new or cloned widget's first publish shows "Widget created and published"; later publishes show "Widget published". If publishing fails after a successful build, Widget Builder shows the error.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Customize a base system widget

Customize a base system or store app widget by cloning it to get an editable copy, or migrate a legacy widget to the current Lux format.

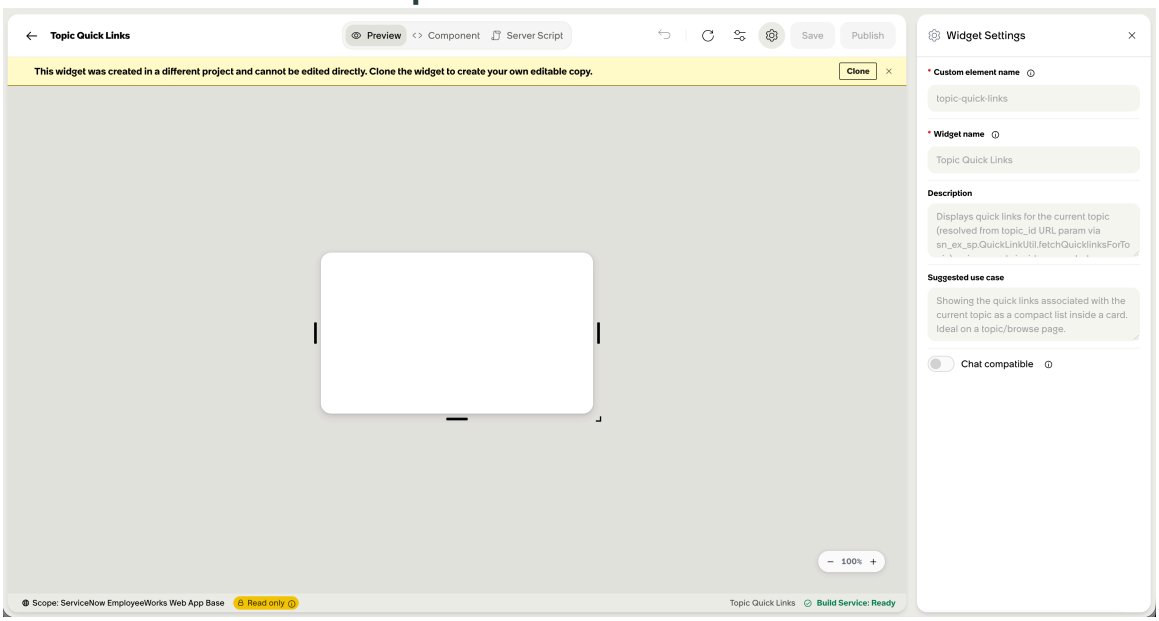
Before you begin

Role required: admin

About this task

A widget included with the base system, a widget that came from a store app, or one you haven't created or cloned yourself, opens read-only in the Editor. Widget Builder shows a banner explaining why: "This widget was created in a different project and cannot be edited directly. Clone the widget to create your own editable copy." Both code editors are read-only, and **Save**/**Publish** are disabled. Trying to delete such a widget from the Widgets catalog is blocked outright, with no confirmation dialog. That's a stricter block than editing, since removing it can affect other widgets that depend on it.

Locked editor banner with an explanation and a Clone button



Note: Clone-to-customize is a new way of customizing widgets, made possible by Lux's code-as-source-of-truth architecture. Because a widget's code is the definitive record, cloning it gives you a real, independent copy to edit. There's no risk of your changes being lost or conflicting the next time the platform or app is upgraded.

Procedure

- Clone to customize

1. Select **Clone** on the banner in the locked editor, or from the widget's action menu in the Widgets catalog.
2. Widget Builder creates a new widget, seeded with the base system or store app widget's current source.
   There's no restriction on scope, you can clone into any application scope, including the base system scope itself.
3. Edit the clone freely, as nothing you do to it changes the original.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

4. Publish the clone to make your customized version available.
   The original base system or store app widget is unaffected and remains locked.

- Migrate a legacy widget to Lux

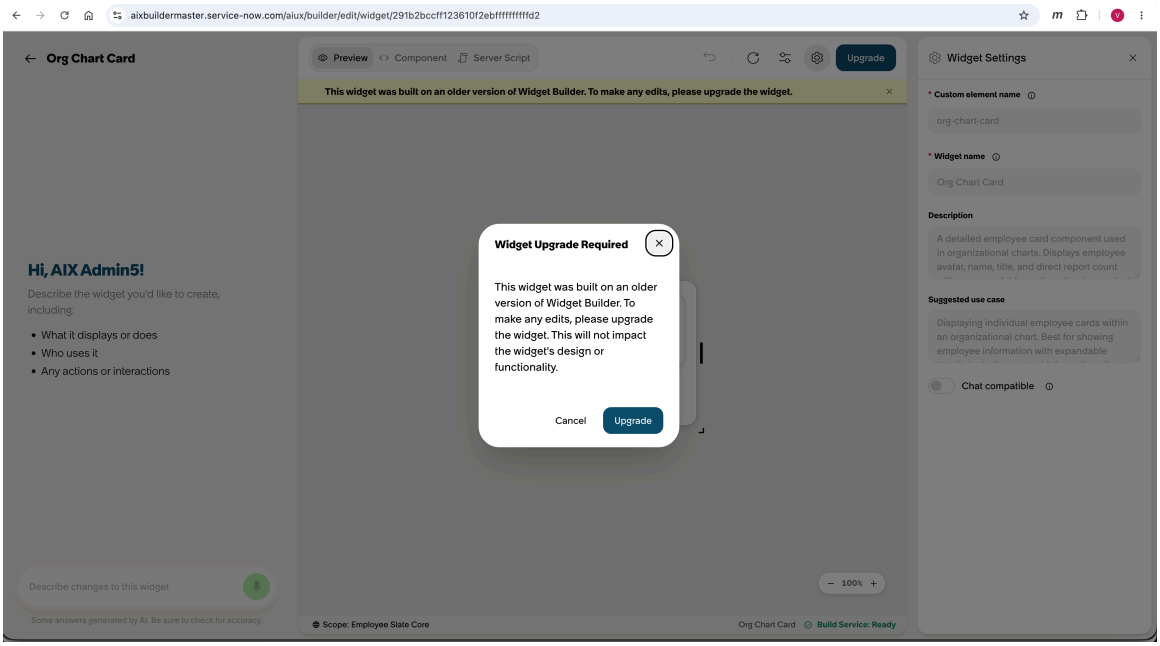
  Older, pre-Lux widgets show an **Upgrade** button in the editor toolbar. Selecting it rewrites the widget's source into the current Lux component format locally, using the widget's existing metadata. No AI call is made, and nothing is sent over the network.

  1. Select **Upgrade** in the toolbar.

  2. Review the rewritten source in the Component tab.
     Widget Builder confirms success, or shows a warning if it couldn't fully convert something.

  3. Select **Save**, then **Publish** as usual to complete the migration. Upgrading doesn't save or publish on its own.

Upgrade button in the editor toolbar



Configure experiences

Configure experiences by setting the properties that drive each instance and using the inline and admin editing surfaces to manage those values. You can arrange widgets into layout regions and apply themes to control appearance without touching code.

Topics in this section cover the configuration model, the configurable property types, the inline configuration modes, and the in-page editor that administrators use. Topics also cover how to edit and save page layouts, and how to apply themes to an app—from brand configuration to individual design tokens.

Configuration model

Lux makes a component's behavior configurable at three levels: code, admin policy, and individual preference without three separate systems to learn. You add one decorator, `@config`, to a class field to opt a property into all of it.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Configuration is stored as metadata override layers rather than as a code fork, so an admin's changes reapply cleanly across upgrades.

Three players

Every configurable value passes through the same three parties:

Declare: developer

Tags which properties are configurable, sets the code-level default, and defines the choices, types, and labels that drive the generated config UI. Nothing is editable until it is declared.

Configure: customer admin

Configures shipped pages and widgets directly on the live page, in non-production, then promotes the change through an update set.

Personalize: end user

Adjusts their own view within admin-defined bounds.

As the developer you only need to read `this.propertyName` in your component. The framework resolves the value that getter returns.

Configuration is not a replacement for UI Builder, a tool for building widgets or pages, or a standalone no-code builder. This is a pro-code workflow: a developer writes the declaration, and an admin configures what the developer declared.

Default value

A `@config` property resolves to its developer default, the value assigned in your code.

Promotion and upgrade safety

`sys_aix_configuration` extends `sys_metadata`, so admin configuration participates in update sets exactly like a business rule or a UI policy; no special export or import path.

Related topics

Configurable properties for Lux experiences

Inline configuration

#unique_1227

Regions and the wrapper contract

Theming model

#unique_1230

Configurable properties for Lux experiences

A property becomes configurable and visible in the props pane when you add a decorator to it. The decorator's options set the field type that the props pane renders, and also set the storage key that the resolved value is read back from.

@config decorator

```
import {AIUXElement, config} from
 '@servicenow/aiux-components-core';
import {customElement} from 'lit/decorators.js';
import {html} from 'lit';
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
@customElement('my-widget')
export default class MyWidget extends AIUXElement {
  @config({
    label: 'Display Density',
    description: 'Controls how compact the widget appears',
    group: 'Appearance',
    type: 'select',
    choices: [
      {value: 'compact', label: 'Compact'},
      {value: 'normal', label: 'Normal'},
      {value: 'comfortable', label: 'Comfortable'}
    ]
  })
  density = 'normal'; // ← the developer default

  render() {
    return html`<div class="widget
 density-${this.density}">…</div>`;
  }

  _onDensityChange(e) {
    this.updateConfig('density', e.detail.value);
  }
}
```

The decorator does three things:

1. Registers metadata. The `label`, `description`, `group`, `choices`, and `type` you pass go into the component's manifest. The props pane auto-generates the settings UI from it, and AIEL reads the same manifest.
2. Turns the field into a getter: `get density() { return this.aiuxConfigData?.density ?? 'normal'; }`. The `this.aiuxConfigData` object is populated from the `aiux-config-data` attribute on the element, which carries the fully-resolved payload for this user.
3. Makes your initial value the bottom layer: the developer default, used when no other layer has a value.

Two constraints that produce silent bugs if you miss them:

- Never assign to a `@config` field. The statement `this.density = 'compact'` is a no-op, because the field is a getter. Use `this.updateConfig('density', 'compact')`, or `setPersonalization` or `setConfigValue` from the config service.
- Never combine `@config` with a parent-controlled `@property()`. The getter reads from `aiuxConfigData`, and a parent setting the same attribute conflicts with it. Use `@state()`, or no Lit decorator at all, for the backing field.

@config options

Option	Type	Description
`label` (required)	string	Name shown in the config UI, and the name AIEL uses to expose the property.
`description`	string	Longer explanation, rendered as tooltip text.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Type	Description
`group`	string (default `"General"`)	Groups related settings in the panel. See Grouping a crowded panel.
`order`	number	Sort position within the group; lower values appear first.
`choices`	`{value, label}[]`	A closed set of values. Renders a select; combine with `multi: true` for a multi-select.
`type`	string	Explicit field type. Takes priority over the legacy constructor flags. See Field types.
`table`	string	Required with `type: 'reference-picker'`; the table to search.
`min` / `max` / `step`	number	Constraints for number fields.
`rows` / `maxlength` / `minlength` / `placeholder`	number or string	Options for textarea and string fields. `rows` defaults to 3.
`pattern`	string	Regex validation for string and URL inputs.
`dependsOn`	object	Conditional visibility. See Conditional fields.
`list`	`{table, query?, view?, columns?}`	Renders a pencil button opening a list iframe modal. Use with `type: Array`; the stored value is the resulting ordered record set.

Field types

Type	Field rendered	Extra options
`'string'` (fallback)	Text input	`maxlength`, `minlength`, `placeholder`, `pattern`
`'url'`	URL input	same as string
`'textarea'`	Textarea	`rows`, `maxlength`, `minlength`, `placeholder`
`Boolean`	Toggle switch	
`Number`	Number input	`min`, `max`, `step`
`choices[]` without `multi`	Drop-down list	`choices` required
`'select-multi'`	Multi-select	`choices` required
`'datetime'`	Date and time picker	`hideTime` renders date-only
`'color'`	Color swatch picker	`choices` optional; defaults to a built-in palette
`'reference-picker'`	Single record typeahead	`table` required
`'table-picker'`	Table name typeahead	`items` optionally pre-seeds the list

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

type	Field rendered	Extra options
`'table-reference'`	Two-step: pick table, then record	
`'role-picker'`	Role typeahead over `sys_user_role`	`items` optionally pre-seeds the list
`'condition-builder'`	Condition builder modal	`tableName` fixes the table; omit for a dynamic picker
`'custom'`	Author-supplied element	`component` required: the element tag name. Pilot.

Values come out of storage as JSON strings, and `type` drives the coercion.

```
@config({label: 'Show SLA Fields', type: 'boolean'})
showSlaFields = true;

@config({label: 'Page Size', type: 'number', min: 5, max: 100,
 step: 5})
pageSize = 20;

@config({label: 'Default Assignment Group', type:
 'reference-picker', table: 'sys_user_group'})
defaultGroup = '';

@config({label: 'Notes', type: 'textarea', rows: 5, placeholder:
 'Enter notes…'})
notes = '';
```

Legacy constructor flags (`referencePicker: true` with `table`, `tablePicker: true`, `tableReference: true`, `rolePicker: true`): resolve to the equivalent string `type` values for backward compatibility. Use the explicit string `type` in new code; the explicit string type value is used when both are set.

Grouping a crowded panel

Give each property a `group` id, and optionally declare `static configGroups` to control order and labels:

```
class MyWidget extends AIUXElement {
  static configGroups = [
    {id: 'text', label: 'Text fields'},
    {id: 'selection', label: 'Selection fields'},
    {id: 'data', label: 'Data fields', defaultExpanded: false}
  ];

  @config({label: 'Title', type: String, group: 'text'})
  title;
}
```

Groups render in the declared order, or in first-appearance order if you omit `configGroups`. Each group's `defaultExpanded` defaults to `true`. Properties with no group, or with a group id not listed, render ungrouped after the accordions.

Grouping earns its keep when the pane is crowded. What matters is how far an admin has to navigate to reach the property they configure most often, not the number of groups.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Conditional fields (dependsOn)

A field can declare that it appears only when a sibling satisfies a condition.

Mode	Schema	Shown when
exists	`dependsOn: {key: 'parentProp'}`	Parent value is non-empty
equals	`dependsOn: {key: 'parentProp', value: 'exact'}`	Parent strictly equals `value`
one-of	`dependsOn: {key: 'parentProp', value: ['a', 'b']}`	Parent is in the array

```
@config({
  label: 'Temperature',
  type: Number,
  dependsOn: {key: 'cookingMethod', value: ['grill', 'oven']}
})
temperature;
```

When a parent value changes, any field whose condition is no longer satisfied resets to its default, and the cascade propagates transitively. Circular `dependsOn` chains are detected at decoration time and logged as an error rather than thrown. The circular segment is silently ignored, which can produce unexpected auto-clear behavior.

Configuration ids for nested components

A page's resolved payload is one flat object with dot-namespaced keys: `"other.button.density": "compact"`. A component gets its slice of that payload by carrying a configuration id.

In a region template, write the attribute.

```
// regions/home/body.js
export default () => region`
  <row>
    <column span=${8}><stat-card
 aiux-config-id="stat-card-1" /></column>
  </row>
`;
```

The `aiux-config-id` attribute is the one region-DSL editor attribute the runtime does not consume and drop. It compiles to exactly the binding a hand-written `elementId(host, 'stat-card-1')` on `aiux-config-data` produces, and that directive stamps `data-aiux-element-id` on the rendered element. So the widget becomes an ordinary configuration entry: stored in `sys_aix_configuration` under `<id>.<prop>`, read back by the same `@config` getter, reset by the same call.

Three rules apply to the attribute:

- It must be a non-empty string. A bare `aiux-config-id` with no value throws, because the value is used both as a DOM id and as an `aiuxConfigData` key prefix.
- It is valid only on a widget tag, not on `row` or `column`.
- Without it, the widget cannot be configured at all: see Placing Lux widgets in regions for what that looks like to an admin.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Avoid `/` in an id. The id travels in a URL path segment when a reset is sent, where a slash either gets rejected or normalized back into a separator. Widgets the visual editor adds are stamped `<tag>__<nonce>` for this reason.

In a hand-written Lit template, call the function. The `elementId(this, 'id')` call is the underlying mechanism, and what you write when the parent is a plain Lit `render()` rather than a region:

```
import {AIUXElement, elementId, config} from
 '@servicenow/aiux-components-core';

@customElement('some-page')
class SomePage extends AIUXElement {
  @config({label: 'Greeting'})
  greeting = 'Hello there!';

  render() {
    return html`
      <h1>${this.greeting}</h1>
      <other-element aiux-config-data=${elementId(this,
 'other')}></other-element>
    `;
  }
}
```

Either way, each element does exactly two things with its slice:

1. Peels its own keys: assigns values whose key matches one of its own `@config` field names.
2. Forwards subtrees: for every id it declares, slices all keys prefixed `id.`, strips the prefix, and passes the rest down.

So given `greeting`, `other.propA`, and `other.button.density`, the page keeps `greeting` and forwards the other two. A page author only needs to specify their direct children's ids; the child is responsible for slicing and forwarding to its own children. The id must be unique among siblings at that level, and a child absent from the payload gets an empty object, so its `@config` defaults take over.

Storage key format

Element	Stored key
Top-level page property	`propKey`
Direct child (id chain `['header']`)	`header.propKey`
Grandchild (id chain `['header', 'badge']`)	`header.badge.propKey`

The `tag` column on the stored row is always the top-level element's custom-element name, not the child's.

Config service methods

The config service is a module, not a DOM element, and it works on top-level element tag names; it does not work on nested children directly.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Method	What it does
`setConfigValue(tag, key, value, options)`	Change the admin-layer value. Optimistic: applies immediately, rolls back if the save fails.
`setPersonalization(tag, key, value, options)`	Change the user's own value, saved to `sys_user_preference`. Optimistic with rollback.
`resetConfig(tag, key, options)`	Clear the current value, falling back to the next lower layer. `{all: true}` clears every layer down to the developer default.
`resetPersonalization(tag, key)`	Clear this user's preference only.

```
import {
  setPersonalization,
  resetPersonalization,
  resetConfig
} from '@servicenow/aiux-config-service';

await setPersonalization('itsm-incident-list', 'density',
 'compact');
await resetPersonalization('itsm-incident-list', 'density');
await resetConfig('itsm-incident-list', 'density', {all: true});
```

Configuration modes

`@config` is one of four modes a property or a whole widget can use, in increasing order of author effort.

Order	Mode	Declared with	Editing UI	Saves to
1	Independent config	`@config` on a field	The framework props pane: value field plus lock toggle per property	`sys_aix_configuration`
2	External config	`@externalConfig` on the class	Same pane, plus a link out to a list, record, or URL	The author's own table
3	Custom URL	`static configMode = 'inline-simple'`	Pane stays; a button opens an author-supplied URL in an iframe	Wherever that tool saves, not through the framework
4	Self-config	`static configMode = 'inline-auto'`	The widget renders its own editor, fully interactive	`sys_aix_configuration`, through the framework's save pipeline

Mode 1 is the default unless the widget is genuinely complex: list, form, action bar, data visualization. Modes 3 and 4 are the path for those specific widget families.

Modes combine. The `configMode` static accepts a comma-separated list: `'inline-auto,inline-simple'`, and either mode can be paired with `@config` decorators on the

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

same widget. The editor then shows both the deep-link button and the props-pane pencil on one outline. A widget can also declare multiple `@externalConfig` entries.

See Inline configuration for each mode in detail.

Actions (@action)

`@action`, a decorator for exposing component methods as AI-invokable, shortcut-configurable actions, is design-stage only. Nothing described for it exists in the current codebase, and it has no entry in the customer-facing configuration documentation. Don't build against it.

Related topics

Configuration model

Inline configuration

#unique_1227

#unique_1232

Inline configuration

Inline configuration turns a property a developer declares in code into something a customer admin can change on a live page, without a code change or a redeploy. An author picks one of four modes per property, and the framework stores and reads back the resulting values.

The mechanics of declaring a property are in Configurable properties for Lux experiences; what the admin sees is in #unique_1227.

Four configuration modes

An author picks a mode per property, or per widget for self-config, in increasing order of effort. A single widget can mix modes: some properties on the standard popover, others deep-linked or self-configured.

Order	Mode	Declared with	What the admin gets	Where data saves
1	Independent config	`@config` on a field	The value field in the framework props pane, plus a lock toggle	`sys_aix_configuration`
2	External config	`@externalConfig` on the class	Same pane, plus a link out to a list, record, or URL	The author's own table
3	Custom URL	`static configMode = 'inline-simple'`	Pane stays; a button opens an author-supplied URL full-screen in an iframe	Wherever the linked tool saves; not through the framework
4	Self-config	`static configMode = 'inline-auto'`	Nothing framework-drawn; the widget renders its own editor	`sys_aix_configuration`, through the framework's save pipeline

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Mode 1 applies to most properties. Only complex widgets (list, form, action bar, data visualization) need one of the others.

External config with @externalConfig

A class-level decorator with no effect on resolution, storage, or the priority stack. It only injects a link into the config panel, so the admin has a quick path to related platform data: assignment group rules, SLA definitions, approval policies.

```
@customElement('some-tag')
@externalConfig({
  key: 'incidentData',
  list: {table: 'incident', fixedFilter: 'active=true'}
})
class SomeElement extends AIUXElement {}
```

Three variants: `list: {table, fixedFilter}`, `record: {table, sysId, view}`, or `link: {url}`.

Custom URL with inline-simple

For complex platform configuration where full self-config is not feasible. The widget's configuration lives in an external platform tool, opened in an iframe.

```
@customElement('my-component')
export class MyComponent extends AIUXElement {
  static configMode = 'inline-simple';

  provideConfigDeepLinking() {
    return {
      label: 'Configure Incident List',
      url: `${window.location.origin}/incident_list.do`
    };
  }
}
```

Two requirements: the `configMode` static (a plain property, not a decorator) and an override of `provideConfigDeepLinking()`, which is a no-op on `AIUXElement` by default. Return a single `{label, url}`, an array of them for several buttons, or `null` for none. The editor renders one link-icon button per entry on the element's outline.

> ⚠ **Warning:** Configuration set inside the opened URL is not returned to the widget. The framework does not receive or store values from the linked tool.

Self-config with inline-auto

For widgets when a flat property form can't express the change: drag-and-drop layouts, a widget toolbox, a canvas, or curating an ordered collection. The framework delegates edit mode only; the UI, the affordances, and the autosave behavior are the author's.

```
class SnLayoutRenderer extends AIUXElement {
  static configMode = 'inline-auto';

  // The framework calls this when config mode opens.
  openConfigEditor({fieldValues, onChange, onSave, onClose}) {
    this._inlineConfig = {onChange, onSave, onClose};
    this._editMode = true;
  }
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
  closeConfigEditor() {
    this._editMode = false;
  }

  _persistChange(key, value) {
    this._inlineConfig?.onChange(key, structuredClone(value));
  }
}
```

Callback	Direction	What it does
`fieldValues`	in	Current resolved value per `selfConfigSchema` key, admin and user layers applied
`onChange(key, value)`	out	Stage a pending edit. Safe to call repeatedly; pass a clone, not a live reference.
`onSave()`	out	Persist all staged edits and exit config mode
`onClose()`	out	Exit, discarding anything unsaved

On open, the editor scans the DOM to collect every configurable entry. Self-config widgets are handed their own editor instead of the standard per-property popover everyone else gets.

Self-config does not bypass the priority stack. The framework writes values to the admin layer through the same save path the props pane uses. The widget owns the editing UI, not the storage.

Custom component in the props pane

Not a fifth mode. A bespoke component is injected into the standard props pane, keeping the familiar surface while still saving to `sys_aix_configuration`. Source marks this as available on request only. It also instructs authors to consult the configuration product and UX owners first, so the injected UI stays consistent with the rest of the configuration surface.

How configuration is stored

sys_aix_configuration

Extends `sys_metadata`, indexed on `(page_path, tag)`, and backs every admin layer.

Column	Notes
`page_path`	Required: matches the request's page path
`experience_path`	Required: matches the experience basename
`tag`	Required: the top-level element's custom-element name
`name`	Required: the full storage key
`value`	JSON, validated at write time
`condition`	Optional GlideScript, evaluated server-side
`order`	Evaluation order for conditional rows; default 100
`personalization_disallowed`	Blocks the user preference from applying; default false

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Column	Notes
`sys_domain` / `sys_domain_path`	Standard domain separation

The `name` column encodes the element's position in the component tree: `propKey` for a top-level property, `header.propKey` or `header.badge.propKey` for a nested one. See Configurable properties for Lux experiences for how an id chain produces that key.

sys_user_preference

No schema changes. User layers use dot-separated keys:

- Scoped: `aix.config.{pageId}.{tag}.{instanceId}.{name}`
- Global: `aix.config-global.{tag}.{name}`

Related topics

Configuration model

Configurable properties for Lux experiences

#unique_1227

Regions and the wrapper contract

Layout configuration

A config admin can restructure a page's layout directly on the rendered page, without modifying code, by editing and persisting the regions a page exposes. This section covers the region and wrapper contract that makes a layout editable, what editing and persisting a layout costs, and how widgets get placed into regions.

Regions and the wrapper contract

A config admin can restructure a page's layout directly on the rendered page and persist the result as an admin override. Config admins can add and remove rows, columns, and widgets, resize column spans, change scrolling, and move containers.

A region is a named, addressable area of a page. A region is authored in code with the `region` template and rendered inside a wrapper element that contains the region. The preceding topics in this series cover admin overrides for property values. This topic covers admin overrides for layouts, including how to edit an already-rendered region and the persistence path that makes an edit stick.

Region wrapper contract

Every rendered region is enclosed in exactly one wrapper element. The wrapper is part of the rendered-output contract and uses a specific data attribute for identification.

A region's name is the thing an override is keyed on.

Region spacing

A region's spacing is not something the editor exposes. The runtime emits one uniform gutter between all widgets and no padding. Spacing is retuned through the `--layout-gap` custom property, or `--spacing-layout` for one subtree, rather than through the DSL. The paddings-and-gaps popover that used to write `gap` and `padding` is gone with them.

`gap` and `padding` are deprecated on the authoring side too. Both are still parsed and consumed by the region, so they never reach the rendered element. But they are ignored on

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

any tag, with any value, and warned once per attribute and tag pair. They existed to tune spacing per container, which is exactly what a uniform gutter cannot allow. A `padding` on a column, or a smaller `gap` on a nested row, doubles or breaks the gutter at that nesting boundary. `margin` was considered for the same purpose and deliberately left out.

Which element owns a region

One element per page owns its regions, and which one depends on the kind of page.

Page	Region owner	Row keyed on
A normal custom page (anything an app author wrote)	The page widget, and nothing else	The page's widget record, read from the manifest with no round trip
A core-ui page (a list page or a record page)	The view widget, and never the page	That widget's `sys_aix_widget` row, read from the instance

The page is the outermost tag in the region's custom-element ancestor chain that the client manifest carries a widget record for. Layouts and app-level lifecycle modules aren't in that set, so they are skipped without a test of their own.

Two consequences follow:

- A normal page's child widgets are not searched for regions. A region is the page's own layout. A widget that renders one is describing a layout it does not own. An override keyed on the page would claim the page owns something its child authored. Such a region is refused rather than reassigned upward; no session, nothing to save; and the widget entries inside it are untouched.
- On a core-ui page, being the page widget is not enough. The page itself is platform chrome nobody configures. The page body is a view widget resolved per request. That widget is what an admin is rearranging, so it is the only element on the page allowed to own a region. Several views can be configured and only the one the platform's view-rule engine picked is on the page, so the DOM settles which one it was.

The printer

The printer converts an edited tree to storable text. Before it returns the text, it verifies its own output. The printer splits, decodes, and re-parses the text it printed. It then compares the two trees. If the trees do not match, the printer throws an error.

This behavior is necessary, not cautious. The renderer catches render errors and falls back to the code-authored template without warning. A mis-printed edit does not cause an error. It causes an edit that appears to do nothing. The printer throws an error at the print step to prevent this.

What the printer rejects

The printer rejects these three items:

- Object-valued interpolations
- Strings that contain `}`
- `<raw>` blocks

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The platform save-time splitter treats the first `}` after a `${` as the end of an interpolation. This is why the printer rejects object-valued interpolations and strings that contain `}`. An admin can trigger the second condition by typing an ordinary label.

Stored DSL rejects `<raw>` blocks as a cross-site scripting risk. A region that contains a `<raw>` block is not editable.

Unstored regions

Unstored regions are editable. Override rows exist only for regions that an admin has already overridden. A region that has never been modified has no row.

The renderer records the code-authored tree it walked. This gives the editor a starting point when no stored row exists. Under server-side rendering, this recording does nothing. This is correct because the editor runs client-side.

The deprecated <stack> tag

The deprecated `<stack>` tag does not reach the editor. Both editor inputs go through the same parser. The parser rewrites every `<stack>` tag before validation runs. It converts each `<stack>` into a `row` that holds one full-width `column`. The parser drops all attributes on the `<stack>` tag and writes a named console warning for each one. The first time an admin saves a region that contained a `<stack>` tag, the stored text uses rows and columns.

Related topics

Editing and persisting layouts

Placing Lux widgets in regions

#unique_1217

Configurable properties for Lux experiences

#unique_1227

Configuration model

Editing and persisting layouts

Configuration mode unlocks every region on a page. Each region edits and saves independently. Learn about layout ownership, persistence, and editor limitations.

Configuration mode and regions

Configuration mode unlocks every region on the page when it opens. There are no modes to enter or leave. The toolbar displays one label, **Configuration**. Each boundary on the page identifies its container type on its own tab.

A region that cannot be loaded gets no session and no canvas. The system does not display an error message. With every region loaded at once, a page with one unsupported region among four would otherwise announce a refusal the administrator never asked for. The region's outline carries the reason instead. Typically reasons for the editor not loading a region include the editor not being able to reconstruct the region or a region that contains a `<raw>` block.

Each region is its own session

A page has several regions, and each one is a separate override row with its own baseline, pending text, and working tree. Sessions are therefore keyed `ownerTag::regionName` which is the same pair `sys_aix_page_region` is unique on. The separator is a doubled colon rather than a single one because a region name may contain `.`, `-`, and `_` but not a colon. An

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

owner tag is a custom-element name, so a doubled colon cannot occur inside either half and the split needs no escaping.

One editor instance exists per region, built as configuration mode opens and torn down as it closes. A single active region key says which region the selection, the open menu, and any gesture in flight belong to. It is not a mode, and nothing is locked while it points at one region. Switching regions clears the selection rather than carrying a path that addresses a different tree. The system prevents switching while a drag or a resize is live.

Layout editing as ownership

An override is a new row, and from the moment it is saved the region's layout stops taking upgrades while the widgets inside it keep taking theirs. A modal dialog displays this information before the action occurs.

The system prompts on the first edit of an unstored region, not on entry. There is no entry to gate, and asking on entry would mean one modal per region the instant the editor opened, about layouts the administrator may never touch. The first edit is the moment the consequence becomes real, because that edit is what creates the row. A region already stored is already overridden, and is never asked about.

A recording edit is held back and re-run through the same path on confirmation, so it gets the same validation and announcement it would have had. A pick-up for a drag is refused outright instead. There is nothing to replay, because where a drop goes depends on where the pointer goes next, so the administrator drags again after they have confirmed.

Persistence

Region overrides are written to sys_aix_page_region directly through the Table API, deliberately rather than through a framework service route. A region override is one field on one platform row, and the Table API already enforces the write access control.

- `(widget, region)` is unique on the table, so a read for one region returns at most one row. An empty result indicates that the region is code-authored with no override yet and drives an insert. A thrown error means the read itself failed, and is never collapsed into "no override," so a refused read does not get reported as an absent one.
- An insert is valid only when the read came back empty; the unique index refuses a second insert for an existing pair.
- An update patches the existing row's **layout** field.

Restore defaults deletes the row

The region's gear menu holds one item, and it is the inverse of taking ownership: the region stops being overridden. Confirming it stages the deletion of the region's row, so the layout the app ships stands again and takes upgrades again.

> ⚠ **Warning:** The action deletes the row rather than clearing the **layout** field. A row with an empty **layout** field still takes precedence over the app's shipped layout, rendering the region as empty.

A 404 response on the delete is treated as success. The row being absent is the intended outcome. The action is available whenever a row exists, regardless of whether the current session has unsaved edits. A region overridden in an earlier session opens without pending changes, which is the common case where restoring the app's shipped layout is relevant.

The deletion is staged, not applied immediately. It enables **Save** and is discarded by exiting the editor, the same as a layout edit. Staging a restore drops any staged layout text for that region.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

A subsequent edit drops the staged delete. Because both are writes to the same row, a save applies only the later instruction.

What is on screen follows immediately, because a staged restore that left the overridden layout rendered would be a menu item with no visible effect until a reload. Dropping the override is also what makes the default layout readable to the editor for the first time. An overridden region has no recorded default until its override goes away.

A save ends the session

Saving persists the layout and exits configuration mode. Because every region is unlocked for the entire session, remaining after a save leaves the administrator on a page full of editing controls with nothing left to commit. Every save path exits, not only the **Save** toolbar button. When a save fails, the editor stays in its current state and displays the reason on the toolbar.

Drag preview and cross-region moves

While a widget is being dragged, the region shows the layout the release produces. The stand-in is a real element and is the node in flight. Pick-up takes the widget out and holds its element detached, and every seam the pointer crosses moves the stand-in there using the same placement the release would use. Release swaps the widget's element back into the slot the stand-in was holding. The element that lands is the element that was picked up, so the widget keeps its state and does not re-run its loader.

A destination indicated by the pointer does not land until it has been rested on briefly. Applying a placement reshapes the region. Applied on every pointer move, a traveling pointer walked the region through every intermediate shape on the way to the one it wanted. Each of those shapes moved the boxes the administrator was aiming at. Resting first makes the aim a fixed question. A release never waits; a cancel discards.

A widget may not leave its region. A move is a path into one region's tree, committed by printing that region's text. No action in the grammar takes a node out of one region and puts it into another. It would be a delete from one stored row and an insert into a second. That means two prints, two overrides, and a widget whose *`aiux-config-id`* may already be taken in the destination. A drag that arrives in a different region is therefore cancelled with a banner, not merely refused in place. The alternative leaves the administrator holding a ghost over a region that cannot take it, with nothing saying so.

A pointer in the space outside all regions, such as the page margin or the gap between two regions, does not trigger a cross-region move. The system treats it as a pointer in transit.

Empty containers

Three rules, and the difference between them is authorship rather than shape.

- An empty column the layout arrived with is kept, and wears an **Add widget** affordance at its center. An author's empty column, or one an administrator added and has not filled, is a slot waiting for a widget.
- A row that arrives with no columns is refused. The parser rejects a column-less row outright: nothing in it can hold a widget, so it is not a layout. A region holding one does not parse, the editor never loads it, and the region's blocked reason carries the parser's message.
- A container this edit emptied is swept after every mutation. Every column with nothing left in it goes, then every row with no columns, bottom-up, so emptying a column can empty the row that held it. Widths go back to the line, and a wrapper left standing around nothing is unwrapped.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Known limitations

- A widget with no *aiux-config-id* cannot be configured. Its surface carries no pencil, only the chevron. The attribute has to be added to the region's code template, and no control writes it. An *id* written by an administrator would be an address with no rows behind it, and no way to tell it from one the author meant. Widgets the editor adds are stamped automatically.
- Undo and redo do not cover region layout edits, and neither does any per-region revert. The undo stack is a widget-property stack. There is no step-by-step history of layout edits, and no way to drop one region's session edits while keeping another's. Exiting the editor discards every pending change at once, and "Restore defaults" is a different act.
- Widget-internal state and scroll are lost on every structural edit. A structural change remounts every widget in the region by design; only the region's own scroll offsets are preserved. A move is the exception, because the dragged widget's element is put back rather than rebuilt. A placement that synthesizes a container still remounts the rest of the region.

Placing Lux widgets in regions

A seam is a placement point in a region, the edge of a widget or the gap between two widgets. Use seams to add widgets from the toolbox and configure widgets placed inside a region.

Seams and widget placement

A seam is a placement point in a region, the edge of a widget or the gap between two widgets. It is not an address. An administrator points at a widget, not at the container around it. Both the + affordances and drag-and-drop resolve through the same function, so the two gestures produce the same structure for the same intent.

A widget's top and bottom edges insert directly into the column the widget is already in. No container is created, and the widget stays mounted. A widget's side edges behave differently depending on whether the widget shares its column:

The pointed-at widget	The placement	Why
Alone in its column	A new cell of the enclosing row, on that side	The cell is the widget, so "beside this" and "beside this cell" are the same request — and the flat shape says it with one fewer invisible container. It costs the row's width rather than a nesting level, so it works at the depth cap where a wrap can't.
Sharing its column	The widget's own slot is wrapped in a nested row of two even cells	Adding a cell to the row would move the widget's stacked siblings too, which is not what pointing at one of them means. The stack stays; the row appears between the two widgets it was already between.

A container boundary is the gap between two siblings, or an empty box. It resolves to whatever chain the grammar needs, outside in. A region's top level takes rows only, so a widget dropped there gets a row and a column. A row takes columns only, so a widget dropped between two cells gets one column. A column takes widgets, so nothing is synthesized.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Where corridors overlap, an interior seam, one sitting in a real gap between two siblings, wins over an end seam sitting on its container's own edge. Only then does the deepest win. Depth alone was wrong in exactly the case the gutters are for. A full-height column's own bottom edge is also its row's bottom edge, so aiming at the gap between two bands appended to the preceding band instead.

Adding a widget from the toolbox

The toolbox is a panel anchored to the specific `+` that was pressed. One search runs per term and three tabs narrow its result client-side, so switching tabs is not a network round trip for an answer already in memory.

Tab	What it shows	Ordering
All widgets	The search result verbatim	Name, A-Z or Z-A
Recents	The intersection with a stored tag list	Recency; the name sort is inactive
My widgets	Widgets whose scope row is an application built on this instance	Name, A-Z or Z-A

My widgets can't be answered from the scope name. An app authored on the instance and one installed from the ServiceNow Store are both prefixed the same way. The question is which table the scope row sits on.

Recents lives in the browser's local storage, holds the most recent twelve, and writes on the pick rather than on Apply or Save. Re-adding a widget already in the list moves it to the front. The stored value validates on read rather than trusted, and a remembered tag with no row in the current catalog drops. So a stale entry can't offer a widget that no longer exists.

A thumbtack keeps the panel open after an add, advancing the insertion point so successive picks stack in order. It is skipped when the placement needed a container to be built. Subdividing or wrapping restructures the tree, and every path the plan named can be stale the instant it is applied.

Configuring a widget inside a region

A region controls a widget's layout, not its properties.

Configuration is an override of a property, so there is one system, and `aiux-config-id` is what opts a widget into it. The attribute compiles to exactly the binding a hand-written `elementId(host, 'id')` produces, and that stamps `data-aiux-element-id` on the rendered element. A region widget carrying one is therefore an ordinary configuration entry. It is found by the same DOM scan, stored in `sys_aix_configuration` under `<id>.<prop>`, read back by the same `@config` getter, and reset by the same call. See Configurable properties for Lux experiences for the attribute itself.

Four consequences follow:

- A widget without an `aiux-config-id` has no address, so keys cannot be saved or retrieved. Its surface shows a chevron instead of a split button.
- Every widget the editor adds gets one, deterministically. New nodes are stamped `<tag>__<nonce>`, using the lowest unused nonce across the region. Colliding IDs would give two widgets the same configuration address, so the rule avoids all existing IDs, not just those of the same tag. IDs the region arrived with are never changed; renumbering one would orphan its stored configuration.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Widget and region configuration are always available together, and neither is exclusive to a mode. The widget's pencil opens its properties pane, while the region's tab and canvas are its own controls. These are distinct affordances, so suppressing the widget's pencil inside a region would override the author's intent.
- Property overrides are staged on the page component's resolved configuration payload and applied on the next render. No layout remounts between keystrokes, and the region's template shape is unchanged, so only the affected attribute updates.

One surface per element

Two layers can draw a box over the same element, and each answers a question the other can't. The outline layer draws per configuration entry; the region canvas draws per node of a loaded region. A region widget carrying `aiux-config-id` is both, and used to wear both. Each element now gets exactly one surface:

The element	Deep links	Pencil	Chevron	Resize brackets	Dashed outline	Drawn by
Widget in a loaded region, with an `aiux-config-id`	One per link	Yes	Yes	Yes	No	Canvas
Widget in a loaded region, without one	One per link	No	Yes	Yes	No	Canvas
Widget outside any region	One per link	Yes	No	No	Yes	Overlay
Configurable element inside a region widget	One per link	Yes	No	No	Yes	Overlay
The region wrapper	—	—	Its tab's gear	No	Yes	Canvas box plus overlay tab

The canvas owns the widget's surface because it is the only layer that handles layout edits and resize. Both operations are addressed by node path. A region widget without an `aiux-config-id` has a path but no configuration entry.

External configuration deep links appear on whichever surface the widget has. A widget whose configuration lives in a platform tool returns one or more label-and-URL pairs. Each pair renders as a leading button on the same control as the pencil: labeled on an outline box corner, icon-only on the canvas toolbar. A deep link does not require an `aiux-config-id` because it is a URL into another tool, not a stored override. A region widget with a deep link and no `aiux-config-id` displays the link and the chevron.

The region wrapper tab displays the word `Region`, not the region's name. The name identifies a slot in a template for authors. Administrators cannot place or act on it. Displaying the name

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

caused the tab to read as a label for the content inside the box rather than a description of the box itself. Administrators need to know the kind, not the name. The name remains the tab's accessible name and the heading the gear menu opens with, where **Restore defaults** must identify which region it restores.

Theming with Lux

Lux generates every theme token from a brand configuration. Developers set an app's default look in code, and admins can change tokens or assign themes per user without a rebuild.

Theming model

Lux generates every color, radius, and font token from a small brand configuration, and a theme record can override any of those tokens per tenant.

Identify which side a given change belongs to before making it:

- `theme.js` and `tailwind.app.css` are developer work. They ship with the app's source, are the same for every tenant, and changing either requires a rebuild.
- The `token_overrides` field on a `sys_aix_theme` record and the multi-theme rows that control which theme a given user gets are admin configuration. They change what a tenant sees without an app source change or a redeploy.

Both are combined in the same generated CSS, which is why one topic covers them.

Brand config instead of component CSS

You don't restyle a Lux app by overriding component CSS. You declare a brand config and the framework generates every color, radius, and font token from it. A scaffolded Lux app's whole theme file is a handful of lines of brand input:

```
// theme.js
export const AppTheme = {
  primary_color: '#032D42',
  accent_color: '#63DF4E',
  neutral_colors: {
    use_hex_value: true,
    hex_value: '#F8F7F4'
  },
  shape: {
    selector_radius: '0.75rem',
    field_radius: '0.5rem',
    boxes_radius: '1rem',
    border_width: '1px'
  }
};
```

`primary_color`, `accent_color`, and `neutral_colors` are base colors, not tokens. Each expands into an 11-stop scale from `50` through `950`, generated once for light mode and once for dark, so contrast holds in both. Semantic tokens map to those stops and are what components consume.

In component styles, use a semantic token instead of a raw stop. Semantic tokens survive a rebrand; raw stops do not.

Token pipeline

Theming is a pipeline of CSS custom properties that flows from brand configuration into every shadow root:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

1. Brand config in. The theme object's colors, shape, fonts, and token overrides are read. They come from the theme record on the instance at request time, or from the app's own theme module.
2. Generation. A generator turns them into accessible color scales, validated `@font-face` rules, up to four font preload links, and a token-override block.
3. Into the document head. Those blocks are emitted as inline `<style>` elements in the server-rendered document, alongside the framework's static token mappings.
4. Into Tailwind. A Tailwind plugin registers the same tokens as base CSS and as prefixed component overrides. The tokens are then reachable as utility classes rather than only as `var()` calls.
5. Into every shadow root. The compiled stylesheet is adopted into each `AIUXElement`'s shadow root. CSS custom properties inherit through shadow boundaries, so a token registered at the document root is readable by every component that adopted the sheet.

Note: If required theme values aren't provided, the generator falls back to the corresponding values from `DEFAULT_THEME`. This verifies the generated CSS has the color scales needed to resolve semantic tokens correctly. For primary and accent colors, this applies when values are omitted or `undefined`; the neutral scale also falls back when its configuration cannot be resolved.

Color scheme and density attributes

Light, dark, and density are attributes rather than themes. Color scheme lives on `data-theme` on the document element, and density on `data-density`. Both are orthogonal to which theme record resolved.

The Lux framework exposes a service for flipping color scheme at runtime, and the user's choice persists across reloads. Most apps don't write that themselves, the nav layout ships a theme toggle you turn on with a feature flag.

Density scales spacing, item heights, and sidebar widths globally, because every Tailwind spacing utility resolves through one spacing unit token. Switching density rescales the whole UI with no component-level change. The three modes and their values can be found in .

Important: A component that needs to react to color scheme rather than set it should reflect its own attribute onto its host and style against that. The generated theme CSS declares `[data-theme]` rules at the document level, which outrank a `:host` rule for the same token. An override written against `data-theme` inside a shadow root does not apply.

Related topics

#unique_1236

theme.js configuration reference

Configuration model

Configure app-level theming

Configure app-level CSS for styles that belong to the app and must stay the same regardless of the tenant or user-resolved theme. Lux supports two optional files for this, `tailwind.app.css` and `head.app.css`.

Before you begin

Role required: admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

About this task

Both files sit at the app root, alongside `aiux.json`. The `aiux build` command discovers both files automatically, so they require no manifest entry or manual import.

The following table compares the two files.

App-level CSS files

File	Scope	Use it for
`tailwind.app.css`	Every app component's shadow root	New or overriding utilities, dynamic-class safelists, and app-specific animations
`head.app.css`	The document `<head>`	CSS that must be registered once for the whole document, especially `@font-face`

For colors, shape, or fonts that a tenant theme should be able to change, use `theme.js` instead. For instructions, see Configure theme-level theming.

Procedure

1. Extend component styles with `tailwind.app.css`

   The app's Tailwind build compiles `tailwind.app.css`, and the compiled sheet is adopted into every app component's shadow root. Use this file when a rule must be available inside components.

   The file supports Tailwind directives and standard CSS, including the items in the following table.

   Supported directives and rules

Directive or rule	Use
`@source inline("…")`	Safelist class names that can't be discovered statically
`@layer utilities { … }`	Add or override utility classes
`@keyframes … { … }`	Define app-specific animations

   Declare overrides in a named layer so that their precedence doesn't depend only on stylesheet order:

   ```
   /* tailwind.app.css */
   @layer utilities, your-app;

   @layer your-app {
     .aiux-btn {
       border-radius: 50px;
     }
   }
   ```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Important:

Leave the following two things out of `tailwind.app.css`:

- Don't add `@import "tailwindcss"`. The build entry already imports Tailwind, and another import duplicates generated styles.
- Don't add an `@theme { … }` block. The app build removes the generated theme layer so that it can't replace the platform's customized tokens.

2. Register document-wide CSS with `head.app.css`

The `head.app.css` file isn't processed by Tailwind or PostCSS, and it's linked directly in the document `<head>`. Its content is preserved except for supported `public/...` URLs inside `@font-face` rules, which the build prefixes with the app basename.

Use this file for CSS that must be registered once outside component shadow roots. The primary use case is `@font-face`:

```
/* head.app.css */
@font-face {
  font-family: 'Inter';
  font-style: normal;
  font-weight: 400 700;
  font-display: swap;
  src: url(https://cdn.example.com/fonts/inter.woff2)
 format('woff2');
}
```

Inside an `@font-face` rule, a path beginning with `public/` is rewritten to `/aiux/<basename>/public/...`. Other URLs, URLs outside `@font-face`, and `@import` values are preserved as authored. For those references, use an `https:` URL, a `data:` URI, or an already-served root-relative path.

The `head.app.css` pipeline doesn't copy local assets, so files under `public/` must also be included by the app's static-content configuration.

Example: Verify

- Build and deploy the app. No additional manifest configuration is required.
- Inspect the document `<head>` and confirm that `head.app.css` is linked after the platform's base `aiux-theme-variables` style block.
- Inspect an app component's shadow root and confirm that the compiled app Tailwind sheet is present.
- For a custom font, confirm that the font file loads once.

What to do next

- Configure app-level theming — configure rebrandable colors, shape, and fonts.
- #unique_1236 — available tokens and their default values.

Configure theme-level theming

Configure brand colors, shape, theme-owned fonts, logos, favicons, and mobile color mappings in a `theme.js` module. Theme-level values can vary by tenant or user without a change to component code or an app rebuild.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin
Role required: admin

About this task
If a value must stay fixed for every deployment of the app, configure it at the app level instead. For instructions, see Configure app-level theming.

Procedure
1. Declare the theme in `theme.js`

   Create `theme.js` at the app root, alongside `aiux.json`, and export an object literal.

   The following example includes the theme's fields. Replace the sample attachment sys IDs and font paths with values from your instance and app.

```
// theme.js
export const AppTheme = {
  primary_color: '#032D42',
  accent_color: '#63DF4E',

  neutral_colors: {
    use_hex_value: true,
    hex_value: '#F8F7F4'
  },

  shape: {
    selector_radius: '0.75rem',
    field_radius: '0.5rem',
    boxes_radius: '1rem',
    border_width: '1px'
  },

  fonts: [
    {
      name: 'Demo Grotesk',
      faces: [
        {
          src: 'public/fonts/DemoGrotesk.woff2',
          format: 'woff2',
          weight: '100 900',
          style: 'normal',
          display: 'swap',
          unicode_range: 'U+0000-00FF',
          preload: true
        },
        {
          src:
 'https://cdn.example.com/fonts/DemoGrotesk-Italic.woff2',
          format: 'woff2',
          weight: '400',
          style: 'italic',
          display: 'swap',
          preload: false
        }
      ]
    }
  ],
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
  mobile_color_variables: {
    'text-primary': '--color-text-primary',
    'surface-primary': '--primary-600',
    'text-on-primary': '--matched-primary-content',
    'custom-background': '#ffffff'
  },

  light_logo: 'aaaaaaaa11111111bbbbbbbb22222222',
  dark_logo: 'cccccccc33333333dddddddd44444444',
  light_favicon: 'eeeeeeee55555555ffffffff66666666',
  dark_favicon: '11111111aaaaaaaa22222222bbbbbbbb'
};
```

For the values each field accepts and what it generates, see Theme field reference.

Important: Use a `theme.js` module that exports an object literal, not `theme.json`. The build parses this module to produce the app's `sys_aix_theme` record, and it doesn't read a `theme.json` file in a Lux app.

2. Connect the theme to the app

   Import the theme in `application.js` and export it as `appTheme`:

```
// application.js
import {AppTheme} from './theme.js';

export default {
  appTheme: AppTheme,
  applicationLayout: 'aiux-nav-layout'
};
```

   If the app has no lifecycle module, set the theme on the layout class instead:

```
// pages/layout.js
import {AIUXAppLayoutElement} from
 '@servicenow/aiux-components-core';
import {AppTheme} from '../theme.js';

export default class AppLayout extends AIUXAppLayoutElement {
  static appTheme = AppTheme;
}
```

   A theme provisioned on the instance takes precedence over the file-based app theme. For file-based configuration, `application.js` takes precedence over the layout static. If none is configured, Lux uses its default theme.

3. Register a theme-owned font

   Use `fonts` when a typeface belongs to the tenant's brand and should be replaceable without a change to app source. Declare each family with one or more faces:

```
fonts: [
  {
    name: 'Demo Grotesk',
    faces: [
      {
        src: 'https://cdn.example.com/fonts/DemoGrotesk.woff2',
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
        format: 'woff2',
        weight: '100 900',
        preload: true
      }
    ]
  }
]
```

The `fonts` field registers font faces but doesn't select a family for components.

For a font that ships with the app and is identical for every tenant, register it in `head.app.css` instead, as described in Configure app-level theming.

Example: Verify

- Build and deploy the app, then confirm that a primary-colored control uses the configured brand color.
- Switch between light and dark modes and verify that text, surfaces, logos, and controls keep sufficient contrast.
- If the theme doesn't appear, confirm that `theme.js` is exported as `appTheme` before checking individual fields.
- For a custom font, confirm that the face loaded and that any expected preload was emitted.
- For a mobile mapping, inspect the mobile theme response and confirm that the semantic name resolves to light and dark values.

Example: Examples

The `examples/custom-font-theme` app demonstrates a font supplied by the theme.

Theme field reference

The fields of a `theme.js` theme object set brand colors, neutral colors, shape, fonts, mobile color mappings, logos, and favicons. Each field's value determines the tokens and document resources that the theme generates.

For the procedure that uses these fields, see Configure theme-level theming.

Theme fields

The following table lists the fields of the theme object.

Field	Required	Value example	What it controls
`primary_color`	Yes	`'#032D42'`	Primary brand scale, primary actions, links, and focus states
`accent_color`	No	`'#63DF4E'`	Accent scale and secondary or AI-focused emphasis
`neutral_colors`	No	`'sand'` or `{use_hex_value: true, hex_value: '#F8F7F4'}`	Neutral scale used by surfaces, text, borders, and dark backgrounds

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Required	Value example	What it controls
`shape`	No	`{field_radius: '0.5rem', …}`	Selector, field, container, and border shape tokens
`fonts`	No	`[{name: 'Demo Grotesk', faces: […]}]`	Font families registered as document-wide `@font-face` rules
`mobile_color_variables`	No	`{'text-primary': '--color-text-primary'}`	Mobile semantic names resolved to light and dark hex values by the server
`light_logo`	No	A `db_image` sys ID	Navigation logo for light mode
`dark_logo`	No	A `db_image` sys ID	Navigation logo for dark mode
`light_favicon`	No	A `db_image` sys ID	Browser favicon for light mode
`dark_favicon`	No	A `db_image` sys ID	Browser favicon for dark mode

Brand colors

The `primary_color` field is the main brand input and should be a hex color. Lux expands it into an 11-stop scale from 50 through 950 for both light and dark modes. Semantic tokens then map buttons, links, focus rings, text, and surfaces to appropriate stops in that scale.

The `accent_color` field generates a separate 11-stop accent scale. Use it for secondary emphasis rather than as a replacement for the primary brand color. If `accent_color` is omitted, Lux uses its default accent color.

```
primary_color: '#032D42',
accent_color: '#63DF4E'
```

In components, use semantic tokens such as `--color-primary` instead of using `--primary-600` directly. Semantic tokens can select different scale stops when the color scheme changes.

Neutral colors

The neutral scale underpins most backgrounds, text colors, and borders. You can configure it in one of two ways.

Reference a shipped neutral swatch by name:

```
neutral_colors: 'sand'
```

The shipped swatches are `sand`, `granite`, `stone`, `zinc`, `dune`, and `slate`. Names are case-insensitive, and an unknown name fails the metadata build.

Alternatively, generate a custom neutral scale from one hex value:

```
neutral_colors: {
  use_hex_value: true,
  hex_value: '#F8F7F4'
}
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The object form creates an app-specific `sys_aix_color_swatch` record. The generated light and dark neutral scales feed tokens such as `--neutral-300`, `--color-text-secondary`, and `--color-border-subtle`. If no usable neutral configuration resolves at runtime, Lux generates the default neutral scale so that these tokens aren't left undefined.

Shape

The `shape` object maps directly to four component-level custom properties.

Field	CSS token	Applies to	Example values
`selector_radius`	`--radius-selector`	Checkboxes, radios, and toggles	`'0'`, `'0.25rem'`, `'9999px'`
`field_radius`	`--radius-field`	Inputs, buttons, selects, and text areas	`'0.5rem'`
`boxes_radius`	`--radius-box`	Cards, panels, and other containers	`'1rem'`
`border_width`	`--border`	Default component borders	`'0'`, `'1px'`, `'2px'`

Values are CSS lengths. The generated declarations apply under the active `[data-theme]` selector, so all components receive the same shape settings.

Logos and favicons

The `light_logo`, `dark_logo`, `light_favicon`, and `dark_favicon` fields are ServiceNow `db_image` reference sys IDs, not image URLs:

```
light_logo: 'aaaaaaaa11111111bbbbbbbb22222222',
dark_logo: 'cccccccc33333333dddddddd44444444',
light_favicon: 'eeeeeeee55555555ffffffff66666666',
dark_favicon: '11111111aaaaaaaa22222222bbbbbbbb'
```

The light and dark logo fields populate the theme's mode-specific logo value. The favicon fields produce mode-specific favicon URLs in the document head. Provide both variants when the same asset doesn't have sufficient contrast in both schemes.

Mobile color mappings

Native mobile clients don't run the browser's color-generation pipeline. The `mobile_color_variables` field maps names that the mobile client expects to design tokens or direct values, which the server resolves for light and dark modes:

```
mobile_color_variables: {
  'text-primary': '--color-text-primary',
  'surface-primary': '--primary-600',
  'text-on-primary': '--matched-primary-content',
  'custom-background': '#ffffff'
}
```

Each key is the semantic name returned to the mobile client. Each value can reference a generated palette token, a semantic color token, or a direct color value. Theme entries are merged over the platform's default mobile mappings, so you only need to declare values that differ from those defaults.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Font family fields

The `fonts` field is an array of font families, and each family contains one or more faces:

```
fonts: [
  {
    name: 'Demo Grotesk',
    faces: [
      {
        src: '/aiux/your-app/public/fonts/DemoGrotesk.woff2',
        format: 'woff2',
        weight: '100 900',
        style: 'normal',
        display: 'swap',
        unicode_range: 'U+0000-00FF',
        preload: true
      },
      {
        src:
 'https://cdn.example.com/fonts/DemoGrotesk-Italic.woff2',
        format: 'woff2',
        weight: '400',
        style: 'italic',
        display: 'swap'
      }
    ]
  }
]
```

The fields of each family are described in the following table.

Field	Required	Accepted values	Effect
`name`	Yes	A family name up to 64 letters, numbers, spaces, `.`, `_`, or `-`	Becomes the `font-family` value in generated `@font-face` rules
`faces`	Yes	An array of face descriptors	Registers weights, styles, ranges, or files belonging to the family

A theme can register up to eight usable families and twelve faces per family. Families without a valid name or at least one usable face are omitted.

Font face fields

The fields of each face descriptor are described in the following table.

Field	Required	Accepted values	Effect
`src`	Yes	An `https:` URL, an already-served root-relative path, or an app asset under `public/`	Identifies the font file
`format`	No	`'woff2'`, `'woff'`, `'ttf'`, or `'otf'`	Adds the CSS `format()` hint and determines the preload MIME type

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Required	Accepted values	Effect
`weight`	No	`'normal'`, `'bold'`, a number such as `'400'`, or a range such as `'100 900'`	Sets `font-weight`; ranges describe variable fonts
`style`	No	`'normal'`, `'italic'`, or `'oblique'`	Sets `font-style`
`display`	No	`'auto'`, `'block'`, `'swap'`, `'fallback'`, or `'optional'`	Sets `font-display`; defaults to `'swap'`
`unicode_range`	No	A CSS Unicode range such as `'U+0000-00FF'` or a comma-separated list	Limits the characters served by that face
`preload`	No	`true` or `false`	Requests an early `<link rel="preload" as="font">` for the face

Prefer `woff2` for modern browsers, because it normally provides the smallest transfer size. Declare separate faces for distinct static weights and styles, or one weight range for a variable font.

Font URLs and preloading

Font URLs are validated before CSS is generated. The following URL forms are dropped:

- `http:` URLs
- `data:` URLs
- Protocol-relative URLs, such as `//cdn.example.com/font.woff2`
- Unsupported relative paths, such as `./fonts/font.woff2`

For a font under the app's `public/` directory, author a path such as `public/fonts/DemoGrotesk.woff2`. The build rewrites it to `/aiux/<basename>/public/fonts/DemoGrotesk.woff2`. The file must also be included by the app's static-content configuration. An `https:` URL or an already-served root-relative path can be used without that rewrite.

At most four faces are preloaded per theme, in declaration order. A preloaded face must include a supported `format`. Without one, Lux can't emit a reliable MIME type and skips that preload. Reserve preloading for faces needed during first paint.

Font registration

The `fonts` field registers font faces but doesn't select a family for components. During theme generation, valid faces become one `theme-font-faces` style block in the document head. This block registers each family once for the page rather than duplicating `@font-face` rules in every shadow root.

Extend experiences

Use extensions to add pages, navigation, and layouts to an experience that another scoped application owns, without forking the host application or changing its code.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Extend existing ServiceNow experiences

Contribute a page to an experience owned by another scoped application, without changing the host application's repository. The contribution can be a net-new route or a replacement for one of the host's existing pages.

Before you begin

- Role required: admin
- The Now SDK is installed and authenticated against a target instance.
- Familiarity with scaffolding an application with `now-sdk init` (see Create your first Lux experience).
- The host application's `scope` and `basename` (its `aiux.json` values). If the host's manifest isn't visible - for example, a pre-existing experience whose scope wasn't derived through this convention - use the experience's sys_id directly instead. For that explicit form, see #unique_1242.

About this task

For the same task through the ServiceNow Lux Lab for VS Code instead of the CLI, see Create an extension application with Lux Lab.

Procedure

1. Scaffold the host application, if you're extending one you also own for this exercise (skip this step against a real, already-deployed host).

```
mkdir host-app && cd host-app
npx @servicenow/sdk@latest init --template javascript.aiux
```

`now-sdk init` scaffolds into the current directory, prompting for the application name, package name, and scope if not passed as flags. Install dependencies, then build and deploy:

```
pnpm install
pnpm build
pnpm run deploy
```

Confirm that `https://<instance>/aiux/<basename>/home` renders the starter page before continuing.

2. Scaffold the extending application, in a separate directory and terminal.

```
mkdir extending-app && cd extending-app
npx @servicenow/sdk@latest init --template
 javascript.aiux-extension
```

Beyond the application name, package name, and scope, `init` also prompts for the host's scope and route ID (its scope and basename from step 1). Install dependencies:

```
pnpm install
```

The extending application's own `aiux.json` and `pages/` are unrelated to what it extends - an extension is declared separately, under `src/extensions/`. A project must not include both. If a project contains any src/extensions/ pages, its own pages/ are still processed. However, they get silently wired into the host's experience instead of its own, even with basename set. Keep an extension project extension-only (pages under src/extensions/ only, no project-owned pages/) A mixed project does carry one trap of its own around per-host lifecycle modules - see #unique_1243.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

3. Add a net-new page, at a route the host doesn't already have, under the host's scope and basename.

```
src/extensions/x_example_host/host/pages/reports/page.js
```

```
import {html} from 'lit';
import {customElement} from 'lit/decorators.js';
import {AIUXElement} from '@servicenow/aiux-components-core';

@customElement('ext-reports-page')
export default class ExtReportsPage extends AIUXElement {
  render() {
    return html`<h1>Reports (contributed by
 extension-app)</h1>`;
  }
}
```

4. Add a dynamic-route page, if the net-new route takes a parameter.
A bracket directory works the same way it does in an owned application:

```
src/extensions/x_example_host/host/pages/sample/[sampleName]/pa
ge.js   # /sample/:sampleName
```

The built `ssr` path for a bracket directory is mangled to `sample/_sampleName_/page.js`. Decorators are now extracted from the real source path, so `@roles([...])` and `@order()` survive the build. Confirm `sys_aix_page.roles` on the installed record if you're gating a dynamic-route page and the installed SDK predates that fix.

5. Override an existing host page, at a route the host already owns.

```
src/extensions/x_example_host/host/pages/home/page.js
```

```
import {html} from 'lit';
import {customElement} from 'lit/decorators.js';
import {AIUXElement} from '@servicenow/aiux-components-core';

@customElement('ext-home-override')
export default class ExtHomeOverride extends AIUXElement {
  render() {
    return html`<h1>Home (overridden by extension-app)</h1>`;
  }
}
```

6. Build and deploy the extension.

```
pnpm build
pnpm run deploy
```

`pnpm deploy` (without `run`) is pnpm's own reserved subcommand and fails with ERR_PNPM_CANNOT_DEPLOY - always use `pnpm run deploy`. No redeployment or reconfiguration of the host is needed. The host's experience acquires the contributed page only after the corresponding records exist and reference it.

7. Verify.

On an instance, open the host application. `/aiux/host/reports` renders the net-new page, and `/aiux/host/home` renders the override rather than the host's own starter page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

`sys_aix_experience_page_rel`, filtered to the host experience, gains one membership row per contributed page. There is no `sys_aix_page_route_map` row for either, because extension routing doesn't use that table. The deploy auto-invalidates the host experience's cache, so the change is visible on the next request with no manual eviction step. For faster iteration, a dev server can render an extension straight from local source, with no deploy.

Important:

Don't verify a role-gated extension page as `admin`

A session with the `admin` role passes `gs.hasRole()` for any role name, including one that has no `sys_user_role` record at all. So `admin` always sees the role-gated override and never triggers the fallback to the host's own page. This is standard ServiceNow behavior, not specific to extensions. To observe the fallback, use a session that genuinely has no roles. Either log in as a no-role user, or impersonate one. To impersonate, use **avatar menu** > **Impersonate user** and pick a stock demo user with no roles. Navigate back to the route while impersonating, then stop impersonating to return to your own session.

Example: Two extensions targeting the same route

Scaffolding a second extending application that overrides the same host route is a supported scenario, though the tooling doesn't resolve it for you. For the ordering rules, see #unique_1243.

Create an extension application with Lux Lab

Lux Lab's VS Code extension scaffolds, builds, and deploys an extension application through a guided UI. An extension application adds pages to an existing ServiceNow experience; it doesn't create a standalone experience.

Before you begin

- Role required: admin
- Complete the Environment Check.
- Connect to the ServiceNow instance that contains the experience you want to extend.

About this task

As an alternative to the CLI walkthrough in Extend existing ServiceNow experiences, the guided UI keeps extension pages under the same path convention the CLI uses:

```
src/extensions/<parentScope>/<parentRouteId>/pages/<page-slug>/page.js
```

Procedure

1. Create the extension application

   a. Open the **Lux Lab** Activity Bar view, then open **Script Runner**.

   b. Select **Create LUX Project**, or run `Lux: Create LUX Project` from the VS Code Command Palette.

   c. Select the ServiceNow instance that contains the host experience, if it isn't listed, add it first.

   d. Select the parent folder, then enter the application name and project folder name.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

e. Select **Extend Existing App**.

f. Select the host experience to extend, then confirm the scope name or scope suffix.

g. If prompted, choose **Add to Current Workspace** or **Open in New Workspace**.
Wait for scaffolding and dependency installation to finish.

h. Lux Lab opens the generated extension project.

If the host experience can't be loaded, verify that the selected instance is active and that your credentials can read the available experiences. The host experience must have a valid scope and URL suffix before you can use it.

2. Create a new page in the extension application

Use this flow when the extension application needs a page that doesn't already exist in the host experience.

a. Open the extension project in VS Code.

b. Run `Lux: Create Page` from the Script Runner or the Command Palette.

c. Select **New Page**.

d. If the project extends more than one host application, select the host application for the new page.
Lux Lab selects it automatically if it extends only one.

e. Enter the page name or title.

f. Enter the page route.
The route is prefilled from the page name and must use lowercase letters, numbers, and hyphens. Lux Lab blocks duplicate routes within the selected host application.

g. Confirm the prompts.
Lux Lab creates and opens `src/extensions/<parentScope>/<parentRouteId>/pages/<page-slug>/page.js`.

h. Implement the page in the generated `page.js` file.

i. Preview the extension application and verify the page against the host experience.

3. Extend an existing host page

Use this flow when the extension application should add an override or extension for a page that already exists in the host experience.

a. Open the extension project in VS Code.

b. Run `Lux: Create Page` from the Script Runner or the Command Palette.

c. Select **Extend from Existing Page**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

d. If the project extends more than one host application, select the host application.

e. Select the host page from the list loaded from the active ServiceNow instance.
Use **Refresh** in the picker if the page was recently added or changed on the instance.

f. Pages already scaffolded in the extension project appear as already added and can't be selected again.

g. Lux Lab derives the extension page route from the selected host page and creates `src/extensions/<parentScope>/<parentRouteId>/pages/<page-slug>/page.js`.

h. Update the generated `page.js` while preserving the host page's route contract.

i. Preview the extension application and compare the result with the original host page.

Create only one extension for a given host page unless you intentionally need multiple extension projects. Verify the host experience and page before deploying.

4. Add an extension page to host navigation

Creating an extension page doesn't automatically add it to the host navigation. Configure the host-level lifecycle module at:

```
src/extensions/<parentScope>/<parentRouteId>/application.js
```

For example, in an extension project that targets the `example-app` host, the file may be `src/extensions/x_aix_example_a/example-app/application.js`.

a. Add a page to L1 navigation

Use `addNavItem` in `application.js`, setting `action.path` to the route created for the extension page:

```
addNavItem({
  icon: 'document',
  title: i18n.getMessage('Reports'),
  action: {type: 'navigate', path: '/reports'},
});
```

b. Add a page to L3 navigation

Use `setL3Nav` to set the nested (L3) tabs for an existing navigation item; the first argument is the parent navigation path:

```
setL3Nav('/reports', [
  {label: 'Overview', path: '/reports/overview'},
  {label: 'Details', path: '/reports/details'},
]);
```

To add tabs without replacing existing tabs, use `appendL3Nav` instead. Import the navigation helper you use from `@servicenow/aiux-components-nav`, then save `application.js`.

c. Wire up and verify the navigation item

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

i. Use the page route created in the new-page flow, or preserve the host route when extending an existing page.

ii. Add the corresponding L1 or L3 navigation configuration in the host-level `application.js`.

iii. Preview the extension application and verify that the page opens from the expected navigation item.

iv. Build and deploy the extension application.

v. Run `Lux: Refresh Instance Explorer` and verify the navigation on the target instance.

5. Build and deploy the extension application

   a. Save the generated page and all related files.

   b. Use the project's preview action in Script Runner to test the extension page.

   c. Run `Lux: Build` from Script Runner and resolve any build errors.

   d. Confirm that the correct ServiceNow instance is active.

   e. Run `Lux: Deploy` to deploy to the active ServiceNow instance.

   f. Use `Lux: Show Command Output` to monitor the deployment.

   g. Run `Lux: Refresh Instance Explorer` and verify the extension application and page on the target instance.

   Navigation visibility depends on the host experience configuration, user access, route metadata, and the active ServiceNow instance.

What to do next

For the decorators, ordering, and record-level mechanics behind what this flow generates, see #unique_1242. The guided flow does not expose decorators, layout contribution, or per-host lifecycle modules; use the CLI path in Extend existing ServiceNow experiences for those.

Test and deploy a Lux app

A Lux app moves from your local machine to a ServiceNow instance through preview, testing, debugging, validation, and deployment.

Topics in this section start with the ways to preview an experience and with testing an app locally with Vitest. They then cover debugging a running page with Lux DevTools and diagnosing slow pages. They close with the validation checks that run between a local build and a deployed instance, and with the deployment pipeline.

Ways to preview a Lux experience

Four mechanisms show you an app before it ships, from the Lux Lab Preview panel to a local production build. No shareable preview link is available, and no server-side staging view exists separate from your local dev server or a deployed instance.

The four are the Lux Lab Preview panel, a local dev server with hot reload, a local production build, and server-rendered output on a deployed instance.

Preview in Lux Lab

Lux Lab has a dedicated **Preview** panel: a live, embedded browser view of your running app, opened as an editor tab. The same local dev server described later in this topic backs the panel, proxied to the ServiceNow instance you are logged in to.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The methods for opening the panel are described in the following table.

Preview panel open methods

Method	What happens
Preview in the title bar	Starts the dev server if it isn't running, then opens the preview tab
Run in the title bar	Starts the dev server; the preview tab opens when the server is ready
Auto-open	The preview opens by itself when the dev server finishes starting

The **Run** button doubles as the stop control. It displays **Run** when the server is stopped, shows a transition indicator while it starts, and is set to **Stop** after the server is running.

The preview panel carries a floating toolbar. Its controls are described in the following table.

Preview toolbar controls

Control	What it does
URL bar	Shows the current preview URL, and is editable. Enter a path or URL and press Enter to navigate
Refresh	Reloads the preview page
Open in browser	Opens the current URL in your default system browser
Device presets	Switches between Mobile, Tablet, and Desktop viewport widths
Screenshot	Captures the preview and sends the image to the Agent Chat panel

Beyond the three presets, you can drag the resize handles at the edges of the preview frame for a custom viewport width. Dragging the tab's edge resizes the preview against the code editor.

Saving a file refreshes the preview automatically through hot module replacement (HMR); the toolbar's Refresh button is for a manual reload. Changes to configuration files such as `aiux.json` are the exception, those need a dev-server restart to take effect.

Logs from the dev server, startup progress, HMR activity, runtime errors, stream into the **Console** tab of the bottom panel. Linter and type errors appear in **Problems**, where selecting an error jumps to the offending line; build-pipeline failures appear in **Build**.

For the Lux Lab app that hosts this panel, see Lux Lab.

Preview locally with the dev server

`pnpm dev` starts the full request pipeline on your machine with hot reload: proxy, version router, server-side rendering (SSR) service, and Core UI. You see real server-rendered output, not a static mock.

```
cd my-app
pnpm dev
```

Open `http://localhost/aiux/<basename>/<landing>` in your browser. Editing a file under `pages/` or `components/` recompiles and reloads the app automatically, and the build writes development artifacts to `.dev` directories.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The dev server starts the same layered pipeline as production. The services and their ports are provided in the following table.

Dev server services and ports

Service	Port	Role
HTTP proxy	80	Entry point for all requests
Version router	3000	Routes to versioned SSR services
SSR service	3001+	Renders your pages server-side
Core-UI service	4000	Serves list, record, and dashboard pages
Core static server	4999	Serves shared widget bundles

To run more than one app's dev server at once, set `AIUX_INSTANCE` to a number from 1 to 9 before starting it. Setting it shifts the whole port range so the stacks don't collide. `AIUX_INSTANCE=1` moves the proxy from port 80 to 31000 and its services into the 31001+ range, `AIUX_INSTANCE=2` uses 32000, and so on.

```
AIUX_INSTANCE=1 pnpm dev
```

If port 80 requires elevated permissions or is already taken, set `PROXY_PORT` in your `.env` file instead, and access your app at `http://localhost:<PROXY_PORT>/aiux/...`.

To diagnose a local setup problem, run the diagnostic tool before digging further:

```
aiux doctor --check
```

It diagnoses and reports common configuration problems.

For the automated checks that run against the same code, see Test an app locally.

Preview a production build locally

`pnpm dev` runs an unoptimized dev build. To see what actually ships, the built `dist/vN/` bundles, no hot reload, build and serve it locally before you deploy:

```
pnpm build
pnpm serve
```

Open the same `http://localhost/aiux/<basename>/<landing>` URL. This is the closest you get to your production artifact without touching a ServiceNow instance.

Verify server-rendered output

Your app can be deployed and running either locally or on an instance. In both cases, the rendered HTML your loader produced appears on the document request's **Preview** tab in **DevTools** > **Network**. A blank Preview with a working browser page means your loader is generating an error server-side, check the service logs at `https://<instance>/aiux/<basename>/logs`.

Do not confuse this DevTools **Preview** tab with the Lux Lab Preview panel or the `pnpm serve` command described earlier: three unrelated mechanisms that share the word "preview." The tab inspects one request's rendered HTML; the panel and `pnpm serve` each run your whole app.

For deeper inspection of a running page, widget properties, re-render causes, network calls, see #unique_1247. That topic covers Lux DevTools, a different tool from the browser's built-in DevTools panel described earlier in this topic.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Confirming that server-rendered output arrives is also an item on the ship checklist in Deploy to an instance.

Test an app locally

Testing a page locally with Vitest catches loader and rendering failures before you deploy. Separate unit tests cover your data-fetching logic, and separate render tests cover the HTML your component produces.

Before you begin

- A page with a `static async loader(ctx)` method and a `render()` method. To learn how a page declares both, see Create a page.
- Vitest configured, which is present by default in an app scaffolded with `now-sdk init`.
- Role required: admin

About this task

Lux uses Vitest with a two-tier strategy: loader unit tests verify your data-fetching logic, and server-side rendering (SSR) render tests verify the HTML your component produces. They live in separate files; mixing them causes `vi.mock()` conflicts, because loader tests mock Lit entirely while render tests require real Lit and a DOM shim. To access the test runner's reference material, see the Vitest documentation.

Procedure

1. Create the test files.

```
my-app/
├── pages/
│   └── risk/page.js
└── test/
    ├── risk-page-loader.test.js   ← Tier 1: loader tests
    └── risk-page.test.js          ← Tier 2: SSR render tests
```

Test files live in `test/` directories alongside `src/`.

2. Write loader unit tests.

Mock Lit and `fetch`, then assert on your loader's behavior, not on the mock.

```
import {describe, it, expect, vi, beforeEach, afterEach} from
 'vitest';

vi.mock('lit', () => ({isServer: true}));
vi.mock('lit/decorators.js', () => ({customElement: () => cls
 => cls}));

const mockGetHeaders = vi.fn(() => ({'Content-Type':
 'application/json'}));
vi.mock('@servicenow/aiux-components-core', () => ({
  AIUXElement: class {},
  getHeaders: (...args) => mockGetHeaders(...args)
}));

const {default: MyPage} = await
 import('../pages/risk/page.js');

describe('MyPage.loader', () => {
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
  it('calls the correct API endpoint', async () => {
    globalThis.fetch = vi.fn(async () => ({json: async () =>
 ({result: []})}));
    await MyPage.loader(mockCtx);
    expect(globalThis.fetch).toHaveBeenCalledTimes(1);
  });
});
```

Assert on the fetch URL and method, that `getHeaders(ctx)` was forwarded, the return value's shape, error handling, and edge cases such as missing fields and empty arrays.

3. Copy the SSR test helper.
   SSR render tests require real Lit and a DOM shim. Copy `test/ssr-utils.js` — about 50 lines — from the framework monorepo into your app's `test/` directory.
4. Write SSR render tests, and render against mock loader data:

```
import {html} from 'lit';
import {setupLitSSR, renderToString, clean} from
 './helpers/ssr-utils.js';

let render;
beforeAll(async () => {
  ({render} = await setupLitSSR());
  await import('../pages/risk/page.js');
});

it('renders the risk list', async () => {
  globalThis.__COMPONENT_DATA__ = {risks: [{number:
 'RSK0001'}], error: null};
  const result = render(html`<my-risk-page></my-risk-page>`);
  const output = clean(await renderToString(result));
  expect(output).toContain('RSK0001');
});
```

During real SSR, the framework sets `globalThis.__COMPONENT_DATA__` with your loader's return value before rendering, and `AIUXElement` reads it in its constructor. In tests, set it directly.

5. Run the tests.

   Run everything, one package, one file, or one test by name.

```
pnpm test                                                    #
 all tests
pnpm exec vitest run test/risk-page-loader.test.js          #
 one file
pnpm exec vitest run -t "calls the correct API endpoint"    #
 by test name
pnpm exec vitest watch test/                                #
 watch mode
pnpm test:coverage                                          #
 coverage report
```

   To run a single package's tests, change into its directory first and run `pnpm test` there.

6. Run lint and type-check before you build.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
pnpm lint
pnpm typecheck
```

Example: Other test types

The loader and SSR render tests described earlier are page-level tests running in Node. Two other test types support client-side verification. All three are provided in the following table.

Test types

Type	File pattern	Command	Runs in
Unit	`*.test.js`	`pnpm test`	Node environment
Browser-mode unit	`*.browser.js`	`pnpm test:browser`	Vitest browser mode
Smoke	`*.smoke.js`	`pnpm test:smoke`	Real Chromium, driven by Playwright — verifies a widget mounts and hydrates

Add `pnpm test:smoke:headed` to run smoke tests with a visible browser window.

Example: Don't test the mock

A common mistake is mocking a call to return X, then asserting the result is X. That verifies the mock, not your code. Mock the dependency, which is the network call, and assert on the behavior, which is what your loader does with the response.

What to do next

After your tests pass, move on to Deploy to an instance, `pnpm test` is the first item on that topic's ship checklist. For diagnosing a failure you can't explain from a test alone, see #unique_1247. For the build-time and CI checks that run beyond what you can trigger locally, see #unique_1249.

Deploy to an instance

Getting a change onto a ServiceNow instance is a three-step pipeline: build the app, deploy its records, then verify the service and the app. The same pipeline runs from the command line, from Lux Lab, or ServiceNow Lux Lab for VS Code.

Before you begin

- Role required: admin, on the target instance.
- `now-sdk` installed.
- Tests and lint passing. For the local checks, see Test an app locally.

About this task

```
Source code  →  pnpm build  →  pnpm deploy  →  ServiceNow
 instance
                  (aiux build)    (now-sdk install)
```

Procedure

1. Register credentials for your target instance, if not already registered.

```
npx now-sdk auth --add local --type basic
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The command prompts for the host, user name, and password. Confirm it was saved.

```
npx now-sdk auth --list
```

This writes `~/.snc/credentials`, which `now-sdk` reads for the instance URL and authentication on every subsequent command. The SDK stores credentials in the OS keychain under the alias you choose.

2. Set `scope` and `scopeId` in `now.config.json`.

   Without both, the build produces browser bundles but no installable records.

```
{
  "scope": "x_myco_my_app",
  "scopeId": "<32-char-scope-sys-id>",
  "name": "My App",
  "appOutputDir": "dist-metadata/app"
}
```

   `scope` is the application's scope name on the instance; `scopeId` is its sys_id.

3. Build the app.

   This emits the metadata and the deployable zip under `target/`.

```
pnpm build
```

   The build compiles your pages and components into versioned bundles under `dist/vN/`, generates a route manifest, and produces `dist-metadata/`, the ServiceNow record XML that gets installed. Don't commit `dist/`, `dist-metadata/`, or `target/`; apps scaffolded with `now-sdk init` gitignore all three.

Build outputs

Output	Contents
`dist/vN/`	Per-page browser and SSR bundles, a `manifest.json` mapping routes to bundles, shared chunks, and pre-compressed `.br` and `.gz` assets
`dist-metadata/app/`	ServiceNow record XML, one scope record plus the `update/` records for widgets, pages, layouts, and dependencies, and `sys_aix_*` metadata
`target/<app>_<version>.zip`	The deployable scoped-app archive that the ServiceNow SDK installs onto an instance

   To clear all caches and force a full rebuild, use `pnpm build:force`; add `-v` or `--verbose` for detailed output.

4. Deploy.

   Note: `now-sdk init` doesn't generate a `deploy` script by default, add one to `package.json`: `"deploy": "now-sdk install"`.

```
npx now-sdk install --auth local
```

   In most apps this is also wired to `pnpm deploy`. The SDK reads `dist-metadata/`, REST-pushes the records to the instance, and prints a rollback URL: save it, because visiting it

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

undoes the install. The first deploy creates the app's `sys_scope` record; later deploys update existing records in place, so re-running it is safe.

To re-run a deploy from a scaffolded app's own scripts, `pnpm run deploy:reinstall` wraps `now-sdk install --reinstall`.

> ⚠ **Warning:** `--reinstall` removes any application metadata on the instance that isn't present in your local build, including metadata other developers added. Run `now-sdk transform --auth <alias>` first to sync down what's already there.

5. Verify the service.

   If `/aiux/health` doesn't answer, nothing you deployed will render and the problem is the instance, not your code:

   ```
   curl https://<instance>/aiux/health
   ```

   A minimal `{"status": "UP", "service": "aiux-service"}` is the healthy answer for an ordinary caller.

6. Verify the app.

   Open `https://<instance>/aiux/<basename>/<landing>` and confirm the app loads. Check server-side rendering (SSR) output through **DevTools** > **Network** > **Preview**, and check the real-time log viewer at `https://<instance>/aiux/<basename>/logs` for server-side loader or SSR failures.

   For the SSR check in detail, see Ways to preview a Lux experience. For the `/aiux/ready` route and the other checks that run before and during a deploy, see #unique_1249.

Example: What gets installed where

Source	ServiceNow table	Record
`aiux.json`	`sys_aix_experience`	One record (the app itself)
`pages/<route>/page.js`	`sys_aix_page` + `sys_aix_widget`	One page record and one widget record per page
`pages/layout.js`	`sys_aix_layout`	One record (the app's layout)
`widgets/<name>/index.js`	`sys_aix_widget`	One record per discoverable widget
Page dependencies and shared widgets	`sys_aix_dependency`	One record per dependency
`src/fluent/tables/*.now.ts`	`sys_db_object` + columns	Tables and dictionary entries
`src/fluent/security/*.now.ts`	`sys_user_role` + ACLs	Roles and access controls
Static assets	`sys_attachment`	Bundled into the scope

A real app produces hundreds of these records; a large app can emit well over a thousand, that volume is normal.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Lux Reference

The Lux reference topics provide the exact names, keys, options, and values that the other chapters explain, from the `now-sdk` command-line interface to the design tokens.

Topics in this chapter cover the `now-sdk` command-line interface and the configuration schemas behind project files, configuration decorators, and configuration tables. They also cover the design tokens, grouped by family, and the `theme.js` file where an application declares its brand.

ServiceNow SDK

The `now-sdk` command-line interface manages changes between a local application and its version on an instance. ServiceNow Fluent defines that application's metadata as TypeScript source code.

For the most up-to-date reference information, see the ServiceNow SDK reference.

theme.js configuration reference

A Lux application declares its brand in one file at the project root, and the framework generates every color, radius, and font token from it. Restyling a Lux application goes through that file rather than through component CSS.

Important: Two files carry the name `theme.js`, and this page is about your application's own root `theme.js`, which holds four keys of brand input. The framework's internal token-definitions module, `libraries/theme-generation/src/algorithm/theme.js`, is documented in #unique_1236. If you are looking for `ROOT_TOKENS`, look there.

Scaffolded theme file

A scaffolded application ships with the following file.

```
// theme.js
export const AppTheme = {
  primary_color: '#032D42',
  neutral_colors: {use_hex_value: true, hex_value: '#F8F7F4'},
  shape: {
    selector_radius: '0.75rem',
    field_radius: '0.5rem',
    boxes_radius: '1rem',
    border_width: '1px'
  }
};
```

That is the whole file. The generated palette propagates to every component in the application.

theme.js Fields

Field	Type	What it controls
`primary_color`	Hex string	Brand color. Drives the `--primary-*` scale, buttons, links, and focus rings.
`accent_color`	Hex string	Secondary calls to action, through the `--accent-*` scale.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Type	What it controls
`neutral_colo`	`{use_hex_value: true, hex_value: '#...'}`	Base for the `--neutral-*` scale, which most surface, text, and border tokens resolve through.
`shape`	Object	Four radius and width keys: `selector_radius` for check boxes, radios, and toggles, `field_radius` for inputs and buttons, `boxes_radius` for cards and panels, and `border_width`.
`fonts`	Array	Font families to register as `@font-face`.
`light_logo` / `dark_logo`	Attachment reference	Per-scheme logo shown in the nav header.
`light_favicon_u` / `dark_favicon_url`	URL	Per-scheme favicon.
`token_overrides`	Object	Override for individual token values.

Only `primary_color` is required. Every other field falls back to a Lux default.

The four `shape` keys map onto the tokens documented in .

Applying the theme

Export the theme from your application's lifecycle module. Nothing else is required.

```
// application.js
import {AppTheme} from './theme.js';

export default {
  appTheme: AppTheme,
  applicationLayout: 'aiux-nav-layout'
};
```

When your application has no lifecycle module, set the theme as a static on your layout class instead.

```
// pages/layout.js
import {AIUXAppLayoutElement} from
 '@servicenow/aiux-components-core';
import {AppTheme} from '../theme.js';

export default class AppLayout extends AIUXAppLayoutElement {
  static appTheme = AppTheme;
}
```

Precedence

The following table lists the four theme sources, highest precedence first.

Theme source precedence

Source	Notes
A theme provisioned on the instance	Wins over everything in code, so an admin can rebrand the same application without a code change

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Theme source precedence (continued)

Source	Notes
`application.js`'s `appTheme` export	Wins over the layout static
`static appTheme` on an `AIUXAppLayoutElement` subclass	Fallback when there is no lifecycle module
The framework default theme	Applies when none of the other sources is present

Generated color scales

`primary_color`, `accent_color`, and `neutral_colors` are base colors rather than tokens. Each is expanded into an 11-stop scale, `50` through `950`, generated twice, once for light mode and once for dark, so contrast holds in both. The semantic tokens are defined in terms of those stops.

That expansion is why one hex value is enough to rebrand an application, and why component styles should use a semantic token rather than a raw stop. See .

token_overrides

`token_overrides` sets any design token without a change to component code. Every built-in token is defined with an override hook already in place, so a key matching a known token replaces that token's value everywhere it is consumed. A key matching nothing is emitted as a new custom property you can `var()` yourself.

Keys at the top level apply everywhere. Nest them under `light`, `dark`, `comfy`, or `cozy` to scope an override to one color scheme or density mode.

```
export const AppTheme = {
  primary_color: '#032D42',
  token_overrides: {
    '--radius-lg': '1.5rem',        // known token: every card
 gets rounder
    '--font-sans': '"Demo Grotesk"',
    '--brand-glow': '0 0 12px #63DF4E', // unknown key: new
 property, use via var()

    light: {'--color-primary': '#032D42'},
    dark: {'--color-primary': '#63DF4E'},
    comfy: {'--space-unit': '0.4rem'},
    cozy: {'--space-unit': '0.1rem'}
  }
};
```

Key to selector

token_overrides keys and the selectors they emit

Source key	Selector emitted
Root-level keys	`:root`
`light`	`[data-theme="light"]`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

token_overrides keys and the selectors they emit (continued)

Source key	Selector emitted
`dark`	`[data-theme="dark"]`
`comfy`	`[data-density="comfy"]`
`cozy`	`[data-density="cozy"]`

A theme with no `token_overrides`, or with an empty object, emits no override block.

Key resolution

`generateTokenOverrideBlocks()` checks each key against ROOT_TOKENS, DATAVIS_ROOT_TOKENS, LIGHT_THEME_TOKENS, DARK_THEME_TOKENS, DENSITY_COMFY, and DENSITY_COZY. A match is renamed to `--theme-level-<name>`, which feeds the `var(--theme-level-<name>, <default>)` indirection already baked into that token. A non-match passes through unprefixed.

Application CSS precedence

Tokens declared in your application's `tailwind.app.css` sit above the theme in the cascade, so a value pinned there isn't affected by whatever a theme sets. Use `token_overrides` for what a rebrand should be able to change, and `tailwind.app.css` for what it should not.

Storage and request flow

The field is stored as a JSON blob on the `sys_aix_theme` record. It is an empty string, rather than `null` or `undefined`, when a theme declares no overrides.

A theme's `theme.js`, or an admin-authored theme record, declares `token_overrides`. `collectThemeRecords()` stringifies it into the `sys_aix_theme.token_overrides` column at build time, and `theme-reader.js` parses it back on every request. `getAppThemeCSS()` calls `generateThemeCSS()`, which includes the override block in the server-rendered `<style id="theme-token-overrides">` tag. On a client-side theme change, `applyTheme()` regenerates and swaps the same block without a full reload.

The mechanism is implemented, with the font route noted in the preceding note.

Switching light and dark at runtime

Theme mode lives on the `data-theme` attribute. The framework exposes a service for flipping it, and the user's choice persists across reloads.

```
import {
  setThemeMode,
  getThemeMode,
  themeContext
} from '@servicenow/aiux-components-core';

_toggleTheme() {
  setThemeMode(getThemeMode() === 'dark' ? 'light' : 'dark');
}
```

Most applications don't need to write that code, because the nav layout ships a theme toggle that you turn on with `features.themeToggle`.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Reacting to the theme mode

Consume `themeContext` and reflect the value onto your own host as `theme-variant`.

```
static contexts = [themeContext];

willUpdate() {
  this.setAttribute('theme-variant', themeContext.get()?.mode ??
 'light');
}

static styles = css`
  :host([theme-variant='dark']) .panel {
    box-shadow: none;
  }
`;
```

Use `theme-variant` on the host rather than `data-theme`. The generated theme CSS declares `[data-theme]` rules at the document level, and those outrank a `:host` rule for the same token. Reflecting your own attribute keeps the override inside your shadow root, where it applies.

Lux Lab

Lux Lab is the desktop app for building Lux applications on the ServiceNow AI Platform. It provides Lux aware tooling for scaffolding experiences, pages, and widgets.

Get started

The Lux Lab documentation is organized into the following sections.

Section	What it covers
Exploring Lux Lab	Learn about Lux Lab.
Configuring Lux Lab	Install Lux Lab and connect it to your instance.
Using Lux Lab	Create and extend application experiences, pages, and widgets.

Panels and workflows

Lux Lab includes the following panels and workflows for building a Lux application:

- Project file tree
- Instance connection
- Code editor
- AI agent chat harness
- Source control
- Terminal and build output
- One-click deployment to a ServiceNow instance

ServiceNow developer resources

- ServiceNow Developer Site
- ServiceNow Community

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- ServiceNow Developer Store
- Personal Developer Instance (PDI)
- Now SDK documentation
- Now SDK community discussions

Exploring Lux Lab

Lux Lab provides Lux aware intelligence and built-in commands to help you create and extend Lux experiences. You build those experiences from pages and widgets tailored to fit your needs.

Lux Lab capabilities

Lux Lab enables you to create and extend Lux experiences. With built-in Lux aware intelligence, you can run commands from the title bar and the integrated terminal. You can also preview entire applications and check which files are relevant to your work from inside the Project Explorer. Create experiences, pages, and widgets that support tailored functionality in your Lux applications.

Lux Lab workflow

1. You create a new experience or extend an existing experience.
2. You create or extend pages and widgets, either manually or with agentic development tools.
3. You preview your changes from within Lux Lab.
4. You deploy your changes.

Lux Lab benefits

Each of the following benefits applies to developers.

Benefit	Feature
Run build and deploy commands from the title bar rather than the terminal.	**Run**, **Preview**, and **Deploy** buttons, and the **Deploy** drop-down list's **Project scripts** list
Preview your application in real time without leaving Lux Lab.	Preview panel
Browse instance data as a reference while building.	Instance Explorer
Develop tailored experiences with AI tools.	Agent Chat Harness (ACP)

Next steps

To get oriented in the app and to learn more about configuring and using Lux Lab, see the related links.

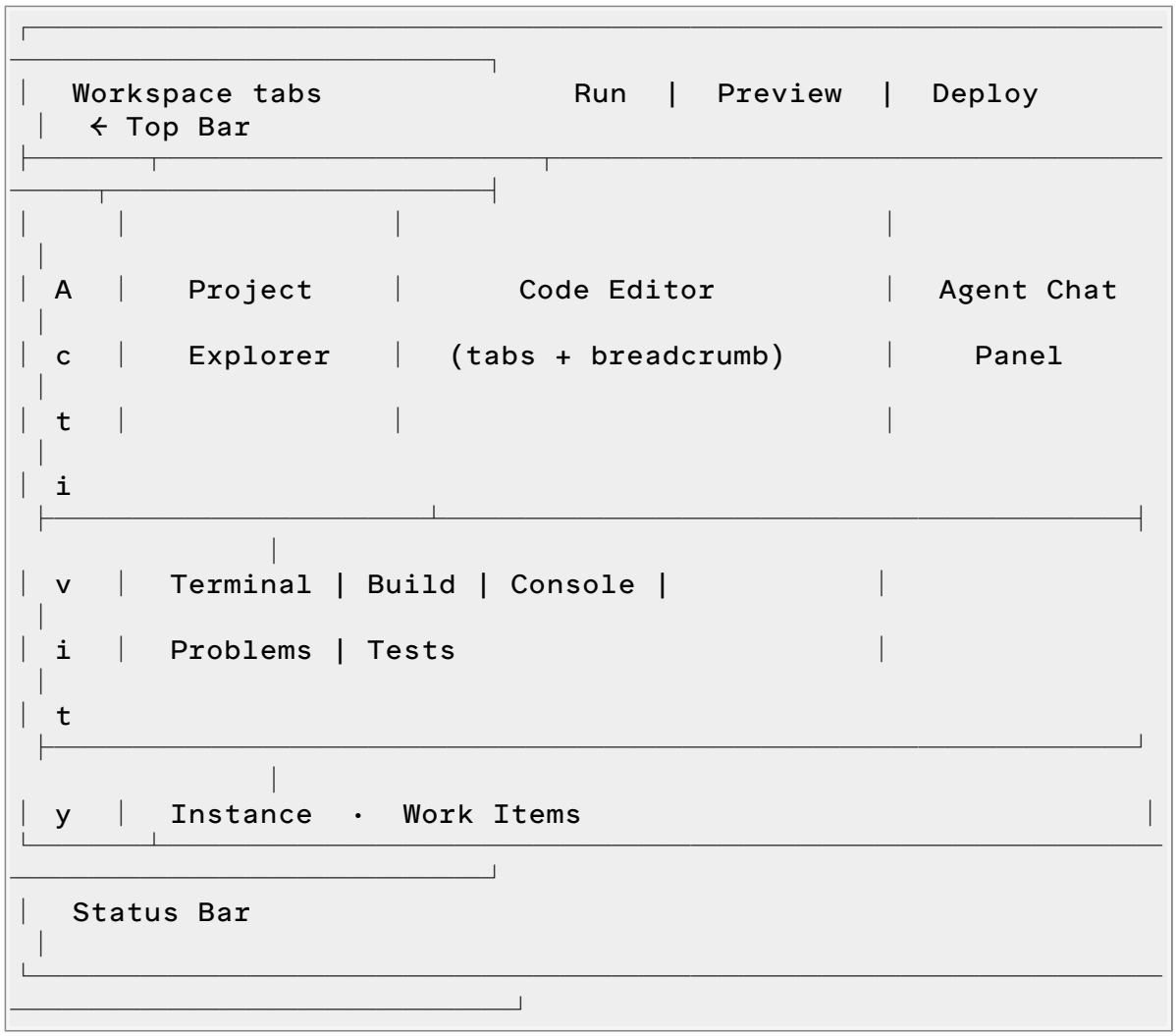
Lux Lab interface

The Lux Lab interface divides the app into regions for project files, code editing, agent chat, panel output, and instance status.

Interface regions

When you open a project, the Lux Lab interface is composed of the following primary regions:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



The following table describes the purpose of each region.

Region	Purpose
Top Bar	Workspace tabs on the left, and the **Run**, **Preview**, and **Deploy** action buttons on the right
Activity Bar	Narrow icon strip on the far left that switches between Explorer, Source Control, Search, and other views
Project Explorer	File tree for your local project, shown when the Explorer icon is active
Code Editor	Main editing area with tabs and a breadcrumb trail, with support for split views
Agent Chat Panel	AI assistant panel for conversational agent interactions
Bottom Panel	Tabbed panel with Terminal, Build, Console, Problems, and Tests
Instance and Work Items	Area pinned at the bottom of the left sidebar that shows the connected instance and current work items
Status Bar	Bottom strip with the branch name, sync status, IntelliSense, and other indicators

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Instance explorer

The Instance Explorer is the Lux Lab connection point to your ServiceNow instance. Use it to connect to an instance, browse its tables for quick data lookups, and switch between instances without leaving the app.

Instance switcher

The active instance appears in the top header bar of Lux Lab. Selecting the instance name opens the instance switcher, where you can do the following:

- See all connected instances
- Switch the active instance in one step
- Add a new instance connection

Instance Explorer panel

To open the dedicated panel, select the **Instance Explorer** icon in the Activity Bar. In the panel, you can browse ServiceNow tables and perform quick data lookups in the following ways:

- Expand a table to see its records.
- Search for specific tables or records using the search bar at the top of the panel.
- Use the data as a reference while building your Lux application.

The panel provides lightweight, read-only data lookup during development. It does not function as a full record editor.

Note: To import an existing application into a new project, use the **Import** option in the New Project dialog. See Create an experience.

Instance connections

For the full connection procedure, see Connect Lux Lab to an instance.

To switch the active instance, select the instance name in the top header bar, then select the instance you want to make active. Lux Lab uses the active instance for deployments, previews, and table browsing.

Instance status indicators

The following table describes what each status indicator means.

Indicator	Meaning
Green dot	Connected and responsive
Yellow dot	Connected but slow or degraded
Red dot	Connection failed or timed out
Gray dot	Disconnected

Project explorer

The Project Explorer is the primary file management panel in Lux Lab. It provides a hierarchical view of your project's files and folders, and is the starting point for navigating, creating, and organizing your work.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Panel location

The Project Explorer occupies the left sidebar when the **Explorer** icon, the top icon in the Activity Bar, is active. It shows the full directory tree rooted at your current project folder.

Opening and switching projects

You open and switch projects from the title bar drop-down list. Selecting the project name in the title bar reveals the project picker, which has the options in the following table.

Option	Description
Recent	Recently opened projects and workspaces. Select any project to switch to it.
Open Folder...	Option to open an existing project folder from disk
New Project	Option to create a Lux project from a template
New Workspace	Option to create a multi-root workspace

For the steps to open an existing project folder, see Open a project folder.

Creating files and folders

You create files and folders directly in the tree. When you hover over any folder, Lux Lab reveals its action icons, including **New File**.

For the steps to add a file, see Create a file. For the steps to add a subfolder, see Create a folder.

File tree navigation

The following table lists the keys for moving through the file tree.

Action	Method
Expand folder or open file	→ arrow or select the chevron
Collapse folder	← arrow
Open file or toggle folder	Enter
Quick peek preview	Space (press Esc to close)
Move focus up or down	↑ / ↓
Delete focused item	Del
Reveal active file in tree	Cmd+Shift+E
Jump to a file by name	Cmd+P (Quick Open)

Explorer keyboard shortcuts

The following table lists the Explorer keyboard shortcuts for macOS and Windows.

Action	macOS	Windows
New file in folder	Cmd+N	Ctrl+N
New folder	Cmd+Shift+N	Ctrl+Shift+N

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Action	macOS	Windows
Rename	F2	F2
Duplicate file	Cmd+D	Ctrl+D
Select all	Cmd+A	Ctrl+A
Copy absolute path	Cmd+Shift+C	Ctrl+Shift+C
Copy relative path	Cmd+Shift+Option+C	Ctrl+Shift+Alt+C
Reveal active file in tree	Cmd+Shift+E	Ctrl+Shift+E
Focus explorer search	Cmd+Shift+F	Ctrl+Shift+F
Toggle dotfiles	Cmd+Shift+.	Ctrl+Shift+.
Undo last file operation	Cmd+Z	Ctrl+Z

Explorer panel context menu

Right-clicking the Explorer panel background, rather than a file, shows the panel-level actions in the following table.

Action	Description
Add Folder to Workspace	Adds another project folder to the current workspace
Expand All	Expands all folders in the tree
Collapse All	Collapses all folders in the tree
Show Dotfiles	Toggles visibility of dotfiles such as `.env` and `.gitignore`
Open Focused Root in Terminal	Opens a terminal session at the project root
Reveal Focused Root in Finder	Opens the project root in Finder (macOS)

File and folder context menu

Right-clicking a file or folder reveals the actions in the following table.

Action	Description
New File	Creates a new file inside the selected folder
New Folder	Creates a new subfolder
Add Page...	Scaffolds a new Lux page inside the project
Add Extension Target...	Adds a new extension target to the project
Add Glide View Page...	Scaffolds a new Glide View page
Copy	Copies the file or folder
Cut	Cuts the file or folder
Duplicate	Creates a copy of the file in the same location
Copy Path	Copies the absolute path to the clipboard
Copy Relative Path	Copies the path relative to the project root

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Action	Description
Reveal in Finder	Opens the file location in Finder (macOS)
Open in Terminal	Opens a terminal session rooted at this folder
Rename	Renames the file or folder inline
Delete	Deletes the file or folder

Note: **Add Page...**, **Add Extension Target...**, and **Add Glide View Page...** are unique to Lux Lab and scaffold the corresponding Lux application artifacts directly into your project.

Inline run and preview icons

For files that support it, Lux Lab shows the action icons in the following table when you hover over the file in the Explorer tree.

File type	Icon	Action
Page file, for example `page.js`	View icon (eye)	Opens the page in the Preview panel
Server script	Play icon (#)	Runs the server script

These icons also appear on the parent folder of the file.

File status markers

The Explorer reflects Git status using the color and badge markers in the following table.

Color or marker	Meaning
Green text	New file (untracked or newly added)
Yellow or amber text	Modified file
Gray text	Ignored by `.gitignore`
Badge with count	Folder contains N changed files

Filtering and searching files

Press Cmd+Shift+F while the Explorer panel is focused to filter the file tree by name. Enter a partial file or folder name, and the tree filters in real time. Press Escape to clear the filter.

Moving, copying, and selecting files

- Move: Drag a file onto a destination folder to move it.
- Copy: Hold Option (macOS) or Ctrl (Windows) while dragging to copy.
- Multi-select: Hold Cmd (macOS) or Ctrl (Windows) and select multiple files, or use Shift+click for a range.

Open a project folder

Open an existing project folder from disk so that its files load into the Project Explorer. Opening a folder makes that project the one you browse, edit, and run in Lux Lab.

Before you begin
Role required: admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Select the project name in the title bar.
2. Select **Open Folder...**.
3. Navigate to and select your project root directory.

Result
The folder loads into the Project Explorer.

What to do next

- Create a file
- Create a folder

Create a file

Add a file to a folder in your project directly from the Project Explorer tree. The file appears in the folder you choose, ready to open in the Code Editor.

Before you begin
Role required: admin

Procedure

1. Hover over the folder in the tree where you want the file, to reveal the action icons.
2. Select the **New File** icon.

   Alternatively, right-click the folder and select **New File**.
3. Enter the file name.
4. Press Enter to confirm.

What to do next
For related information, see the following topics:

- Create a folder
- Code editor

Create a folder

Add a subfolder inside any folder of your project, directly from the Project Explorer tree. Use subfolders to group related files in your project.

Before you begin
Role required: admin

Procedure

1. Select and hold (or right-click) the folder in the tree where you want the subfolder.
2. Select **New Folder**.
3. Enter the folder name.
4. Press Enter.

What to do next
For more information, see the following topics:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Create a file
- Project explorer

Code editor

The Code Editor is the central workspace in Lux Lab. It provides syntax highlighting, context-aware code completion, multi-file navigation, and integration with the Lux platform.

Editor layout and tabs

Tabs

Each open file is represented by a tab in the tab bar. Tabs support the following interactions:

- Dragging to reorder
- Right-click for tab-specific actions, such as close, close others, close to the right, pin, and copy path
- Middle-click to close a tab
- Pinned tabs, which stay in the tab bar and cannot be closed accidentally

Preview mode

Files you open with a single click in the Project Explorer open in preview mode, with the tab title in italics. The next file you open replaces a preview tab. Double-click the tab or start typing to promote it to a permanent tab.

Tab groups and split views

You can split the editor to view multiple files side by side in the following ways:

- Split Right: Select and hold (or right-click) a tab and select **Split Editor Right**, or press Cmd+\ (macOS) / Ctrl+\ (Windows/Linux).
- Split Down: Right-click a tab and select **Split Editor Down**.

You can open as many editor groups as you need, and there's no enforced limit. Each group maintains its own tab list and scroll position. Drag tabs between groups to reorganize them. To close a split group, close all its tabs or drag the remaining tabs to another group.

For the steps to split the editor, see Split the editor.

Breadcrumb navigation

A breadcrumb trail in each editor group shows the file path and the current symbol, such as a function, class, or component. Select any segment to navigate to that location.

Syntax highlighting and language support

Lux Lab includes built-in grammar and highlighting support for the languages in the following table.

Language	Extensions
TypeScript	`.ts`, `.tsx`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Language	Extensions
JavaScript	`.js`, `.jsx`, `.mjs`
HTML	`.html`
CSS and SCSS	`.css`, `.scss`
JSON	`.json`, `.jsonc`
Markdown	`.md`
YAML	`.yaml`, `.yml`
XML	`.xml`

Language detection is automatic, based on the file extension. To override the detected language, select the language indicator in the status bar.

IntelliSense and code completion

Lux Lab provides language server-based, context-aware completions as you type, using the same IntelliSense engine as VS Code.

Triggering completions

- Completions appear automatically after a short delay.
- Press Ctrl+Space to trigger the completion list manually at any time.
- Use Tab or `Enter` to accept the selected suggestion.
- Press Escape to dismiss.

Parameter hints

When calling a function, press Cmd+Shift+Space (macOS) / Ctrl+Shift+Space (Windows/Linux) to see parameter documentation inline.

Go to definition

The following table lists the shortcuts for navigating to a definition.

Action	macOS	Windows
Go to definition	F12 or Cmd+Click	F12 or Ctrl+Click
Peek definition (inline pop-up)	Option+F12	Alt+F12

Code folding and minimap

Folding

Folding collapses blocks of code to reduce visual noise. The following table lists the folding shortcuts.

Action	Shortcut
Fold the current block	Cmd+Option+[/ Ctrl+Shift+[
Unfold the current block	Cmd+Option+] / Ctrl+Shift+]

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Action	Shortcut
Fold all	Cmd+K, Cmd+0
Unfold all	Cmd+K, Cmd+J
Fold level N (1–7)	Cmd+K, Cmd+N

Folding indicators, shown as chevrons, appear in the gutter when you hover over a foldable region.

Minimap

The minimap in the editor gives a condensed view of the entire file. Select anywhere on the minimap to jump to that position. To hide the minimap, right-click it and select **Hide Minimap**, or toggle it in **App Settings** > **Editor** > **Show Minimap**.

Multi-cursor and multi-selection

Adding cursors

The following table lists the shortcuts for adding cursors.

Action	Shortcut
Add cursor above	Cmd+Option+↑ / Ctrl+Alt+↑
Add cursor below	Cmd+Option+↓ / Ctrl+Alt+↓
Add cursor at click position	Option+Click / Alt+Click
Add cursors to all occurrences of selection	Cmd+Shift+L / Ctrl+Shift+L
Add next occurrence of selection	Cmd+D / Ctrl+D
Skip current occurrence	Cmd+K, Cmd+D / Ctrl+K, Ctrl+D

Column (box) selection

Hold `Shift+Option` (macOS) or Shift+Alt (Windows/Linux) and drag to create a rectangular selection across multiple lines.

Find and replace in a file

The following table lists the find and replace shortcuts.

Action	Shortcut
Open Find	Cmd+F / Ctrl+F
Open Find and Replace	Cmd+H / Ctrl+H
Next match	Enter or F3
Previous match	Shift+Enter or Shift+F3
Toggle case sensitive	Cmd+Option+C / Alt+C
Toggle whole word	Cmd+Option+W / Alt+W
Toggle regex	Cmd+Option+R / Alt+R

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Action	Shortcut
Replace current	Cmd+Option+E / Alt+E
Replace all	Cmd+Option+A / Alt+A

Editor keybindings

The following table lists the essential editor shortcuts for macOS and Windows/Linux.

Action	macOS	Windows/Linux
Open command palette or quick open	`Cmd+Shift+P`	`Ctrl+Shift+P`
Go to file	`Cmd+P`	`Ctrl+P`
Search in all files	`Cmd+Shift+F`	`Ctrl+Shift+F`
Save file	`Cmd+S`	`Ctrl+S`
Undo	`Cmd+Z`	`Ctrl+Z`
Redo	`Cmd+Shift+Z`	`Ctrl+Y`
Cut line	`Cmd+X`	`Ctrl+X`
Copy line	`Cmd+C`	`Ctrl+C`
Delete line	`Cmd+Shift+K`	`Ctrl+Shift+K`
Move line up	`Option+↑`	`Alt+↑`
Move line down	`Option+↓`	`Alt+↓`
Duplicate line down	`Shift+Option+↓`	`Shift+Alt+↓`
Comment or uncomment	`Cmd+/`	`Ctrl+/`
Block comment	`Shift+Option+A`	`Shift+Alt+A`
Format document	`Shift+Option+F`	`Shift+Alt+F`
Go to line	`Cmd+G`	`Ctrl+G`
Navigate back	`Cmd+[`	`Alt+←`
Navigate forward	`Cmd+]`	`Alt+→`
Split editor	`Cmd+\`	`Ctrl+\`
Toggle file explorer	`Cmd+B`	`Ctrl+B`
Toggle AI chat panel	`Cmd+Shift+B`	`Ctrl+Shift+B`
Toggle bottom panel	`Cmd+J`	`Ctrl+J`
Toggle terminal	`Ctrl+\``	`Ctrl+\``

Diff view

You can compare two files or check changes against the last committed version in the following ways:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Compare with saved: Right-click a modified file in the Explorer and select **Compare with Saved**.
- Compare with branch: Right-click and select **Compare with Branch...** to diff against any Git ref.
- Source control diff: Select a changed file in the Source Control panel to open an inline diff.

In the diff view, use F7 / Shift+F7 to jump between changed hunks.

Split the editor

Split the Code Editor into editor groups to work with multiple files side by side. Each group keeps its own tabs, so you can compare or edit related files without switching back and forth.

Before you begin

Role required: admin

Procedure

1. Select and hold (or right-click) any tab.
2. Select **Split Editor Right** or **Split Editor Down**.

   Alternatively, press Cmd+\ (macOS) or Ctrl+\ (Windows/Linux).
3. Open the file you want in the new group.

Result

The new editor group appears beside or below the original. Each group maintains its own tab list and scroll position, and there's no limit on the number of splits.

Terminal and panels

The bottom panel in Lux Lab contains five tabs that help you run commands, monitor your running application, track build and deploy progress, and review code quality issues.

The following table describes the purpose of each tab.

Bottom panel tabs

Tab	Purpose
Terminal	Full OS shell for running any CLI command
Console	Live logs from the running dev server
Build	Build output and deployment details
Problems	Linter diagnostics across the project
Tests	Test runner output

To toggle the entire bottom panel, press Cmd+J (macOS) / Ctrl+J (Windows). To open directly to the **Terminal** tab, press Ctrl+`.

Terminal

The integrated terminal is a full OS shell. On macOS it runs bash or zsh, and on Windows it runs PowerShell or cmd, depending on your system default. You can run any command you would normally run in a terminal, such as installing dependencies, using Git, running scripts, and invoking CLIs.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Terminal sessions

- Press Ctrl+` to open the Terminal tab.
- Select the **+** button in the terminal toolbar to start a new terminal session.
- Multiple sessions appear as tabs in the terminal tab bar. Select any tab to switch between them.
- Right-click a tab to rename, split, or close a session.

You can keep a dev server running in one session while running other commands in separate sessions at the same time.

Shell configuration

Lux Lab uses your system's default shell. To change it, go to **App Settings** > **Terminal** > **Default Shell** and select from the detected shells or enter a custom path.

Panel size

Drag the top edge of the bottom panel up or down to resize it. Press Cmd+J / Ctrl+J to hide and show the panel.

Console

The **Console** tab shows the live log output from your running dev server. When you start the development server (`pnpm dev`), the Console streams all console logs, warnings, and errors from the running application in real-time.

Use the Console to do the following:

- Monitor network requests and application logs during development.
- Catch runtime errors and warnings from your Lux application.
- See Hot Module Replacement (HMR) activity as you edit files.

Build

The **Build** tab shows the output from build operations and deployments:

- Build logs: Output from compiling and bundling your Lux application, such as `pnpm build`
- Deploy logs: Progress and results when you deploy to a ServiceNow instance

Each time you trigger a build or deploy, the output streams into this tab so you can monitor progress and diagnose any failures.

Problems

The **Problems** tab lists all linter diagnostics reported across your entire project, including errors, warnings, and informational messages. It works the same as the Problems panel in VS Code.

Problems list columns

Each entry shows the columns in the following table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Column	Description
Severity icon	Red circle (error), yellow triangle (warning), or blue circle (info)
Message	Description of the problem
Source	Tool that reported the problem, for example `eslint` or `ts`
File	File containing the problem
Line:Column	Exact location

Navigating to a problem

- Select any entry to jump to the exact location in the editor.
- Use F8 / Shift+F8 to cycle through problems in the current file from the editor.

Filtering

Use the filter bar at the top of the panel to search by message text or file name. To filter by severity, toggle the **Errors**, **Warnings**, or **Info** buttons.

Tests

The **Tests** tab captures output from your test runner scripts. When you run your tests from the Terminal, Lux Lab captures and displays the results here.

Next steps

- Agents and skills
- Deploy changes

Work items

Work items are plans stored as local files in your Lux Lab project. Each one contains a structured description for you and your AI agents of what needs to be built or fixed.

Work items and plan mode

Work items act as a lightweight task-tracking system tied directly to your project. Lux Lab creates a work item when you or the agent work in plan mode. When you ask the agent to plan a task before implementing it, Lux Lab saves the resulting plan as a work item file.

Work items have the following characteristics:

- They're stored as files in your local project.
- They're tied to a specific project, and each work item records which project it belongs to.
- AI agents can execute or iterate on them, using the plan as their guiding context.
- They're visible and manageable from within the app.

Work item access

You can open work items from the Activity Bar or from the Explorer sidebar.

Activity Bar

Select the **Work Items** icon in the Activity Bar to open the dedicated Work Items panel. The panel lists all work items associated with the current project.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Sidebar

A **Work Items** accordion is pinned at the bottom of the left sidebar, below the file tree. Expanding it shows all active work items for the project without leaving the Explorer view.

Work item creation

You create work items through the Agent Chat Harness, using plan mode. The agent produces a structured plan from your requirements, and you save that plan as a work item. You agree on an explicit plan before the agent makes any code changes.

For the steps, see Create a work item.

Work item storage

Work items are stored as files within your project directory. Because they're project-local files, they have the following properties:

- They're tied to the project they were created for.
- They're portable, so you can commit them to source control and teammates can see the planned work.
- They're readable by agents. When an agent starts working on a project, it can reference existing work item files as context.

Create a work item

Have an agent plan a task in plan mode, and then save that plan as a work item in your project. You agree on an explicit plan before the agent makes any code changes.

Before you begin

Role required: admin

The Agent Chat Harness panel is open. For ways to open it, see Agent chat harness.

Procedure

1. Switch the chat panel to plan mode by toggling the plan mode option in the input toolbar.
2. Enter a description of what you want to build or fix.
3. Press Enter.

   The agent processes the task and produces a structured plan of action.
4. When Lux Lab prompts you to create a work item from the plan, select **Create Work Item**.
5. **Optional:** To have the agent begin executing the plan, select **Start Implementation** on the new work item.

Result

Lux Lab saves the plan as a work item file in your project.

Agent chat harness

The Agent Chat Harness is the conversational AI interface built into Lux Lab. It uses the Agent Communication Protocol (ACP) to connect external AI agents to your project context.

ACP is a structured layer between your project and the agent, and it gives agents the context to interpret your code, your instance, and your intent.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Chat panel context

The Agent Chat Harness is a persistent chat panel embedded in the app, and it gives you direct conversational access to AI agents with context on the following:

- Your open files and project structure
- Your connected ServiceNow instance
- Your conversation and session history
- Git state and source control context

Agent Communication Protocol

ACP is the communication layer between Lux Lab and external AI agents. It defines how Lux Lab packages context from your project and sends it to the agent. It also defines how Lux Lab surfaces the agent's responses and proposed actions in the app.

Supported agents

Lux Lab supports connecting to the ACP-compatible agents in the following table.

Agent	Notes
Claude Agent	Supported; set as default
Devin	Supported
Codex	Supported; requires your own API key

Important: Lux Lab doesn't provide subscriptions or API keys for any agent. You must have your own active subscription or API key for the agent you use. Your own account with the respective provider governs agent usage and costs. Install and configure agents at your own discretion. Lux Lab only facilitates the connection.

Opening the chat panel

You can open the chat panel in the following ways:

- Select the **Agent Chat** icon (speech bubble) in the Activity Bar.
- Press Cmd+Shift+B (macOS) / Ctrl+Shift+B (Windows).

By default, the panel opens on the right side of the app.

Chat input toolbar

The bottom of the chat panel contains the message input and a toolbar with the controls in the following table.

Control	Description
+	Attaches context, for example images, files, code snippets, or plan board items
<> Code #	Switches the code context attached to the message

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Control	Description
Default (recommended) #	Model and agent selector for switching between configured agents
Gear icon (##)	Agent-specific settings, for example effort level, display when the active agent supports them
Microphone icon (#)	Voice input
Send icon (↑)	Sends the message

Switching agents

Select the model selector (**Default (recommended) #**) in the input toolbar to open the agent switcher, which lists the following options:

- **Claude Agent**: Default, displays a check mark when active
- **Devin**: Expand to configure
- **Codex**: Requires an API key (displays key icon)
- **Manage models & agents**: Opens the full agent management settings

Select any agent to switch to it for the current conversation. When the agent supports them, agent-specific settings such as effort level may appear next to the model selector.

Connecting a custom endpoint

Select **Add Custom Model** (top right) to open the connection dialog for any other OpenAI-compatible endpoint, including self-hosted or local servers such as Ollama. The dialog exposes the generic OpenAI-compatible fields in the following table.

Field	Description
Provider name	Label shown in the model selector for this provider
Base URL	Endpoint's OpenAI-compatible base URL, ending in `/v1`
Model id	Exact model identifier as reported by the endpoint
Display name	(Optional) Friendlier name shown in the model list
API key	(Optional) Key required by most hosted endpoints; local servers may ignore the value but still require the field to be non-empty
Context	(Optional) Model's context window size

Select **Test connection** to verify the endpoint responds before saving, then select **Add model**.

Attaching context with @

Enter `@` in the message input to open the context picker, which has the options in the following table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Description
Web	Includes web search results as context
Agents	References or switches to a specific agent
Skills	Invokes an available skill to guide the agent
Files	Attaches a specific file from your project
Git	Includes Git state, for example the diff and commit history, as context

Navigate the picker with ↑ / ↓, press Enter to select, and press Esc to cancel.

Quick-start actions

When you open a new conversation with no prior messages, the chat panel shows quick-start action buttons for example **Create a widget** and **Create a page**. These buttons pre-fill a partial prompt in the input. You then complete the prompt with the specific details of what you want to build before sending.

Agent responses

Agent responses stream into the conversation in real-time and can include plain text, code blocks, proposed file changes, and terminal commands.

Plain text

Explanations and analysis appear as formatted markdown.

Code blocks

Inline code snippets with syntax highlighting. Select **Copy** to copy the snippet, or **Insert at Cursor** to insert it directly into the active editor.

Proposed file changes (diffs)

When the agent proposes editing a file, Lux Lab shows a diff preview inline with the actions in the following table.

Button	Action
View Full Diff	Opens a side-by-side diff in the editor
Apply	Applies the change to the file immediately
Reject	Dismisses the proposed change

You can undo any applied change with Cmd+Z / Ctrl+Z.

Terminal commands

The agent may propose running a terminal command. A **Run** button appears next to the command. You can also copy the command and run it manually.

Conversation history and sessions

Lux Lab stores each conversation as part of the project, with the following results:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Conversation history persists when you reopen the project.
- Multiple agents can reference the shared conversation history as context.
- You can start a new conversation from the chat panel header.
- Past conversations are accessible from the conversation history list in the chat panel.

The chat panel also shows consumption details: how much of the agent's context window or quota your current session is using.

MCP manager

The three-dot menu (...) in the chat panel header opens the Model Context Protocol (MCP) manager. In the manager, you can view and manage all configured MCP servers that extend the agent's capabilities.

General guidelines for working with agents

The following general guidelines apply when you work with agents in the chat panel:

- **Be specific.** Instead of "fix my code," try "the error on line 42 of NavBar.js says X — how do I fix it?"
- **Use @ context.** Attach the relevant file or Git diff rather than pasting large code blocks.
- **Review diffs carefully.** Apply agent-proposed changes one at a time and verify before continuing.
- **Manage your quota.** Keep track of consumption details, because each agent has its own usage limits based on your subscription.

Agents and skills

The Agents and Skills panel is an explorer that surfaces all agents and skills available in your current project and on your system. It gives you a single place to check what is configured and ready to use.

Opening the panel

Select the **Agents and Skills** icon in the Activity Bar to open the panel. The panel lists the following items:

Agents

All ACP-compatible agents configured on your system, such as Claude, Devin, and Codex

Skills

All skills bundled with Lux Lab and any project-specific skills present in your project

Panel contents

The panel is a read-only explorer. It does not configure or install agents. It reflects the following existing setup:

- Bundled Lux skills are shown automatically.
- Project-level skills, if present in your project directory, are listed alongside the bundled ones.
- Configured agents appear as they're registered through the Agent chat harness.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

To configure agents or install new ones, use **Manage models & agents** from the agent switcher in the chat panel.

Skills in the chat

You can attach any skill visible in this panel to a chat message with the @ context picker, by selecting **Skills** from the picker. For details on using context in the chat, see Agent chat harness.

Related topics

- Agent chat harness
- Configuring Lux Lab

Configuring Lux Lab

Configuring Lux Lab means installing it and connecting it to your ServiceNow instance. You also set your preferences in the **App Settings** panel, including your profile, appearance, editor behavior, and integrations.

Minimum requirements

Lux Lab requires the operating system, tools, and connectivity in the following table.

Requirement	Minimum version
macOS	12 (Monterey) or later
Windows	10 (64-bit) or later
Node.js	v24 or later
pnpm	v10 or later
Git	v2.30 or later
Internet connection	Required, for instance connectivity and agent features

Setup tasks

Setup has two parts. First, you install Lux Lab from the ServiceNow® Developer Store. Then you connect it to your instance, so that you can access relevant files and deploy your changes. For the procedures, see Install Lux Lab and Connect Lux Lab to an instance.

App settings

Beyond installation and the instance connection, most other configuration is in the App Settings panel. For the panel's sections and options, see App Settings panel.

Install Lux Lab

Install Lux Lab on macOS or Windows from the ServiceNow® Developer Store. On first launch, a guided setup takes you through environment, agent, and instance sign-in steps.

Before you begin
Role required: admin

You need a ServiceNow® developer account to sign in to the ServiceNow Developer Store.

Verify that your machine meets the minimum requirements in the following tables. The recommended specifications aren't required. However, the recommended specifications support running the dev server, agent chat, and preview panel simultaneously.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

macOS requirements

Requirement	Minimum	Recommended
OS version	macOS 12 (Monterey) or later	macOS 13 (Ventura) or later
Processor	Apple Silicon (M1 or later)	Apple Silicon M2 or later
RAM	8 GB	16 GB or more
Disk space	2 GB free	10 GB or more free

Windows requirements

Requirement	Minimum	Recommended
OS version	Windows 10 (64-bit) or later	Windows 11
Processor	Intel Core i5 or AMD Ryzen 5	Intel Core i7 or AMD Ryzen 7 or better
RAM	8 GB	16 GB or more
Disk space	2 GB free	10 GB or more free

Lux Lab also requires the following:

- Node.js v24 or later, for local build tooling
- pnpm v10 or later, the recommended package manager
- Git v2.30 or later
- An active internet connection, for instance connectivity and agent features

About this task

Download the Lux Lab app using the following link. Lux Lab installation files.

Procedure

1. macOS

   a. Download the Lux Lab `.dmg` installer using the following link.
   Lux Lab installation files.

   b. Open the `.dmg` file.

   c. Drag Lux Lab into your `/Applications` folder.

   d. If macOS displays a security prompt on first launch, select **Open Anyway** in **System Settings** > **Privacy & Security**.

   Lux Lab opens and prompts you to sign in.

2. Windows

   a. Download the Lux Lab `.exe` installer using the following link.
   Lux Lab installation files.

   b. Run the installer.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

c. Follow the on-screen prompts.

d. When the installer completes, launch Lux Lab from the Start menu or the desktop shortcut.

Result
Lux Lab is installed. On first launch, a guided setup screen takes you through three steps before you reach the app: Environment, Agents, and Instance Sign-in.

What to do next
To complete setup, connect Lux Lab to your instance; this step is required, because you can't reach the app without it. For the procedure, see Connect Lux Lab to an instance.

Connect Lux Lab to an instance

Connect Lux Lab to your ServiceNow instance so that you can access instance files, deploy changes, and preview your work. Instance sign-in is required before you can reach the app.

Before you begin
Role required: admin

Your machine meets the minimum system requirements. For the requirements, see Configuring Lux Lab.

About this task
On first launch, instance sign-in is the third and final step of the guided setup screen. You must complete it before you can reach the home page. You can also add or switch instances at any time after setup from the Instance Explorer.

Procedure
1. Start a new instance connection in one of the following ways:
   - During first launch, go to the **Instance Sign-in** step of the guided setup.
   - After setup, select the instance name in the top header bar, or open the **Instance Explorer** panel from the Activity Bar. In either case, select **Add Instance**.
2. Enter your instance URL, including the protocol (for example, `https://dev12345.service-now.com`).
3. Enter your instance user name and password.

   Lux Lab authenticates to the instance with basic authentication.
4. Select **Connect**.

Result
An indicator next to the instance name shows the connection status, as described in the following table.

Connection status indicators

Indicator	Meaning
Green dot	Connected and responsive
Yellow dot	Connected but slow or degraded
Red dot	Connection failed or timed out

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Connection status indicators (continued)

Indicator	Meaning
Gray dot	Disconnected

What to do next

After you connect, you can create an experience or browse instance tables from the Instance Explorer panel. For the procedure to create an experience, see Create an experience.

App Settings panel

The App Settings panel contains configuration for your profile, appearance, editor behavior, instance connection, integrations, and updates.

Ways to open the panel

You can open the App Settings panel in the ways shown in the following table.

Method	Action
Keyboard shortcut	Cmd+, (macOS) or Ctrl+, (Windows)
Activity Bar	Select the gear icon at the bottom of the Activity Bar

The Settings panel opens as a modal dialog and displays the Lux Lab version at the top left, for example v2.0.8.

Settings sections

The panel's sidebar lists the sections in the following table.

Section	Description
Profile	Your name, email, and profile picture
Appearance	Theme, brand colors, and visual depth settings
Editor	Code editor preferences
Context	How the agent collects and uses project context
Instance	ServiceNow instance connection settings
Integrations	Figma integration and Now SDK version
Updates	Lux Lab update preferences
Advanced	Advanced configuration options

Profile

The Profile section shows your connected account. You can update the display information in the following table.

Field	Description
Name	Your display name

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Email	Your account email address
Profile picture	Custom avatar that you upload, or a preset that you choose

The Profile section also has the following controls:

- **Presets** — Displays built-in avatar options to select from.
- **Upload** — Uploads a custom profile image.
- **Refresh** — Re-syncs your profile information from your account.

Appearance

The Appearance section sets the theme, brand colors, and dark surface depth.

Theme

The theme control switches between **Light** and **Dark** mode.

Brand colors

Brand colors customize the app color palette, as described in the following table.

Color	Description
Primary	Main accent color used for active states and highlights
Accent	Secondary accent color
Neutral	Neutral or background color

The brand colors have the following controls:

- **Shuffle** — Randomly generates a new color combination.
- **Reset** — Restores the default color set.

Dark surface depth

When Dark mode is active, **Dark Surface Depth** controls how much surface elevation contrast Lux Lab applies. The options are **2**, **4**, **7**, and **10**. Higher values give more contrast between layered surfaces.

Editor

The Editor section controls code editing preferences such as tab size, font settings, word wrap, and formatter behavior.

Context

The Context section controls how the agent gathers and uses project context in the Agent Chat Harness. This includes how many files, how much history, and which parts of the project the agent includes automatically with each request.

Instance

The Instance section holds the following ServiceNow instance connection settings:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Default instance on startup
- Connection timeout settings
- Auto-refresh behavior for the Instance Explorer

Integrations

The Integrations section has the following settings:

- **Figma** — Connects Lux Lab to Figma for design-to-code workflows directly from the app.
- **Now SDK Version** — Sets the version of the Now SDK that your project uses. The default is **latest**.

Updates

The Updates section controls how Lux Lab checks for and applies updates. You can choose stable releases or pre-release builds.

Advanced

The Advanced section holds configuration options for power users and non-standard setups.

Default settings

Selecting **Reset to Defaults** at the bottom of the Settings panel restores all settings to their original values. Your projects, work items, and connected instances are unaffected.

Using Lux Lab

With Lux Lab, you can create new experiences or extend existing ones, and build the pages and widgets they need. You work on your own or with an AI agent, preview changes as you go, and deploy them to your instance.

Lux Lab development tasks

Lux Lab combines Lux-aware intelligence, AI tools, and title-bar commands for building tailored experiences on the ServiceNow AI Platform. You start by creating a new experience or extending an existing one. You can preview changes in real time, and deploy them to your instance with one click when you're ready.

Depending on your project, you create or extend pages manually or with an AI agent in the Agent Chat Harness. The following topics cover each task:

Create an experience

Build a Lux experience from scratch in Lux Lab. The new project comes with sample pages, widgets, and the standard project structure, ready for you to add your own pages and widgets.

Before you begin

Your machine meets the minimum system requirements in Configuring Lux Lab. Lux Lab is connected to an instance. For instructions, see Connect Lux Lab to an instance.

Role required: admin

About this task

To create an experience that extends an existing application instead, see Extend an existing experience.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Select the project name in the title bar.
2. Select **New Project**.
3. In the **Create New Project** dialog, on the **Create New** tab, select the **Base AIUX template**.
4. Enter a **Project Name**.
5. Select a **Project Location** on your local drive.

   To pick a folder, select **Browse**.
6. Confirm or edit the **Scope**.

   The field is auto-filled from your connected instance, for example with `x_snc_`. The full scope takes the form `x_abc_myapp`.
7. Select **Create Project**.

Result

Lux Lab scaffolds the project, installs dependencies, and opens it in the app automatically. The new project contains sample pages, widgets, and other Lux Lab experience metadata, following the standard project structure (`aiux.json`, `application.js`, `pages/`, `widgets/`, and so on).

What to do next

Start creating pages and widgets in your experience. For the ways to add a page, see Create a page with Lux Lab.

Extend an existing experience

Create an extension project in Lux Lab to add pages and widgets to an experience that already exists on the ServiceNow AI Platform. Lux Lab scaffolds the project from its Extension template.

Before you begin

- Role required: admin
- Your machine meets the minimum system requirements in Configuring Lux Lab.
- Lux Lab is connected to an instance. For instructions, see Connect Lux Lab to an instance.

Procedure

1. Select the project name in the title bar.
2. Select **New Project**.
3. In the **Create New Project** dialog, on the **Create New** tab, select the **Extension template**.
4. Enter a **Project Name**.
5. Select a **Project Location** on your local drive.
6. Confirm or edit the **Scope**.
7. Select **Create Project**.

Result

Lux Lab scaffolds your extension project and opens it in the app automatically.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Create a page with Lux Lab

Create a page in a new experience so that you can add elements to it, preview it, and deploy it to an instance. You can create the page from a context menu, by hand, or with an AI agent.

Before you begin

- Role required: admin
- Your machine meets the minimum system requirements in Configuring Lux Lab.
- You have an experience to add the page to. For instructions, see Create an experience.

About this task

Procedure

In your experience, create a page using one of the following methods.

- Create a page from the Explorer panel context menu
  1. Right-click the `/pages` folder, or a file or folder within your project.
  2. Select **Add Page...**.

  This Lux-specific action scaffolds a new Lux page directly into your project.
- Create a page manually
  1. Select the `/pages` folder.
  2. With the `/pages` folder highlighted, select the create folder icon.
     Alternatively, right-click and select **New Folder**.
  3. Enter a name for the new folder and press Enter.
  4. In the new folder, select the create file icon.
     Alternatively, select and hold (or right-click) in the folder and select **New File**.
- Create a page with an AI agent
  1. Open the Agent Chat panel.
  2. Use the **Create a page** quick-start action, or enter a prompt describing the page to create.

Result

Your new page is created. You can add elements to the page and preview it as needed.

What to do next

- Preview a page
- Deploy changes

Preview a page

Preview a page from within Lux Lab to test that it appears and functions as intended. The preview updates automatically as you save source files, without a full page reload.

Before you begin

Role required: admin

Your machine meets the minimum system requirements in Configuring Lux Lab.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

About this task

You can also start the development server by selecting **Run** or **Preview** in the title bar, which opens the running application in the Preview panel.

Procedure

1. Open a Lux Lab project.
2. In the Explorer panel, navigate to the `/pages` folder.
3. Hover over the file of the page to preview, or over its parent folder, to reveal the inline **View** icon (an eye).
4. Select the **View** icon.

Result

A preview of your page opens as an embedded browser tab. The preview toolbar has the controls in the following table.

Preview toolbar controls

Control	Description
URL bar	Path or URL editor that navigates when you press Enter
Refresh	Preview page reload button
Open in browser	Button that opens the current URL in your default system browser
Device presets	Viewport size selector for Mobile, Tablet, and Desktop presets
Screenshot	Screenshot tool that captures the current preview and sends it to the Agent Chat panel

Saving a source file triggers Hot Module Replacement (HMR), which updates the preview automatically without a full page reload.

Lux Lab development server

Running your Lux application starts a local development server that serves your application and reflects code changes automatically.

Starting the development server

From the title bar

You can start the development server from the title bar at the top of the app, using the following buttons:

- **Run** starts the development server.
- **Preview** starts the development server and opens the Preview panel at the same time.

As the server starts, the button updates to reflect its current state. The button labels for each state are listed in the following table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Run button states

State	Button label
Server stopped	**Run**
Server starting	Transitioning (indicator shown)
Server running	**Stop**

You can select **Stop** at any time to stop the development server.

From the terminal

You can also start the development server manually from the integrated terminal.

```
pnpm dev
```

Monitoring the development server

While the development server is running, all of its logs stream into the **Console** tab of the bottom panel. These logs include startup messages, HMR updates, errors, and console output from your application.

The Console tab shows the following activity:

- Development server startup progress
- Hot Module Replacement (HMR) activity when files are saved
- Runtime logs and errors from your running application

Hot Module Replacement

While the development server is running, saving any source file triggers HMR. Lux Lab recompiles only the changed module and pushes the update to the preview without a full page reload, preserving component state where possible.

Changes to configuration files, such as `aiux.json`, take effect only after you restart the development server.

Stopping the development server

To stop the development server, select **Stop** in the title bar, which is shown while the server is running. You can also press `Ctrl+C` in the terminal session that is running the development server.

Build errors

Errors that keep the development server from starting or from serving a module appear in the following tabs:

- **Console** tab: runtime errors from the running application
- **Problems** tab: linter and type errors in your source files
- **Build** tab: build pipeline failures

Selecting an error in the **Problems** tab opens the offending line in the editor.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Production build

To build a production bundle before deploying, run the build command.

```
pnpm build
```

The build writes its output to the `dist/` directory. Build logs stream into the **Build** tab of the bottom panel.

Deploy changes

Deploy your project from Lux Lab to a connected instance, where you can preview and publish your experiences, pages, and widgets. Lux Lab builds the project and installs it on the instance in one action.

Before you begin

Your machine meets the minimum system requirements in Configuring Lux Lab.

Lux Lab is connected to an instance. For instructions, see Connect Lux Lab to an instance.

Role required: admin

About this task

You can deploy manually, as the following steps describe, or conversationally with an AI agent in the Agent Chat Harness.

To show more options, select the arrow next to **Deploy**.

Deploy options

Option	Description
Build	Build step only, without deploying
Custom Deploy	Deployment to a different, previously connected instance instead of the one you're logged into
Project scripts	Scripts defined in the current project's `package.json`, runnable directly from the drop-down list

Procedure

1. Select **Deploy** in the title bar, or press `Cmd+Shift+D` (macOS) or `Ctrl+Shift+D` (Windows).
2. If you have multiple projects open in the workspace, select the project to deploy.

Result

Lux Lab builds the application, compiling and bundling the source code into deployable artifacts. It then runs `now-sdk install` to deploy the built artifacts to your active connected instance. Build and deploy output streams into the **Build** tab of the bottom panel. If the deployment succeeds, a confirmation banner appears.

What to do next

Preview and test your changes on your instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ServiceNow Lux Lab for VS Code extension

The ServiceNow Lux Lab for VS Code extension transforms how you build on the ServiceNow AI Platform. You can scaffold experiences, pages, and widgets with framework intelligence built in, cutting development time and ensuring consistency across your projects.

Get started

Explore  Learn about the ServiceNow Lux Lab for VS Code extension.	Configure  Install the ServiceNow Lux Lab for VS Code extension and connect it to your instance.
Use  Create and extend application experiences, pages, and widgets.	

Helpful resources

Some ServiceNow resources that can provide you with helpful information are:

ServiceNow Community

ServiceNow Community

Developer

developer.servicenow.com

Impact

http://impact.servicenow.com

ServiceNow University

ServiceNow University

Best Practices

Best Practices

Partner

https://www.servicenow.com/partners.html

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ServiceNow

http://servicenow.com

ServiceNow Store

http://servicenow.com

Support

- https://support.servicenow.com/now
- Known Error Portal

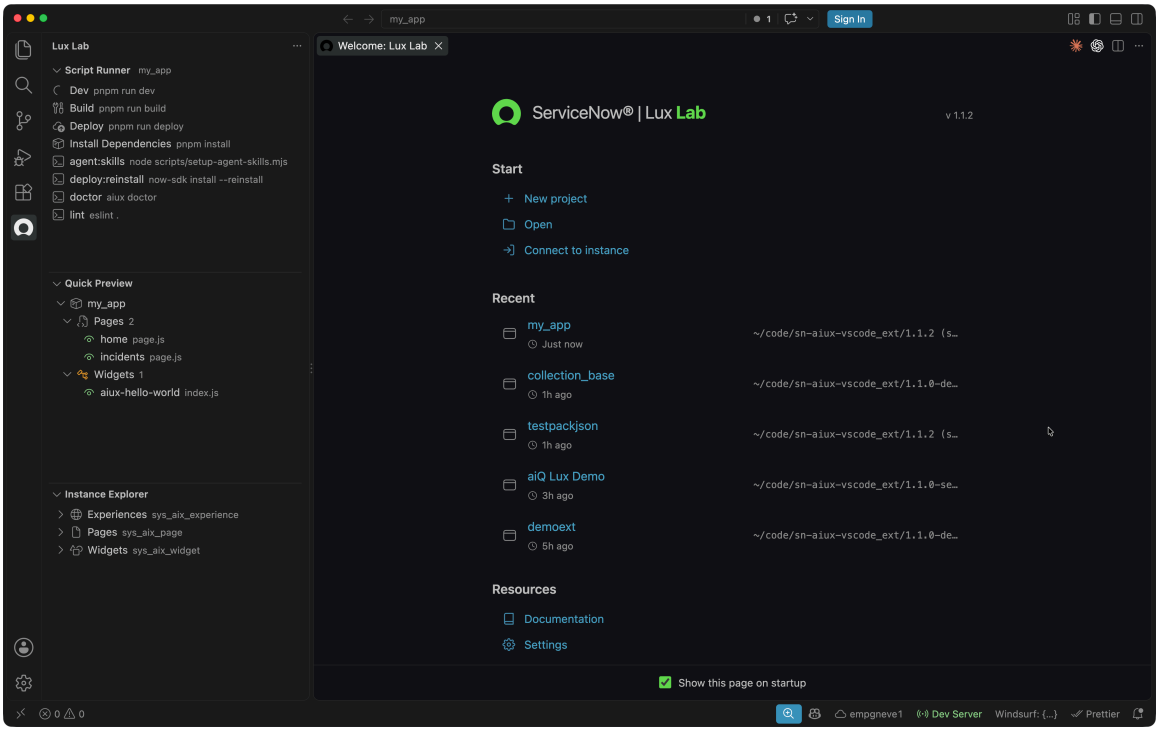
Exploring ServiceNow Lux Lab for VS Code extension

The ServiceNow Lux Lab for VS Code extension provides framework intelligence and quick commands to help you create and extend ServiceNow experiences with pages and widgets tailored to fit your needs.

ServiceNow Lux Lab for VS Code extension overview

The ServiceNow Lux Lab for VS Code extension enables you to create and extend ServiceNow experiences. With built-in framework intelligence, you can run quick commands, preview entire applications, and see which files are relevant for your work from inside the explorer. Create experiences, pages, and widgets that support tailored functionality in your ServiceNow applications.

ServiceNow Lux Lab for VS Code Welcome screen



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ServiceNow Lux Lab for VS Code users

Users

User	Description
Developer	Build new experiences, pages, and widgets. Or extend base system ServiceNow applications with tailored experiences, pages, and widgets.

ServiceNow Lux Lab for VS Code workflow

1. Create a new experience or extend an existing experience.
2. Create or extend pages and widgets manually or using agentic development tools.
3. Preview your changes from within Visual Studio Code.
4. Deploy your changes.

ServiceNow Lux Lab for VS Code benefits

Key benefits

Benefit	Feature	Users
Run commands via the ServiceNow Lux Lab for VS Code extension interface rather than using the terminal.	Script Runner	Developer
Preview your application in real time without leaving Visual Studio Code.	Preview	Developer
Access relevant experiences, pages, and widgets from your instance.	Explorer	Developer
Develop tailored experiences with AI tools.	Claude Code in Visual Studio Code	Developer

What to explore next

To learn more about configuring and using ServiceNow Lux Lab for VS Code, see:

- Configuring ServiceNow Lux Lab for VS Code extension
- Using ServiceNow Lux Lab for VS Code extension

Configuring ServiceNow Lux Lab for VS Code extension

Install the ServiceNow Lux Lab for VS Code extension and connect it to your instance.

Configuration overview

The ServiceNow Lux Lab for VS Code extension requires the following:

Application	Version	Resources for more information
Visual Studio Code	1.97 or later	Visual Studio Code updates
Node.js	24 or later	Node.js

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Application	Version	Resources for more information
pnpm	10 or later	pnpm
ServiceNow SDK	4.12.1 or later	ServiceNow SDK
ServiceNow instance	• Australia Patch 5 or later • Zurich Patch 12 or later	Prepare your upgrade

1. Install ServiceNow Lux Lab for VS Code extension.

   Install the ServiceNow Lux Lab for VS Code extension from the Visual Studio Code marketplace.

2. Connect ServiceNow Lux Lab for VS Code extension to an instance

   Connect to your instance to access relevant files and deploy your changes.

Install ServiceNow Lux Lab for VS Code extension

Install the ServiceNow Lux Lab for VS Code extension from the Visual Studio Code marketplace.

Before you begin

Role required: none

The ServiceNow Lux Lab for VS Code extension requires the following:

Application	Version	Resources for more information
Visual Studio Code	1.97 or later	Visual Studio Code updates
Node.js	24 or later	Node.js
pnpm	10 or later	pnpm
ServiceNow SDK	4.12.1 or later	ServiceNow SDK
ServiceNow instance	• Australia Patch 5 or later • Zurich Patch 12 or later	Prepare your upgrade

Procedure

1. Open Visual Studio Code and navigate to the **Extensions** tab by clicking the Extensions tab icon () or pressing Control+Shift+X on Windows or Command+Shift+X on macOS.
2. In the Visual Studio Code Marketplace, enter *`ServiceNow Lux Lab for VS Code`* in the **Search Visual Studio Code Extensions** search box.
3. Select *`ServiceNow Lux Lab for VS Code`* in the search results
4. Select **Install** to proceed with installation.
5. After the installation process completes, restart Visual Studio Code.

Result

You installed the ServiceNow Lux Lab for VS Code extension.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

What to do next

Complete the ServiceNow Lux Lab for VS Code extension setup process by connecting to an instance.

Connect ServiceNow Lux Lab for VS Code extension to an instance

Connect the ServiceNow Lux Lab for VS Code extension to your ServiceNow instance to access instance files, deploy changes, and preview your work.

Before you begin

Role required: none

The ServiceNow Lux Lab for VS Code extension requires the following:

Application	Version	Resources for more information
Visual Studio Code	1.97 or later	Visual Studio Code updates
Node.js	24 or later	Node.js
pnpm	10 or later	pnpm
ServiceNow SDK	4.12.1 or later	ServiceNow SDK
ServiceNow instance	• Australia Patch 5 or later • Zurich Patch 12 or later	Prepare your upgrade

Procedure

1. Add a new instance.
   - From the home screen, select **Add a new instance**.
   - From the command palette, select **Add a new instance**.
2. In the authentication type field, select either **Basic** or **OAuth** to match your instance configuration.
3. In the instance URL field, enter the URL for your instance, including the protocol.
   For example, `https://<instance-name>.service-now.com`.
4. Enter your instance username and password.
   For OAuth authentication, complete the additional authentication steps in the browser window that opens.
5. In the instance alias field, enter a name for your instance.

Result

If your instance is connected successfully, a notification in Visual Studio Code indicates that your instance has been added and lists the instance name.

Using ServiceNow Lux Lab for VS Code extension

Use the ServiceNow Lux Lab for VS Code extension to create or extend experiences with pages and widgets tailored to fit your needs.

Overview of using ServiceNow Lux Lab for VS Code

With the ServiceNow Lux Lab for VS Code extension, you can harness framework intelligence, AI tools, and easy-to-use commands to create tailored experiences on the ServiceNow AI Platform®. Start by creating a new experience or extending an existing one. Build or extend

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

pages and widgets on your own or with the help of your preferred AI tools. Discover relevant tables and files and preview changes in real-time as you work. When you're ready, deploy changes to your instance with one click.

- Create a new experience

  Build new experiences for the ServiceNow AI Platform®.

- Extend an existing experience

  Add tailor-made pages and widgets to existing ServiceNow applications.

- Create pages for new experiences or existing ones, or extend existing pages
- Depending on your project type, create or extend pages manually or using your preferred AI tools.
- Preview a page

  Test that your pages appear and function as intended from within Visual Studio Code.

- Add a page to the navigation

  Make your pages accessible from ServiceNow application sidebars or within other pages.

- Deploy changes

  When you're ready, deploy your changes to a target instance.

Create a new experience

Develop ServiceNow Lux Lab for VS Code experiences from scratch.

About this task

This procedure describes how to create net new experiences. To create an experience that extends an existing application, see Extend an existing experience.

Before you begin

Role required: none

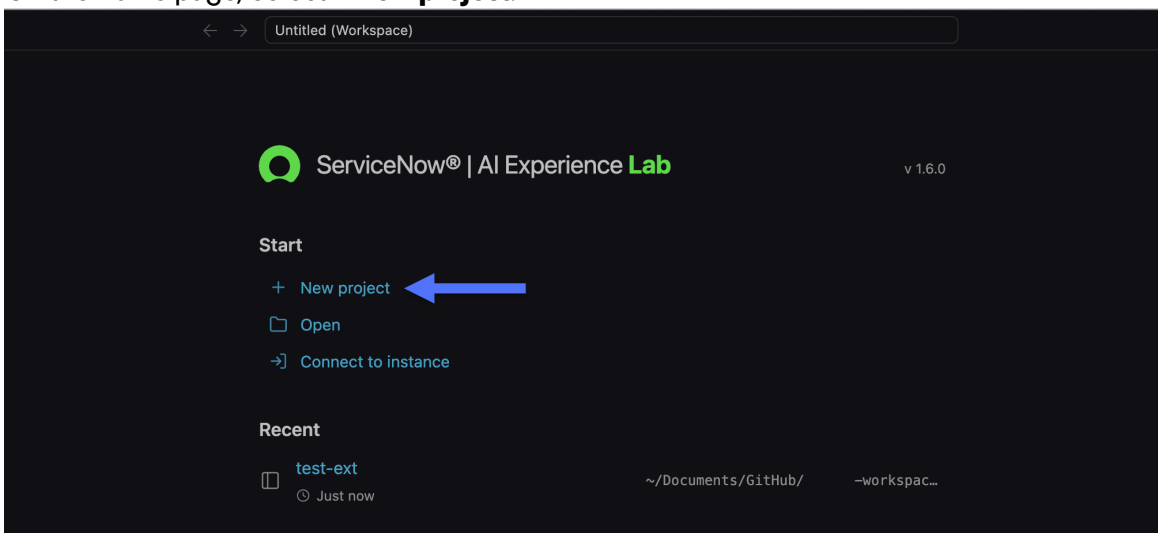
The ServiceNow Lux Lab for VS Code extension requires the following:

Application	Version	Resources for more information
Visual Studio Code	1.97 or later	Visual Studio Code updates
Node.js	24 or later	Node.js
pnpm	10 or later	pnpm
ServiceNow SDK	4.12.1 or later	ServiceNow SDK
ServiceNow instance	• Australia Patch 5 or later • Zurich Patch 12 or later	Prepare your upgrade

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to Visual Studio Code.
2. Select the ServiceNow icon in the side panel to open the ServiceNow Lux Lab for VS Code extension.
3. On the home page, select **+ New project**.



4. Select the instance that you want to deploy changes to.
   During ServiceNow Lux Lab for VS Code extension setup, you connect to the instance that contains the files you want to access and where you want to preview and deploy changes. For more information or to connect another instance, see Connect ServiceNow Lux Lab for VS Code extension to an instance.
5. Select the location where you want your want your project to live on your local drive.
6. Enter a name for your experience and then select the Enter key.
7. Enter a name for the folder for your experience and then select the Enter key.
8. From the list, select the option **Standalone App**.
9. Enter a name for your application scope and then select the Enter key.
   ServiceNow Lux Lab for VS Code extension scaffolds your experience. Once created, Visual Studio Code reloads with your experience files loaded.

Result

Your new experience is created. Inside your experience, you can see sample pages, widgets, and other ServiceNow Lux Lab for VS Code experience metadata.

What to do next

Start creating pages and widgets within your experience. For more information, see Create a page.

Create a page

Create a page using the ServiceNow Lux Lab for VS Code extension to create tailored experiences.

About this task

This procedure describes how to create pages for new experiences. To create a page that extends an existing experience, see Create a page.

The following procedure describes how to complete this task manually. You can also complete the task using agentic development tools, such as Claude Code.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

Role required: none

The ServiceNow Lux Lab for VS Code extension requires the following:

Application	Version	Resources for more information
Visual Studio Code	1.97 or later	Visual Studio Code updates
Node.js	24 or later	Node.js
pnpm	10 or later	pnpm
ServiceNow SDK	4.12.1 or later	ServiceNow SDK
ServiceNow instance	• Australia Patch 5 or later • Zurich Patch 12 or later	Prepare your upgrade

Procedure

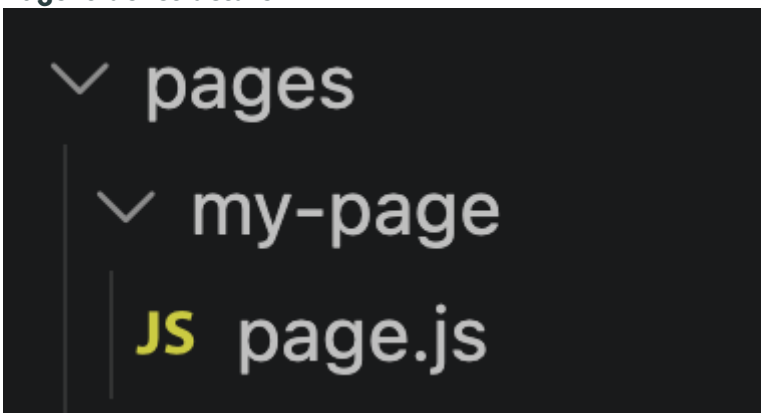
1. In the ServiceNow Lux Lab for VS Code extension, create an experience or open an existing experience.
   For information about creating an experience, see Create a new experience.
2. In your experience, create a page.
   - Create a page manually in the explorer panel.
     a. Select the /pages folder.
     b. With the /pages folder highlighted, select the create folder icon or right click and select **New folder**.
     c. Enter a name for the new folder and select the Enter key.
     d. In the new folder, select the create file icon or right click in the folder and select **New file**.
   - Create a page using the command palette.
     a. Access the command palette by pressing Ctrl+Shift+P on Windows or Command+Shift+P on macOS, or by navigating to **View** > **Command Palette**.
     b. Select **Lux: Create page**.
   - Create a page conversationally using Claude Code or your preferred AI tool.
     a. Open the panel for your preferred AI tool.
     b. Enter a prompt describing the page that you want to create.

        For example, `Create a new page in my example-experience project.`
3. Enter a name for the page and select the Enter key.
4. Enter a URL route for the new page and select the Enter key.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

After entering the URL route, the ServiceNow Lux Lab for VS Code extension creates the folder and page file using the information you provided.

Page folder structure



Result

Your new page is created. You can now add elements to the page and preview as needed.

What to do next

- Preview a page
- Add a page to the navigation
- Deploy changes

Extend an existing experience

Create an experience that extends an existing experience on the ServiceNow AI Platform®.

Before you begin

Role required: none

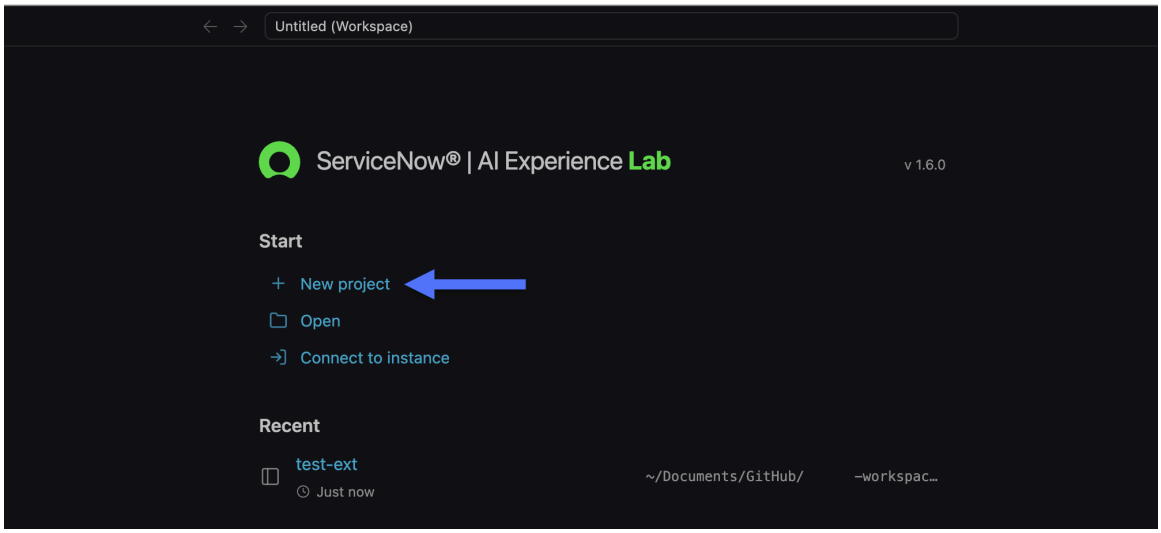
The ServiceNow Lux Lab for VS Code extension requires the following:

Application	Version	Resources for more information
Visual Studio Code	1.97 or later	Visual Studio Code updates
Node.js	24 or later	Node.js
pnpm	10 or later	pnpm
ServiceNow SDK	4.12.1 or later	ServiceNow SDK
ServiceNow instance	• Australia Patch 5 or later • Zurich Patch 12 or later	Prepare your upgrade

Procedure

1. Navigate to Visual Studio Code.
2. Select the ServiceNow icon in the side panel to open the ServiceNow Lux Lab for VS Code extension.
3. On the home page, select **+ New project**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



4. Select the instance that you want to deploy changes to.
   During ServiceNow Lux Lab for VS Code extension setup, you connect to the instance that contains the files you want to access and where you want to preview and deploy changes. For more information or to connect another instance, see Connect ServiceNow Lux Lab for VS Code extension to an instance.
5. Select the location where you want your want your project to live on your local drive.
6. Enter a name for your experience and then select the Enter key.
7. From the list, select the option **Extend Existing App**.
8. Select the name of the experience that you want to extend from the list.
   The list of experiences that can be extended are populated from the instance that you're connected to. Verify that the experiences you created and want to extend have been deployed to your instance.
9. Enter a name for your application scope and then select the Enter key.
   ServiceNow Lux Lab for VS Code extension scaffolds your experience. Once created, Visual Studio Code reloads with your experience files loaded.

Result

Your extension experience is created. The pages for the experience are located under the `/src/extensions/<your-experience-scope-name>/your-experience-name` folder structure. When you create an extension experience, a sample page and widget are also created.

What to do next

Create a page or extend an existing experience for your extension experience.

Create a page

Create a new page for an existing experience.

About this task

The following procedure describes how to complete this task manually. You can also complete the task using agentic development tools, such as Claude Code.

Before you begin

Role required: none

The ServiceNow Lux Lab for VS Code extension requires the following:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Application	Version	Resources for more information
Visual Studio Code	1.97 or later	Visual Studio Code updates
Node.js	24 or later	Node.js
pnpm	10 or later	pnpm
ServiceNow SDK	4.12.1 or later	ServiceNow SDK
ServiceNow instance	• Australia Patch 5 or later • Zurich Patch 12 or later	Prepare your upgrade

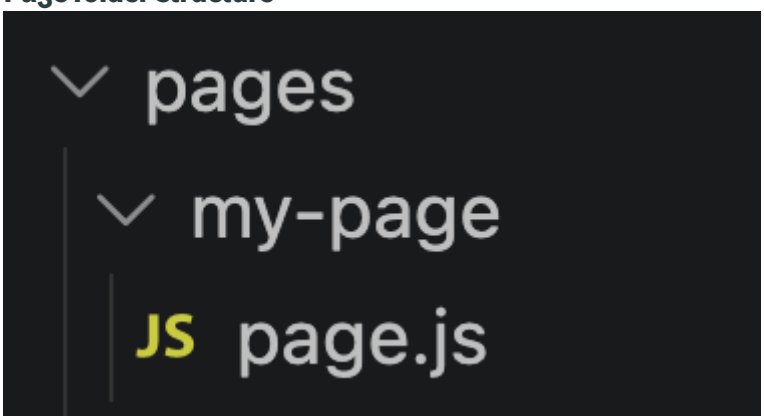
Procedure

1. In the ServiceNow Lux Lab for VS Code extension, open an experience that extends an existing experience.
   For more information about extending existing experiences, see Extend an existing experience.
2. Access the command palette by pressing Ctrl+Shift+P on Windows or Command+Shift+P on macOS, or by navigating to **View** > **Command Palette**.
3. Select **Lux: Create Page**.
4. From the list, select **AIUX: New Page**.
   You can also create pages that extend existing pages. For more information, see Extend an existing page.
5. Enter a display name for the page and select the Enter key.

   Important: The display name you enter must be unique. If the display name matches the name of another page in your extension experience, the new page will override the existing page with the same name. For more information about overriding pages, see Override an existing page.

6. Enter a URL route for the new page and select the Enter key.
   After entering the URL route, the ServiceNow Lux Lab for VS Code extension creates the folder and page file using the information you provided.

Page folder structure



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Result
Your new page is created. You can now add elements to the page and preview as needed. Certain pages aren't previewable in the ServiceNow Lux Lab for VS Code extension and must be previewed by deploying to your instance.

What to do next
- Preview a page
- Add a page to the navigation
- Deploy changes

Extend an existing page

Modify an existing page by extending it.

About this task
The following procedure describes how to complete this task manually. You can also complete the task using agentic development tools, such as Claude Code.

Before you begin
Role required: none

The ServiceNow Lux Lab for VS Code extension requires the following:

Application	Version	Resources for more information
Visual Studio Code	1.97 or later	Visual Studio Code updates
Node.js	24 or later	Node.js
pnpm	10 or later	pnpm
ServiceNow SDK	4.12.1 or later	ServiceNow SDK
ServiceNow instance	• Australia Patch 5 or later • Zurich Patch 12 or later	Prepare your upgrade

Procedure
1. In the ServiceNow Lux Lab for VS Code extension, open an experience that extends an existing experience.
   For more information about extending existing experiences, see Extend an existing experience.
2. Access the command palette by pressing Ctrl+Shift+P on Windows or Command+Shift+P on macOS, or by navigating to **View** > **Command Palette**.
3. Select **Lux: Create Page**.
4. Select **Extend Existing Page**.
5. Select the page that you want to extend from the list.
   The list of pages that appear is based on the instance that you're connected to and which pages can be extended. Pages that appear with the circle slash icon  have already been extended.
   After selecting the page that you want to extend, ServiceNow Lux Lab for VS Code extension loads the folder and page file.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Result
Your page extension is created. You can now add elements to the page and preview as needed. Certain pages aren't previewable in the ServiceNow Lux Lab for VS Code extension and must be previewed by deploying to your instance.

What to do next
- Add a page to the navigation
- Deploy changes

Override an existing page
Have existing ServiceNow AI Platform® experiences use pages that you create instead of base system pages.

About this task
The following procedure describes how to complete this task manually. You can also complete the task using agentic development tools, such as Claude Code.

Before you begin
Role required: none

The ServiceNow Lux Lab for VS Code extension requires the following:

Application	Version	Resources for more information
Visual Studio Code	1.97 or later	Visual Studio Code updates
Node.js	24 or later	Node.js
pnpm	10 or later	pnpm
ServiceNow SDK	4.12.1 or later	ServiceNow SDK
ServiceNow instance	• Australia Patch 5 or later • Zurich Patch 12 or later	Prepare your upgrade

Procedure
1. In the ServiceNow Lux Lab for VS Code extension, open an experience that extends an existing experience.
   For more information about extending existing experiences, see Extend an existing experience.
2. Access the command palette by pressing Ctrl+Shift+P on Windows or Command+Shift+P on macOS, or by navigating to **View** > **Command Palette**.
3. Select **Lux: Create Page**.
4. From the list, select **Lux: New Page**.
   You can also create pages that extend existing pages. For more information, see Extend an existing page.
5. Enter the display name for the page you want to override and select the Enter key.

> **Important:** The display name must exactly match the display name for the page you're overriding, in order for the override to be successful. Pages that appear with the circle slash icon have already been overridden.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

6. Enter the URL route for the page you want to override and select the Enter key.

> **Important:** The URL route must exactly match the URL route for the page that you're overriding, in order for the override to be successful.

After entering the URL route, the ServiceNow Lux Lab for VS Code extension creates the folder and page file using the information you provided.

Result
Your page is created. You can now add elements to the page and preview as needed.

What to do next
To complete the override process, you must deploy your changes to your instance. For more information, see Deploy changes.

Create a collection page

Add a collection page to a route using the ServiceNow Lux Lab for VS Code extension.

About this task
A route belongs to an experience, and an experience can contain multiple routes. A route maps a URL to a page. A collection page lets you add one or more additional pages under an existing route without changing its base page. Use this when you want several variations of a page to share the same URL.

When a user navigates to the route, the extension evaluates the route's collection pages in ascending order by their order value. For each collection page, it checks whether the user has one of the roles required to view it. The user sees the first collection page whose role requirement they meet. If the user does not meet the role requirement for any collection page, they see the route's base page instead.

The following procedure describes how to complete this task manually. You can also complete the task using agentic development tools, such as Claude Code.

Before you begin
Role required: none

You must have an existing route before you create a collection page for it. For information about creating a page, see Create a page.

The ServiceNow Lux Lab for VS Code extension requires the following:

Application	Version	Resources for more information
Visual Studio Code	1.97 or later	Visual Studio Code updates
Node.js	24 or later	Node.js
pnpm	10 or later	pnpm
ServiceNow SDK	4.12.1 or later	ServiceNow SDK
ServiceNow instance	• Australia Patch 5 or later • Zurich Patch 12 or later	Prepare your upgrade

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. In the ServiceNow Lux Lab for VS Code extension, create an experience or open an existing experience.
   For information about creating an experience, see Create a new experience.
2. Start the collection page flow.
   If you already have the page folder for the route open in the file explorer, starting there skips the experience and route selections. Otherwise, use the command palette, which walks you through selecting the experience and route.
   - Using the command palette.
     a. Access the command palette by pressing Ctrl+Shift+P on Windows or Command+Shift+P on macOS, or by navigating to **View** > **Command Palette**.
     b. Select **Lux: Create Collection Page**.
     c. If your project contains more than one experience, select the experience that contains the route.
     d. Select the route that you want to add the collection page to. Routes are listed alphabetically.
   - Using the file explorer.
     a. In the explorer panel, select and hold (or right-click) one of the following:
        - An existing collection page folder for a route.
        - An individual folder for a collection page within it (for example, `_collections/admin`).
     b. Select **Lux: Create Collection Page**. Selecting and holding (or right-clicking) any other item in the explorer opens the same guided flow as the command palette, starting with the experience and route selections.

     Note: Selecting and holding (or right-clicking) a collection page's page.js file directly does not show this option.
3. Enter a name for the collection page and press Enter.
   The name cannot be blank and must contain at least one character usable in a folder name. The name cannot duplicate an existing collection page name on that route. Matching is case-insensitive.
4. Enter an order for the collection page, or accept the suggested value, and press Enter.
   The extension suggests an order based on the highest order already used on that route, plus 10 (or 10 if the route has no collection pages yet). If you enter a value that duplicates an existing order or reaches the route's base page order, the extension shows a non-blocking warning but still accepts the value.
5. Select whether to restrict the collection page by role.
   Select **No additional role restriction** so that any user who can reach the route can open the collection page. Select **Restrict by role** to open a searchable, check box list of roles from the connected instance. Search for and check one or more roles, then select **OK**. A user sees the collection page only if they have at least one of the selected roles.

Result

The extension creates the collection page under the selected route and opens the new page file automatically. Collection pages are stored in a _collections folder within the route's page folder, with each collection page in its own subfolder (for example, `home/_collections/admin/page.js`). The route's original `page.js` file remains unchanged as the base page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

This procedure only changes local files; the collection page isn't visible on your instance until you deploy your changes.

What to do next

- Preview a page
- Add a page to the navigation
- Deploy changes

Preview a page

Test that your page appears and functions as intended.

About this task

Certain pages aren't previewable in the ServiceNow Lux Lab for VS Code extension and must be previewed by deploying to your instance. To learn about deploying, see Deploy changes.

Before you begin

Role required: none

The ServiceNow Lux Lab for VS Code extension requires the following:

Application	Version	Resources for more information
Visual Studio Code	1.97 or later	Visual Studio Code updates
Node.js	24 or later	Node.js
pnpm	10 or later	pnpm
ServiceNow SDK	4.12.1 or later	ServiceNow SDK
ServiceNow instance	• Australia Patch 5 or later • Zurich Patch 12 or later	Prepare your upgrade

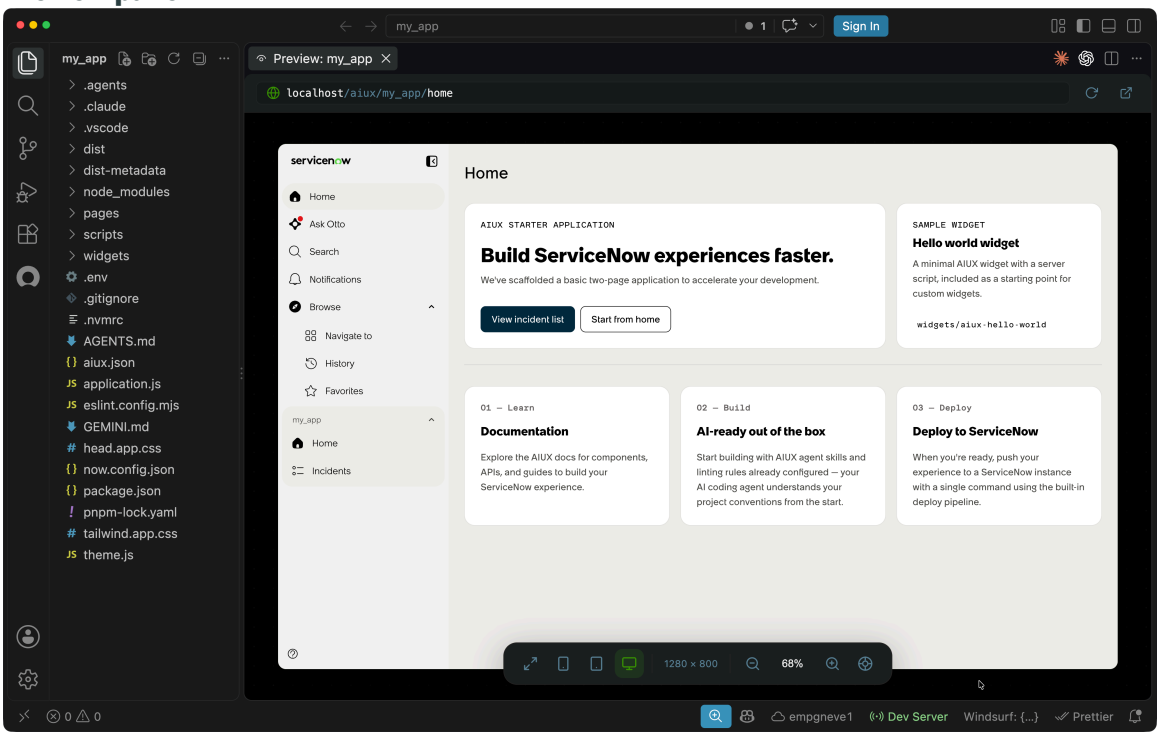
Procedure

1. Open a ServiceNow Lux Lab for VS Code project in Visual Studio Code.
2. In the Explorer panel, navigate to the /pages folder.
3. Select the expand icon for the page that you want to preview.
4. Right click or select and hold the page.js file that you want to preview and select **Lux: Launch Preview**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

A preview of your page appears. You can preview how your page appears on different devices, expand the preview panel, and interact with page elements.

Preview panel



What to do next

If your page isn't already accessible from other pages, add the page to the navigation. Otherwise, you can deploy your changes.

Add a page to the navigation

Add pages to the navigation to make them accessible from other pages, such as a home page.

About this task

Some pages are added to the navigation by default when you create them. For pages that aren't added automatically added to the navigation, use the following procedure to make them display and accessible via other pages.

The following procedure describes how to complete this task manually. You can also complete the task using agentic development tools, such as Claude Code.

Before you begin

Role required: none

The ServiceNow Lux Lab for VS Code extension requires the following:

Application	Version	Resources for more information
Visual Studio Code	1.97 or later	Visual Studio Code updates
Node.js	24 or later	Node.js
pnpm	10 or later	pnpm
ServiceNow SDK	4.12.1 or later	ServiceNow SDK

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Application	Version	Resources for more information
ServiceNow instance	• Australia Patch 5 or later • Zurich Patch 12 or later	Prepare your upgrade

Procedure

1. Select the ServiceNow icon in the side panel to open the ServiceNow Lux Lab for VS Code extension.
2. Open a net new or extension experience.
3. **Optional:** If your project doesn't already have an application.js file, at the experience folder root (not under `src/extensions/`), add an application.js file.

   Tip: Use the nav helper API from `@servicenow/aiux/aiux-components-nav` rather than adjusting the `navLayoutContext` by hand.

4. Double-click (or use the keyboard shortcut) to open the `application.js` file in explorer.
5. Add the following lines of code to the application.js file with the page name and target path that you want to add.

   Note: To add pages to L2, include an `l2Nav: [{icon, title, action}]` array on the L1 page you pass to `addNavItem`. The page accepts {icon, title, action, l2Nav?, l3Nav?}.

   A navigation entry has the following format:

```
setup(ctx) {
  const items = [
    {
      icon: 'home',
      title: i18n.getMessage('Home'),
      action: {type: 'navigate', path: '/home'}
    },
    {
      icon: 'list',
      title: i18n.getMessage('Incidents'),
      action: {type: 'navigate', path: '/incidents'}
    }
  ];
}
```

Action	Code to add in application.js
Add pages to L1 sidebar	`addNavItem(item, {source?})`
Set (replace) the L3 tab array on any L1 or L2 page by path	`setL3Nav(targetPath, l3Nav)`
Append tabs to an L1 or L2 page, keeping any tabs already present	`appendL3Nav(targetPath, l3Nav)`

6. Save the file.
7. In the **Script Runner** panel, select **Build**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

8. Verify that the added page appears at the expected navigation level using the **Quick Preview** panel or by deploying to your instance.
   For more information, see the following resources:
   - Preview a page
   - Deploy changes

Deploy changes

Deploy your changes to an instance to preview and publish experiences, pages, and widgets.

About this task

The following procedure describes how to complete this task manually. You can also complete the task using agentic development tools, such as Claude Code.

Before you begin

Role required: none

The ServiceNow Lux Lab for VS Code extension requires the following:

Application	Version	Resources for more information
Visual Studio Code	1.97 or later	Visual Studio Code updates
Node.js	24 or later	Node.js
pnpm	10 or later	pnpm
ServiceNow SDK	4.12.1 or later	ServiceNow SDK
ServiceNow instance	• Australia Patch 5 or later • Zurich Patch 12 or later	Prepare your upgrade

Procedure

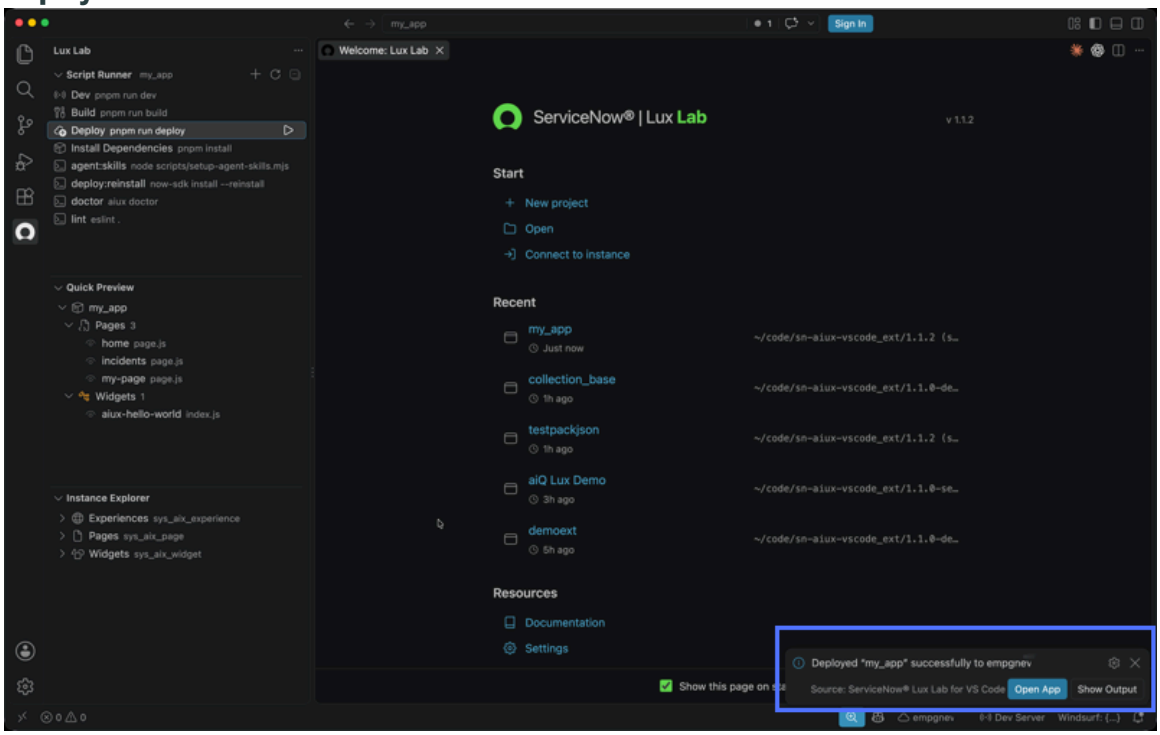
1. Expand the **AI Experience Lab: Script Runner** panel.
2. Select **Deploy**.

Result

The ServiceNow Lux Lab for VS Code extension attempts to deploy your changes. If successful, a notification appears confirming that the deployment to your target instance was successful.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Deployment success notification

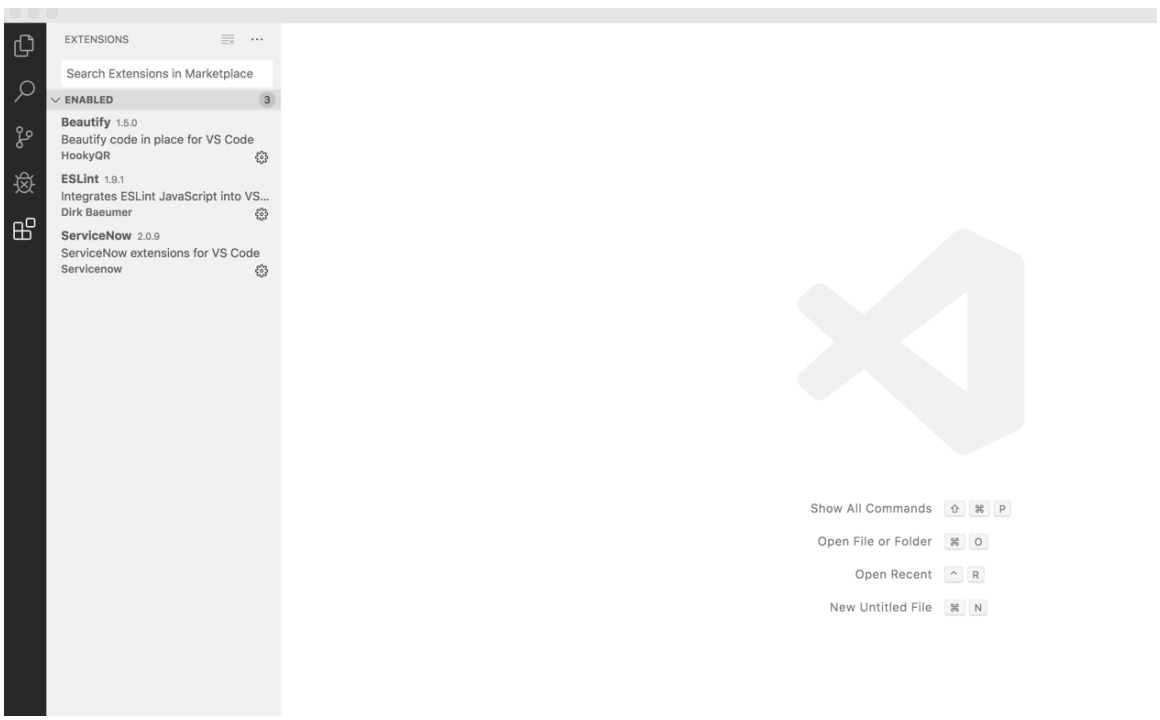


What to do next
Preview and test your changes on your instance.

ServiceNow Extensions for Visual Studio Code

Edit your ServiceNow applications in Visual Studio Code with the help of the ServiceNow Extensions for VS Code.

The ServiceNow Extensions for the VS Code editor enables you to edit applications within your ServiceNow instance.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Advantages of using Visual Studio Code

Edit your applications offline

Use Visual Studio Code to download and edit a local copy of your application. You can edit the application offline, and then synchronize again when your instance is available.

Use Visual Studio Code JavaScript features to reduce development time and improve code quality.

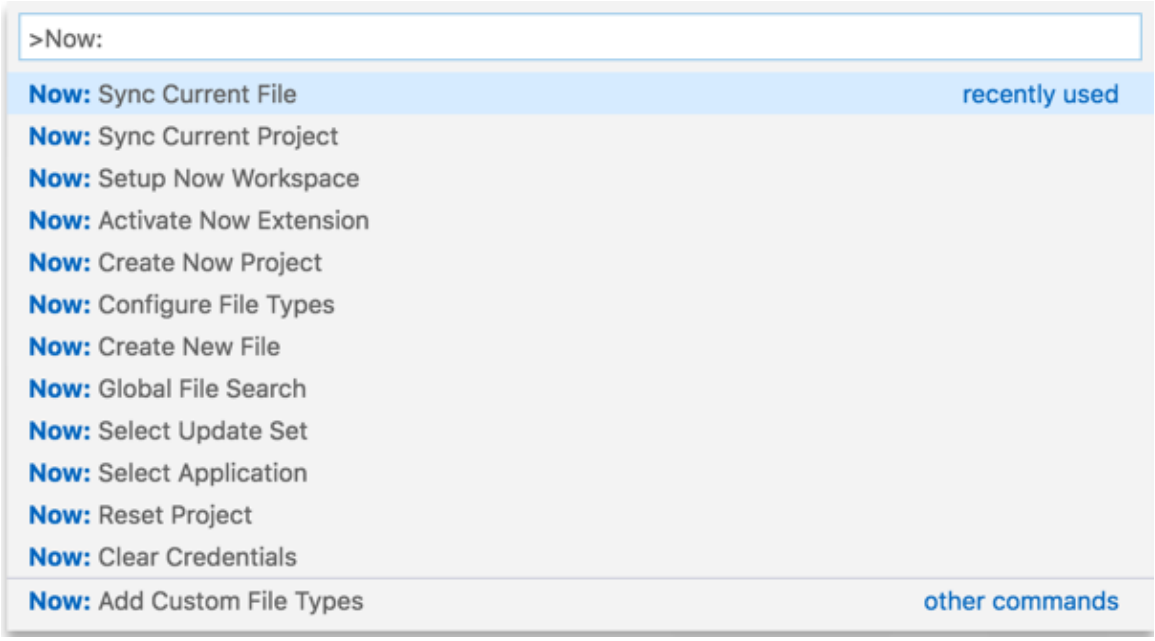
Visual Studio Code IntelliSense includes editing features such as code completion, code suggestions, and quick information. Use these tools to complete your coding tasks quickly and reduce errors. For more details on using IntelliSense in your applications, see IntelliSense in VS Code. The extension also supports Linting using ESLint. The same standard ESLint rules used in the ServiceNow instance are available.

Functions of ServiceNow Extensions for VS Code

The ServiceNow Extensions for VS Code extension includes tools for developing on the ServiceNow AI Platform.

The extension adds several functions to your Visual Studio Code (VS Code) implementation. You access these functions through the command palette.

Visual Studio Code command palette



Available functions in the ServiceNow extension for VS Code

Functionality	Description
Setup Now Workspace	Create the project folder to work with ServiceNow applications. For details on creating this workspace, see Set up a workspace in VS Code.
Activate Now Extension	Activates the ServiceNow extension. See Activate ServiceNow Extensions for VS Code.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Available functions in the ServiceNow extension for VS Code (continued)

Functionality	Description
Create Now Project	Import existing ServiceNow applications to a Visual Studio Code project. See Create a project in VS Code.
Sync Current Project	During development, synchronize all files between VS Code and the instance. See Synchronize the current project between a Visual Studio Code workspace and a ServiceNow instance.
Sync Current File	Synchronize the current file you are working on. See Sync the current file between a Visual Studio Code workspace and a ServiceNow instance.
Clear Credentials	Deletes the entered credentials in the Settings page. See Clear instance credentials in Visual Studio Code.
Reset Project	Sets project to its original state. See Reset a project in Visual Studio Code.
Configure File Types	Modify the options selected in the metadata. Metadata determines which elements of your application you want to synchronize with VS Code. See Import an application into Visual Studio Code.
Add Custom File Types	Add file types to your project using the Custom File Types wizard. See Add custom file types in Visual Studio Code.
Select Application	Switch between application within the workspace in VS code. See Import an application into Visual Studio Code.
Select Update Set	Select an update set. Changes synchronized to your instance are applied to the selected update set. For more information on update sets, see Import an application into Visual Studio Code.
Create New File	Create records in your application. For details, see Create a file in VS Code.
Global File Search	Find files within the instance. See Search files on your instance in VS Code.
Run Background Scripts	Run a background script on your instance. See Run background scripts using VS Code.
IntelliSense	An IntelliSense code-completion aid available for Glide APIs. See IntelliSense in VS Code.

Install ServiceNow Extensions for VS Code

Download and install ServiceNow extensions to begin editing your applications in VS Code.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin
Role required: none
In order to install ServiceNow Extensions for Visual Studio Code, you must have VS Code version 1.38.0 or later installed.

Procedure

1. Open VS Code and navigate to the **Extensions** tab by clicking the Extensions tab icon () or pressing Control+Shift+X on Windows or Command+Shift+X on MacOS.
2. In the Visual Studio Code Marketplace, enter `ServiceNow` or `ServiceNow Extensions` in the **Search Visual Studio Code Extensions** search box.
3. Select **ServiceNow Extensions for VS Code** in the search results and click **Install** to proceed with installation.
4. After the installation process completes, restart Visual Studio Code.
5. Navigate to **View** > **Command Palette** in Visual Studio Code.
   You can also use a keyboard shortcut (Control+Shift+P on Windows and Command+Shift+P on MacOS) to open the command palette.
6. Choose **Developer: Reload Window** from the list.

What to do next
Once you reload VS Code activate the ServiceNow extension.

Activate ServiceNow Extensions for VS Code

Activate the ServiceNow Extensions for VS Code to be able to edit applications within your ServiceNow instance.

Before you begin
Install ServiceNow Extensions for VS Code to activate the extension and begin editing your applications in VS code.

Role required: none.

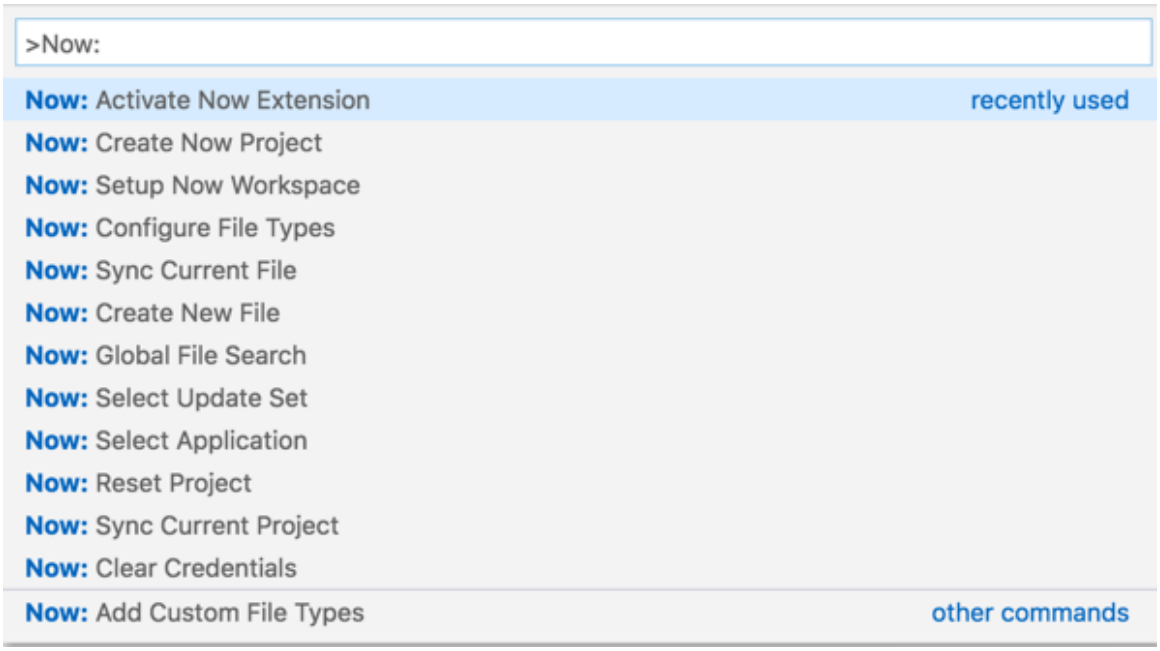
About this task
Activating the extension is the first step you must perform after installation of the extension to access its functionalities.

Procedure

1. Navigate to **All** > **View** > **Command Palette** to open the command palette.
   You can also use a keyboard shortcut, Control+Shift+P on Windows or Command+Shift+P on MacOS, to open the command palette.
2. Choose **Now: Activate Now Extension** from the list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Activate Now extension



The **Setup Workspace** link appears at the bottom of the VS Code IDE. After the extension is activated, Set up your workspace in VS Code to begin editing ServiceNow applications.

Set up a workspace in VS Code

Using the ServiceNow Extensions for VS Code, create a project work folder to use as a workspace for ServiceNow applications.

Before you begin

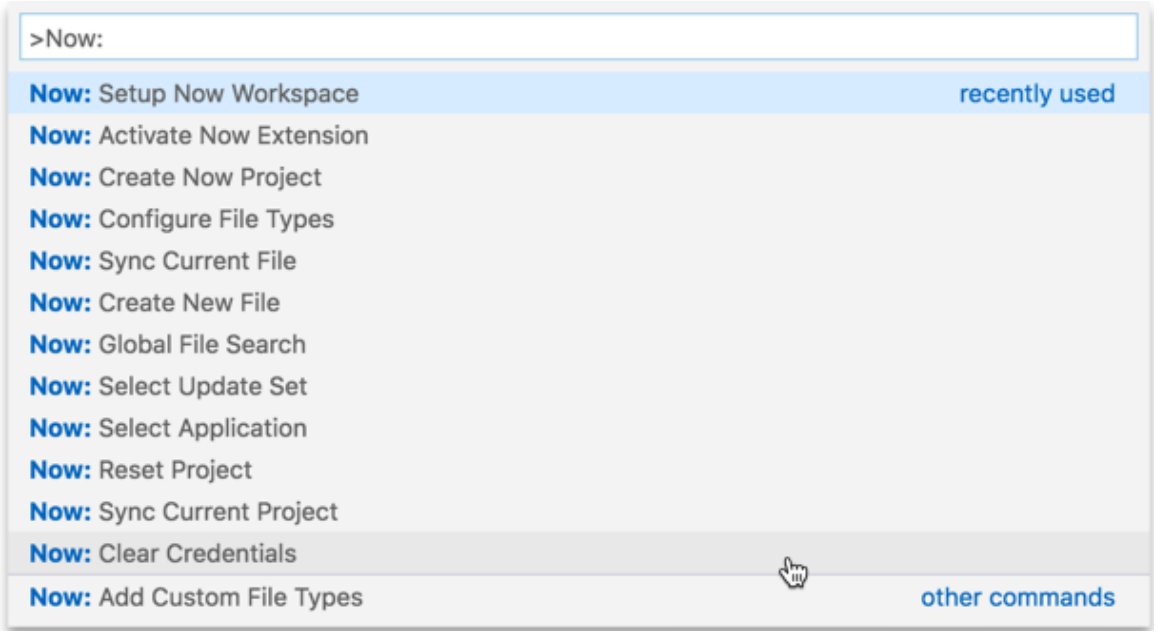
Activate the workspace to access the functionalities of ServiceNow Extensions for VS Code. Role required: admin

Procedure

1. Click the **Setup Workspace** on the status bar at the bottom of the VS Code IDE.
   You can also use a keyboard shortcut, Control+Shift+P on Windows or Command+Shift+P on MacOS, to open the command palette and choose **Now: Setup Now Workspace** from the list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Setup Now Workspace command



2. Do any of the following actions.

Option	Description
To set up a workspace from a new folder	**a.** Click **New Folder**. **b.** Enter a name for the folder in the **Name of new Folder** window. **c.** Click **Create**.
To set up a workspace from an existing folder	**a.** Select the folder on your system. **b.** Click **Select project folder**.

The workspace is created from the selected folder.

What to do next

Create a project in your workspace. You can create multiple projects of different project types within a workspace.

Create a project in VS Code

ServiceNow applications are contained within a project in the VS Code IDE. Import an application from your ServiceNow instance with the help of ServiceNow Extensions for VS Code.

Before you begin

Setup your workspace to create a project and start editing your ServiceNow applications.

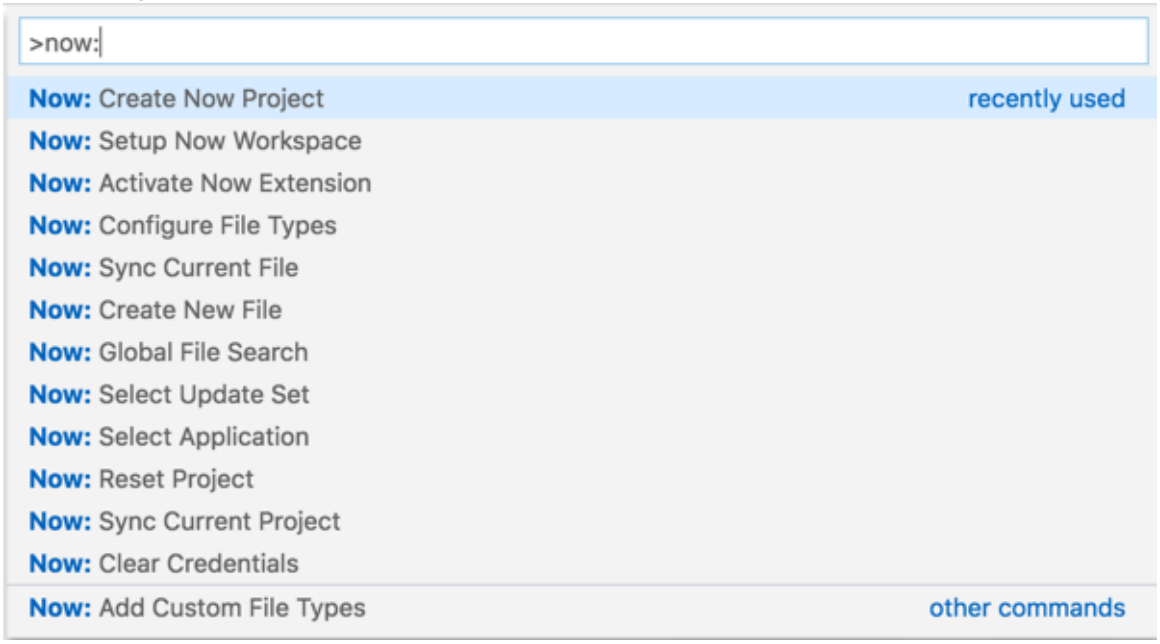
Role required: none.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Click **Create Project**in the status bar on the bottom of the VS Code IDE.

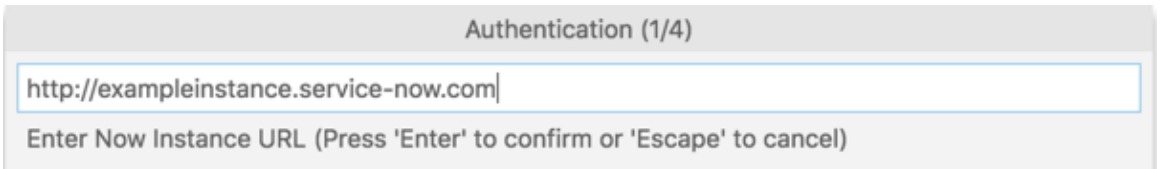
Create Project



You can also use a keyboard shortcut, Control+Shift+P on Windows or Command+Shift+P on MacOS, to open the command palette and choose **Now: Create Now Project** from the list.

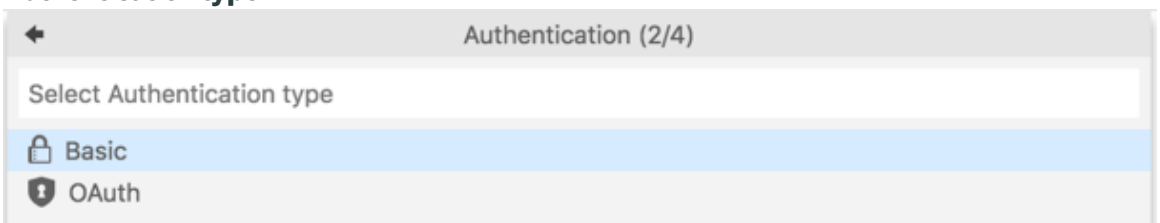
The **Create Project** wizard opens.

2. Enter the URL of the ServiceNow instance and press Enter to confirm.
Include the complete URL for your instance, for example, `https://example.service-now.com`



3. When prompted, select the Authentication type.

Authentication type



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- To create a basic application, choose **Basic** and fill in the following fields, when prompted.

Basic application

Field	Description
User Name	Enter the User name of the ServiceNow instance URL.
Password	Enter the password of the ServiceNow instance URL.

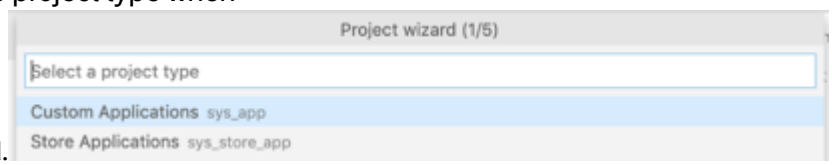
Note: The user account being used for authenticating must not contain the admin role.

- To create an OAuth application, choose **OAuth** application and fill in the following fields, when prompted.

OAuth application

User Name	Enter the Username of the ServiceNow instance URL.
Password	Enter the password of the ServiceNow instance URL.
Client ID	Client ID is automatically generated by ServiceNow OAuth server.
Client secret	TClient secret for the OAuth application.

4. Select the project type when prompted.



Note: You can enable the Packages and Plugins project types by enabling them from the Settings. Navigate to **Preferences** > **Settings** > **Extensions** > **Now** > **Project Settings** to enable them.

5. Select **Import Existing**.

Import an application into Visual Studio Code

After you create a project, import an application from your instance into the project to begin editing.

Before you begin
Role required: none.

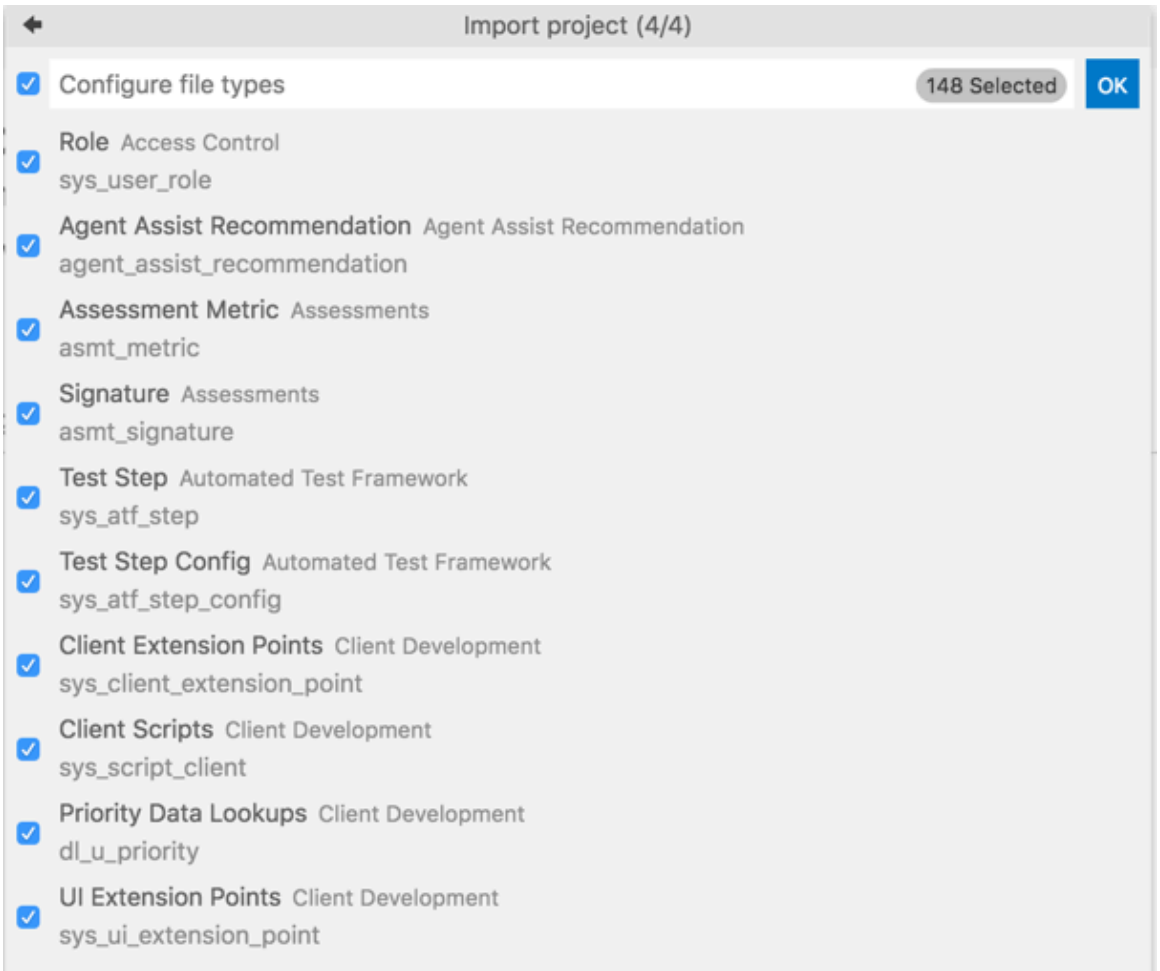
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Visual Studio Code displays the list of applications available in your instance for the selected application type.

   The list of applications shows once the Manage Developers settings have been applied for the custom application intended to import. If the settings are left unchecked, applications will not be listed in the drop-down.

2. Select an application.
3. At the **Configure file types** prompt, click **OK** to select all file types.



By default, Visual Studio Code imports all the file types of your application. You can modify the selection of the file types while editing the application, by choosing **Now: Configure File Types** from the command palette and choosing the file types from the list.
Your ServiceNow application is imported into your workspace.

Note: Even if your application exists on your instance, you must create a project for it in Visual Studio Code.

You can switch applications within the workspace by clicking the name of the application, for example, EmployeeApp in this case, in the status bar at the bottom of the VS code IDE or choosing **Now: Select Application** from the command palette and selecting the application name from the list. Similarly, you can modify the update set by clicking the current update set icon, for example, EmployeeApp in this case, in the status bar at the bottom of the VS code IDE or by choosing **Now: Select Update Set** from the command pallet and selecting the name of the updateset.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Synchronization between Visual Studio Code and a ServiceNow instance

You can synchronize your applications between Visual Studio Code and your ServiceNow instance.

Synchronization

Changes made to your applications in Visual Studio Code are stored on your local file system only until you synchronise with your instance.

Synchronization accounts for the following changes:

- File modifications on the instance or in the local file system
- File creation on the instance or in the local file system
- File deletion on the instance or in the local file system

You can sync your currently selected record or all records in the current application.

Sync the current file between a Visual Studio Code workspace and a ServiceNow instance

Identify conflicting files and merge or overwrite the changes, between the current file in your Visual Studio Code workspace and your instance.

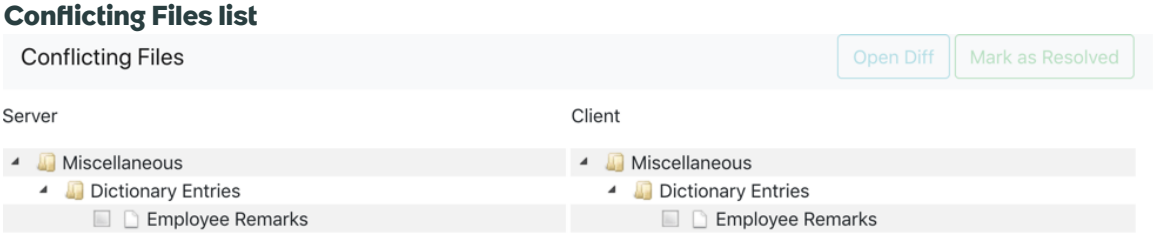
Before you begin

Role required: admin

Procedure

1. Click the **Sync** icon from the menu on the top-right corner of the Visual Studio Code IDE. You can also use a keyboard shortcut Control+Shift+P in Windows and Command+Shift+P in MacOS to open the command palette and choose **Now: Sync Current File** from the list.
   - If the changes occur either on the server or on the client, the extension synchronizes the file successfully without showing up any errors.
   - If there are no differences between the local and server versions, then the message **No changes detected** appears on the bottom of the page.
   - If the same file is modified both on the client and the server, then the system displays the conflicts in a **Conflicting Files** dialog box.
2. If a conflict list is displayed in a **Conflicting Files** dialog box, check the conflicts.
   - Compare the **Server** and **Client** version using the **Diff** window, and evaluate the changes.
   - Override the **Server** or **Client** changes.

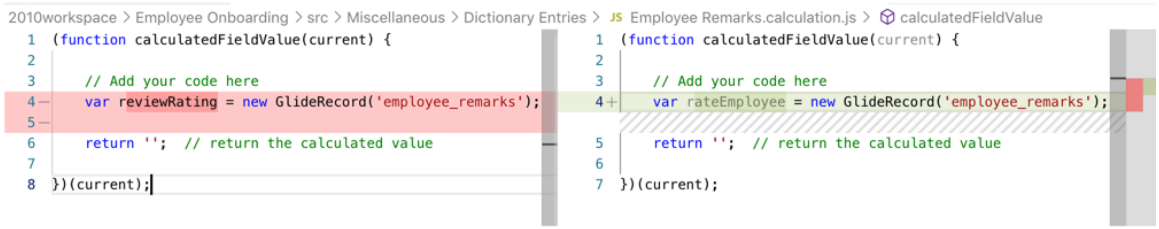
Conflicting Files list



3. Check for the differences in the **Conflicting Files** list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Conflicting changes display



A dialog box displays the following options:

- Open Diff: Check for the differences between two versions and make the necessary changes.
- Mark Resolved: Resolve the differences and sync the versions.
- Overwrite Server: Choose if you want to override the server version with the client version.
- Overwrite Local: Choose **Override Local** to override the local version with the server version.

4. Click **Open Diff**.
   You can see two versions of the file with the differences highlighted.

Differences in client and server



5. Make changes to the server version from the client version and save the file.
6. Click **Mark as Resolved** under **Conflicting Files**.
   The changes are successfully synchronized upon the next synchronization. When the changes are merged successfully, the message **Sync completed successfully** appears at the bottom of the page.

Synchronize the current project between a Visual Studio Code workspace and a ServiceNow instance

Synchronize files between the Visual Studio Code workspace (client/local) and the instance (server) bi-directionally.

Before you begin

Create your project and start editing your applications.

Role required: none.

About this task

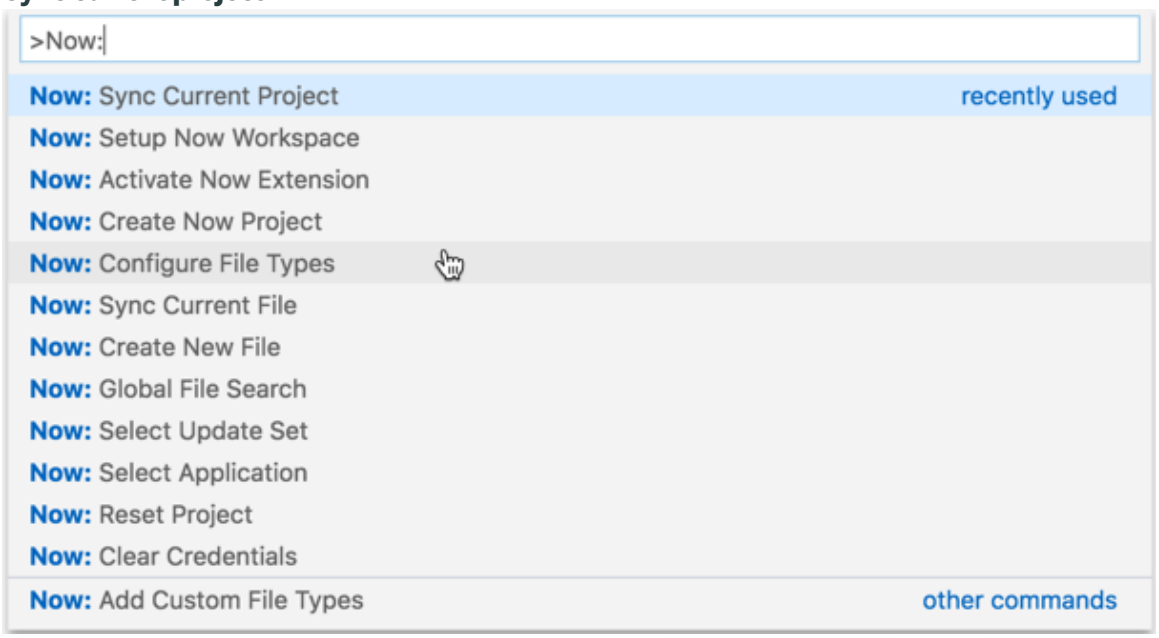
The changes made to the application files in the Visual Studio Code workspace are initially stored in the local file system. When you run the **Sync** command, the local changes made to the application files are sent to the server for persistence.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Click **Sync** on the status bar on the bottom of the Visual Studio Code IDE.
   You can also use a keyboard shortcut, Control+Shift+P on Windows or Command+Shift+P on MacOS, to open the command palette and choose **Now: Sync Current Project** from the list.

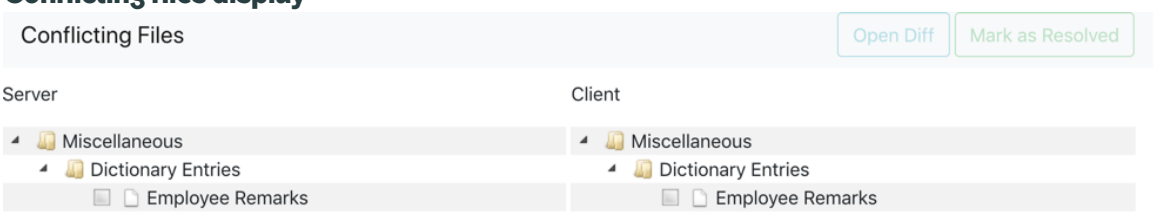
Sync current project



   - If the changes occur either on the server or on the client, the extension synchronizes the file successfully without showing up any errors.
   - If there are no differences between local and server versions, then the message **No changes detected** appears on the bottom of the page.
   - If the same file is modified both on client and server, then the system shows up the conflicts.

2. If a conflict list is displayed, right-click on the conflicting file to open the context menu and select from the context menu.
   Alternatively, select the file and click **Open Diff** menu from the top of the page.

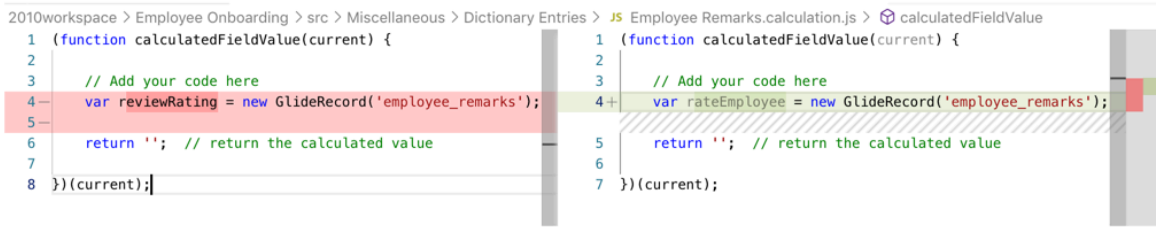
Conflicting files display



   - Choose **Override Server** if you want to overwrite the server version with the client version
   - Choose **Override Local** to overwrite the client version with the server version.
   - Choose **Mark as resolved** to remove the flag indicating a conflict. The server version will be overwritten in the next synchronization.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Differences in client and server versions



3. Retain the desired changes in the client file and save.
4. Once you have resolved all file conflicts, click **Mark as Resolved** under **Conflicting Files**.
   The changes are synchronized on the next sync. When the changes are merged successfully, the message **Sync completed successfully** appears in the bottom of the page.

> **Note:** The files are checked for syntax errors at the beginning of synchronization process. Any errors found are presented in the problem tab at the bottom of the screen.

What to do next

All the changes are tagged to the update set in the status bar. When a project is loaded or selected, the default update set is displayed in the status bar. Click the update set picker to choose the current update set (valid until the IDE or project is closed). If you do not explicitly select an update set, all writes happen to the default update set.

Clear instance credentials in Visual Studio Code

Clear the stored credentials of the current project from the system. User credentials (instance URL, user name, password) of a project are stored in the operating system credentials vault so that the user does not need to log in each time.

Before you begin

Role required: none.

About this task

Once you clear credentials from the current instance, the ServiceNow Extensions for Visual Studio Code extension prompts for authentication the next time you open VS Code. You can continue working in the current session.

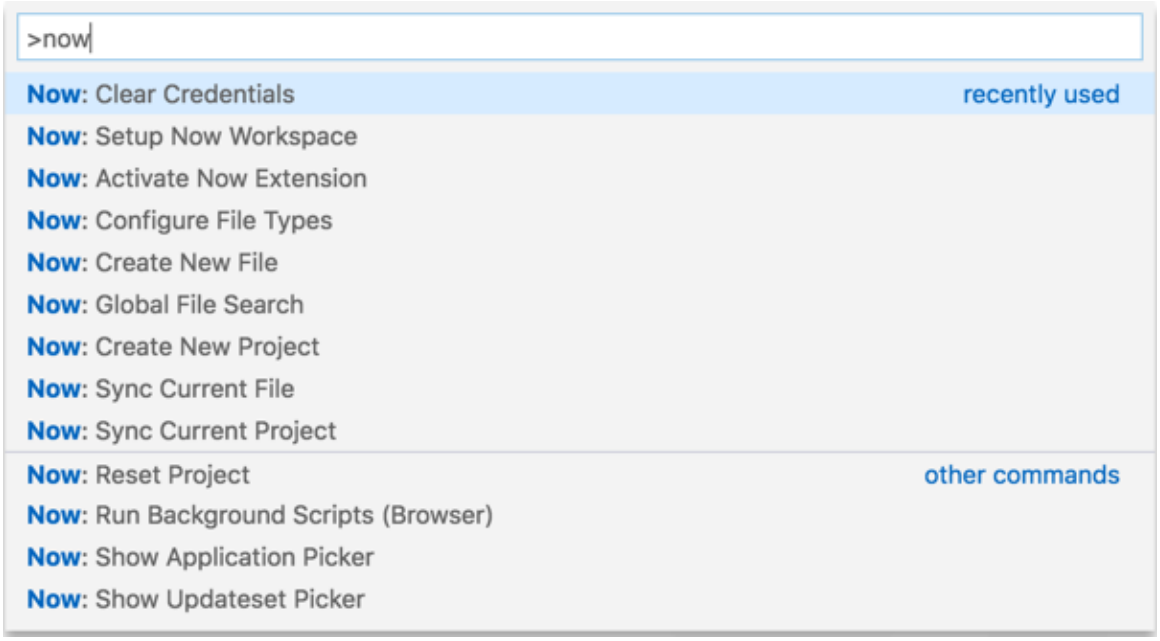
> **Note:** The password is not stored in the Visual Studio Code settings.

Procedure

1. Navigate to **All** > **View** > **Command Palette**.
   You can also use a keyboard shortcut, Control+Shift+P on Windows or Command+Shift+P on MacOS to open the command palette.
2. Choose **Now: Clear Credentials** from the command palette.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

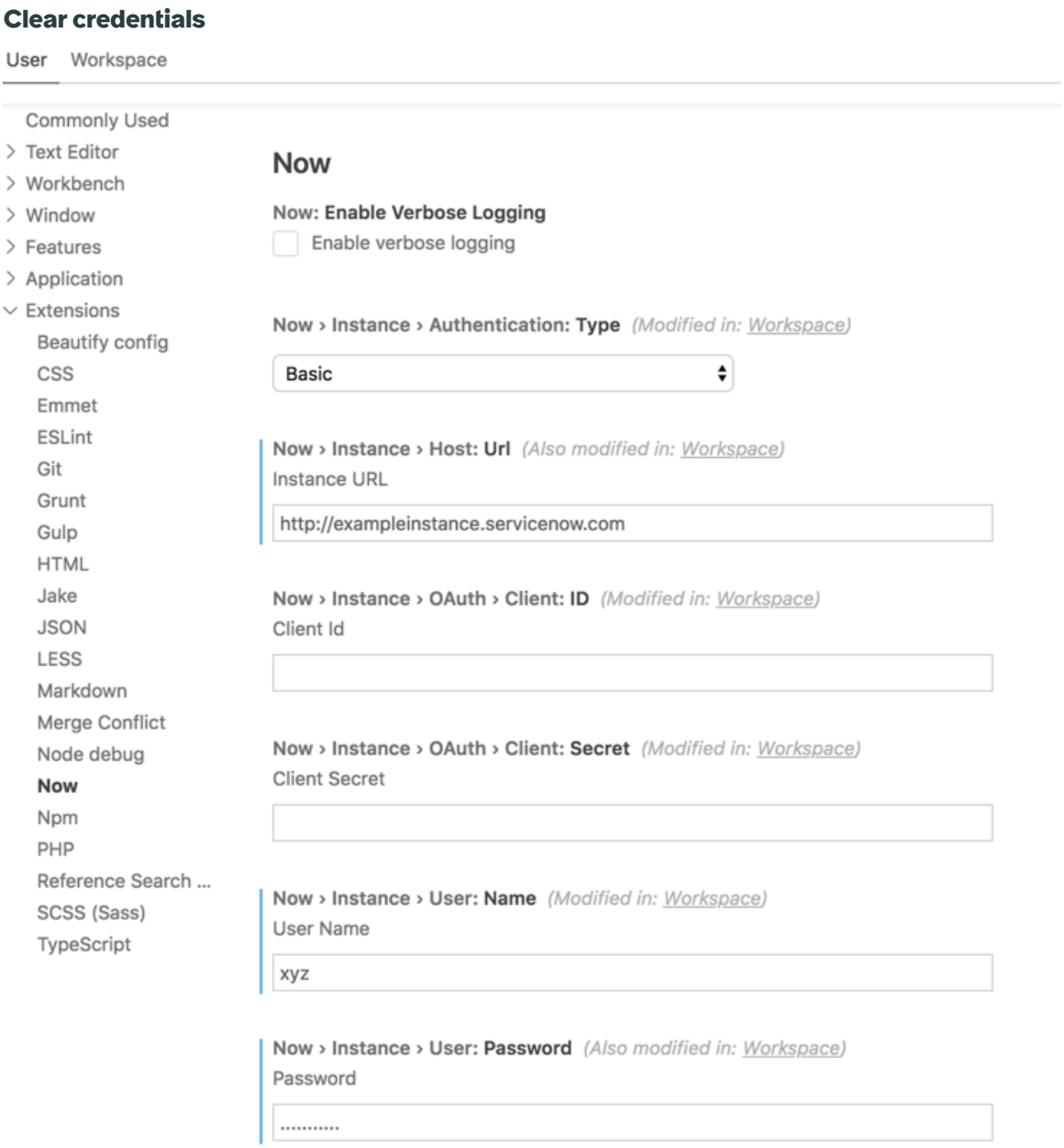
Now: Clear credentials command



The credentials of the current project are cleared.

3. **Optional:** Navigate to **Code** > **Preferences** > **Settings** > **Workspace** > **Extensions** > **ServiceNow** to check whether the credentials and the instance URL have been deleted.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Note: To re-enter your credentials, navigate to **Code Preferences Settings Workspace Extensions Now**. Enter your credentials in **Username** and **Password** fields.

Reset a project in Visual Studio Code

Reset the project to the state on the server by discarding all the local changes if you encounter any serious sync issues.

Before you begin

Role required: none.

About this task

When you reset a project to its server state, all unsynchronized changes are lost.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **View** > **Command Palette** in.
   You can also use a keyboard shortcut, Control+Shift+P on Windows or Command+Shift+P on MacOS, to open the command palette.
2. Choose **Now: Reset Project** from the drop-down list.
3. Click **Yes** in the dialog box.

Result

The project is reset to its original state.

Note: This option should be used only when something goes wrong, for example, if the project is corrupted or you are facing sync issues.

Related topics

Create a project in VS Code

Add custom file types in Visual Studio Code

If you have work with file types other than the default types provided, you can add additional file types to your instance and edit them in ServiceNow extension for Visual Studio Code.

Before you begin

Role required: admin

Procedure

1. Create a new file type in your ServiceNow instance and inherit it from the application file.
2. Add custom columns of the type *`Script`* and *`String`*, and enter some data.
3. Navigate to **View** > **Command Palette** in Visual Studio Code.
   You can also use a keyboard shortcut, Control+Shift+P on Windows or Command+Shift+P on MacOS, to open the command palette.
4. Choose **Now: Add Custom File Types** from the command palette.
   The command fetches all the existing tables in the current application and opens the **Add new file type** wizard.
5. Select the file type you created in your ServiceNow instance from the list.
6. At the Select tags prompt, select the tags you created in the instance.

   All tags are selected by default. Click a tag's check box to deselect it.

Select tags dialog box



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The new file type is successfully added to the `app.config.json` file. You can also add the desired file types manually to the `app.config.json` file under your project. See the example below.

7. Absorb this new file type to the current project using the **Now: Configure File Types** command from the command palette.
8. Select the file type you created from the list of file types and click **OK**.
   The selected file type is ready for editing in the ServiceNow Extensions for Visual Studio Code extension.

Example:
The following is an example for app.config.json

```
{
        "CustomFileTypes": {
        "sc_ic_aprvl_type_defn": {
            "superCoverName": "Miscellaneous",
            "coverName": "Approval Type Definition",
            "tags": {
                "approver_script": "js",
                "approver_html”: "html",
            }
        }
    }
}
```

- superCoverName is a name of the super parent directory (Should always pointtoMiscellaneous).
- coverName is a descriptive name of the table.
- sc_ic_aprvl_type_defn is a table identifier.
- tags represent the set of scriptable columns in the table
- approver_script is a name of the table column / xml tag
- js is an extension of the file (js | html | css | json)

Create a file in VS Code

Use the ServiceNow extension to create a new file for your application with your project.

Before you begin
Import an application in your VS Code project to create a file within your project.

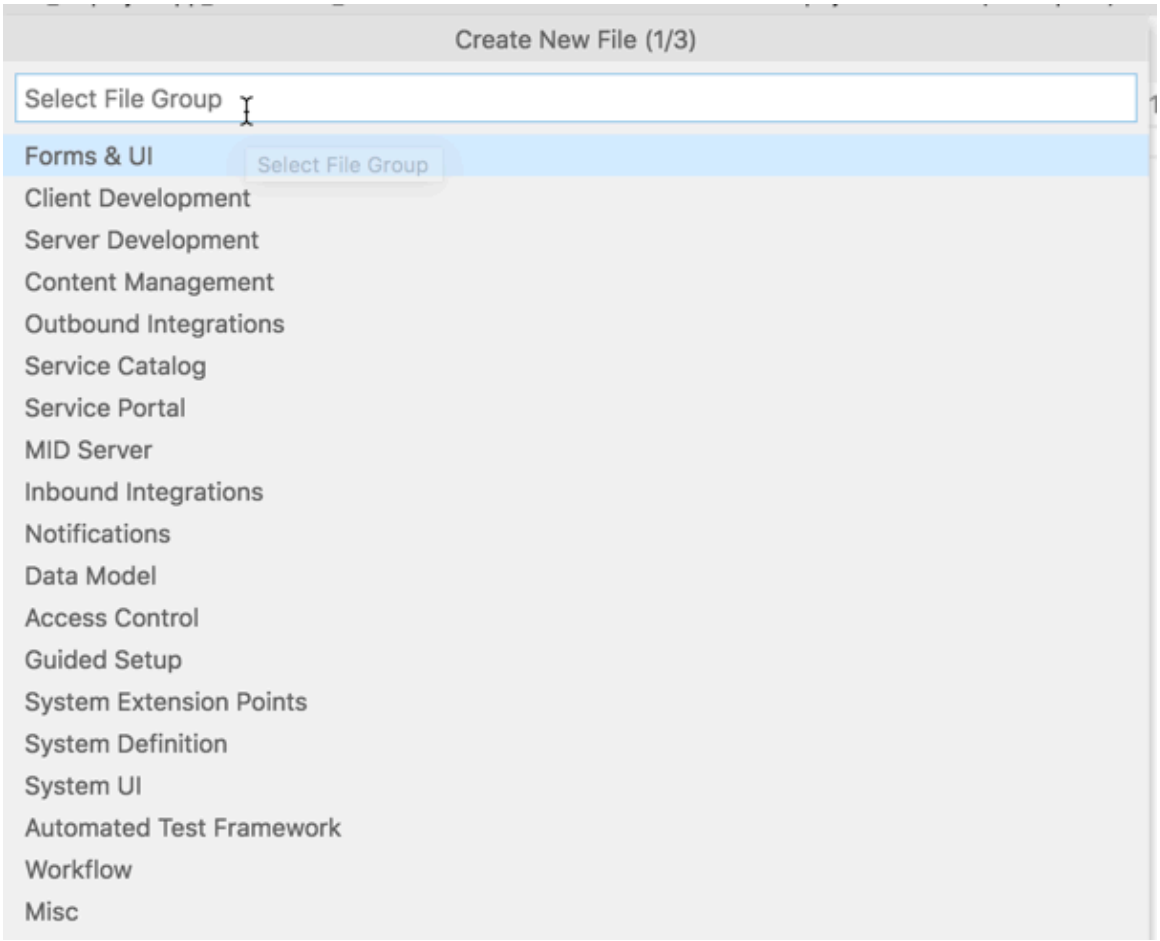
Role required: none.

Procedure

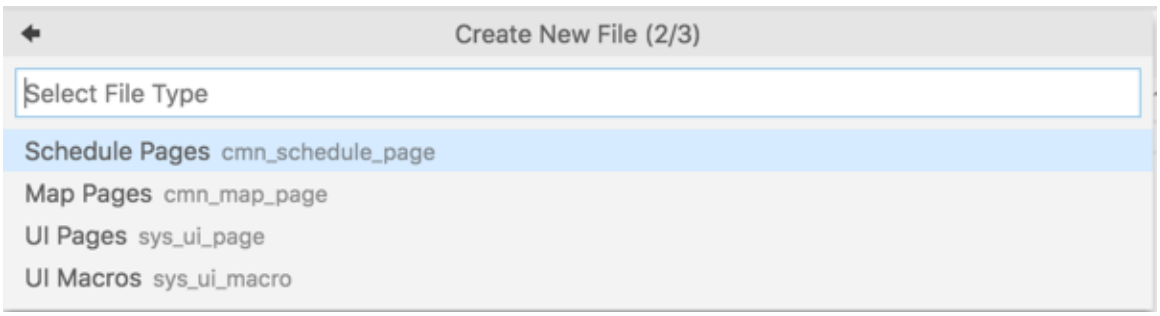
1. Open the VS Code project containing your application.
2. Click the **Create File** (

) on the status bar on the bottom of the VSCode IDE.
   You can also use a keyboard shortcut, Control+Shift+P on Windows or Command+Shift+P on MacOS, to open the command palette and choose **Now: Create New File** from the list.
3. Select the file group from Select File Group list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



4. Select the file type from the **Select File Type** list from the command palette at the top of the screen.



5. Enter a name for the file in the Create New File dialog box.
   The extension creates a file in your application on your local file system. The new file is added to your instance when you synchronize your workspace.

Search files on your instance in VS Code

Search and download any script files on your instance using the ServiceNow Extensions for VS Code extension.

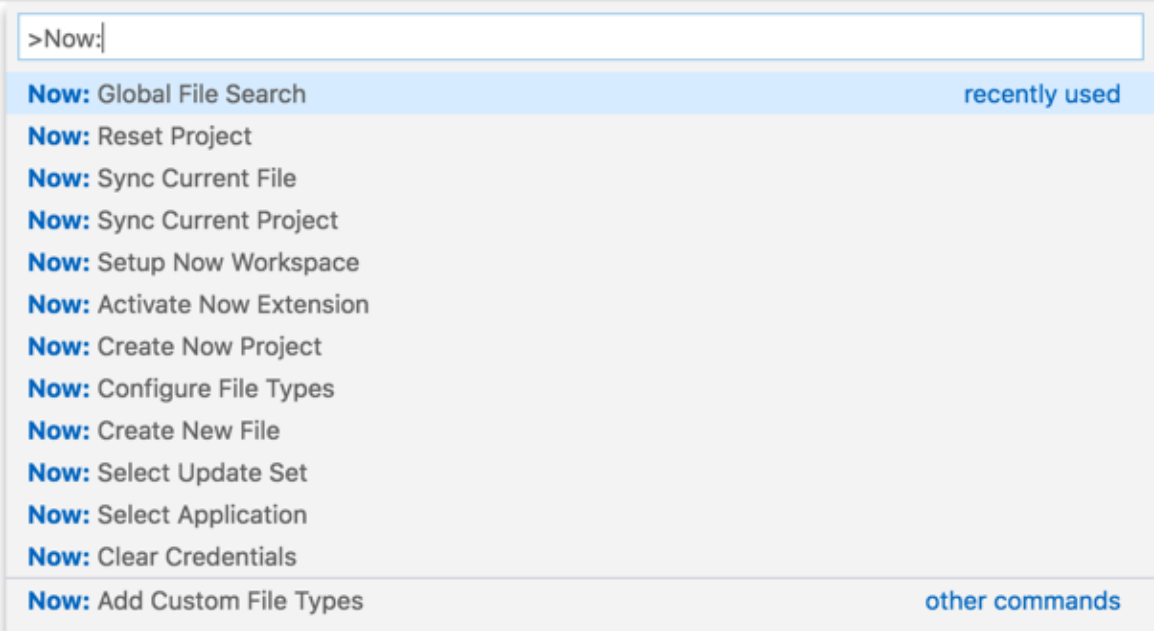
Before you begin
Role required: none

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Open a project in VS Code containing your application.
2. Click the **File Search** on the status bar on the bottom of the VSCode IDE.
   You can also use a keyboard shortcut, Control+Shift+P in Windows or Command+Shift+P on MacOS, to open the command palette and choose **Now: Global Search** from the list.

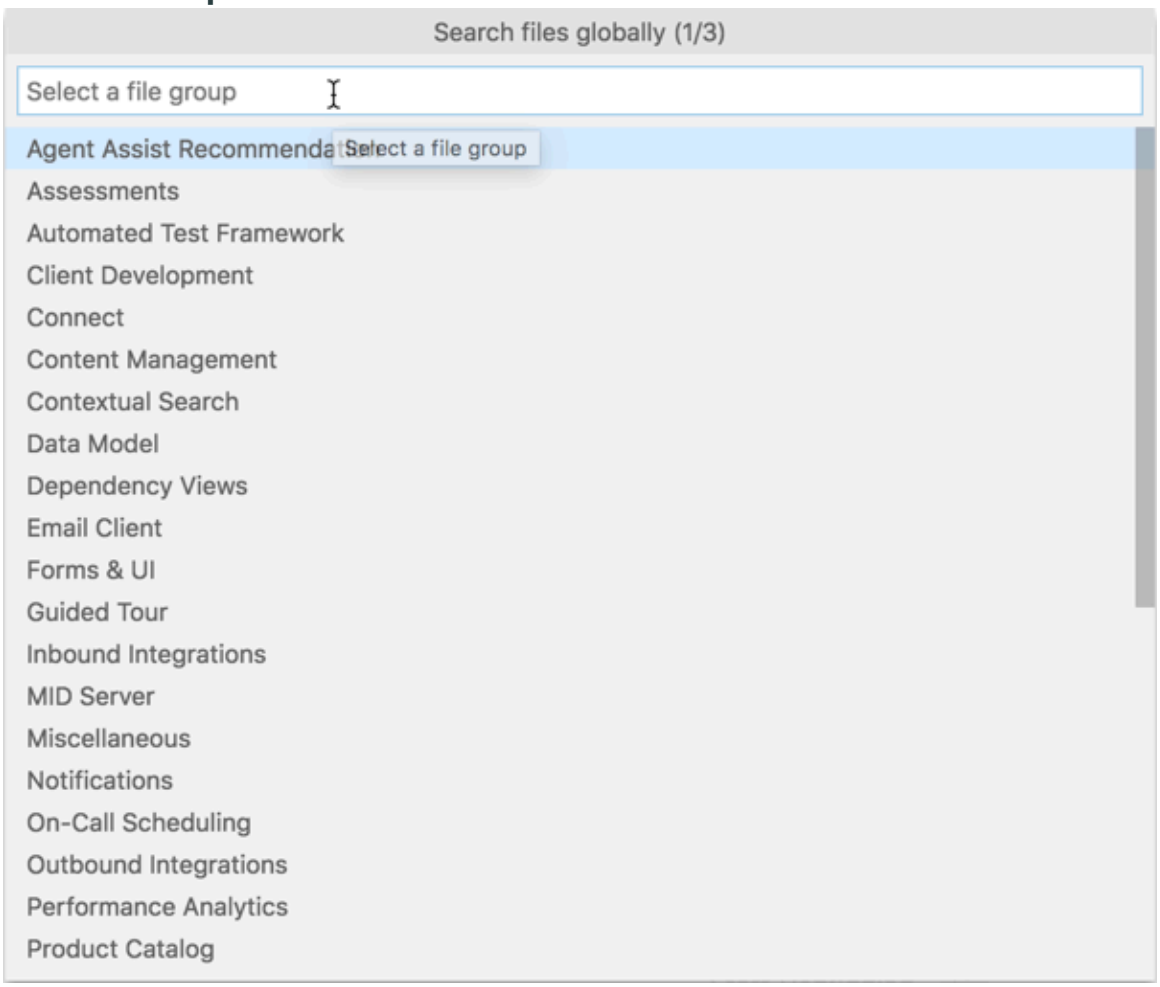
Global File Search command



3. Select the file group from the **Select File Group** list in the command palette on the top of the screen.

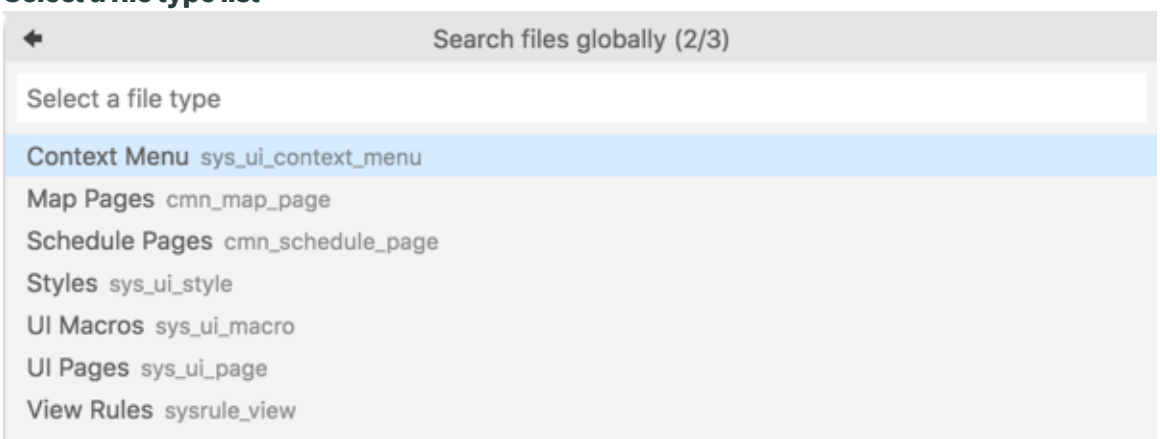
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Select File Group list



4. Select the file type from the Select a File Type list in the command palette on the top of the screen.

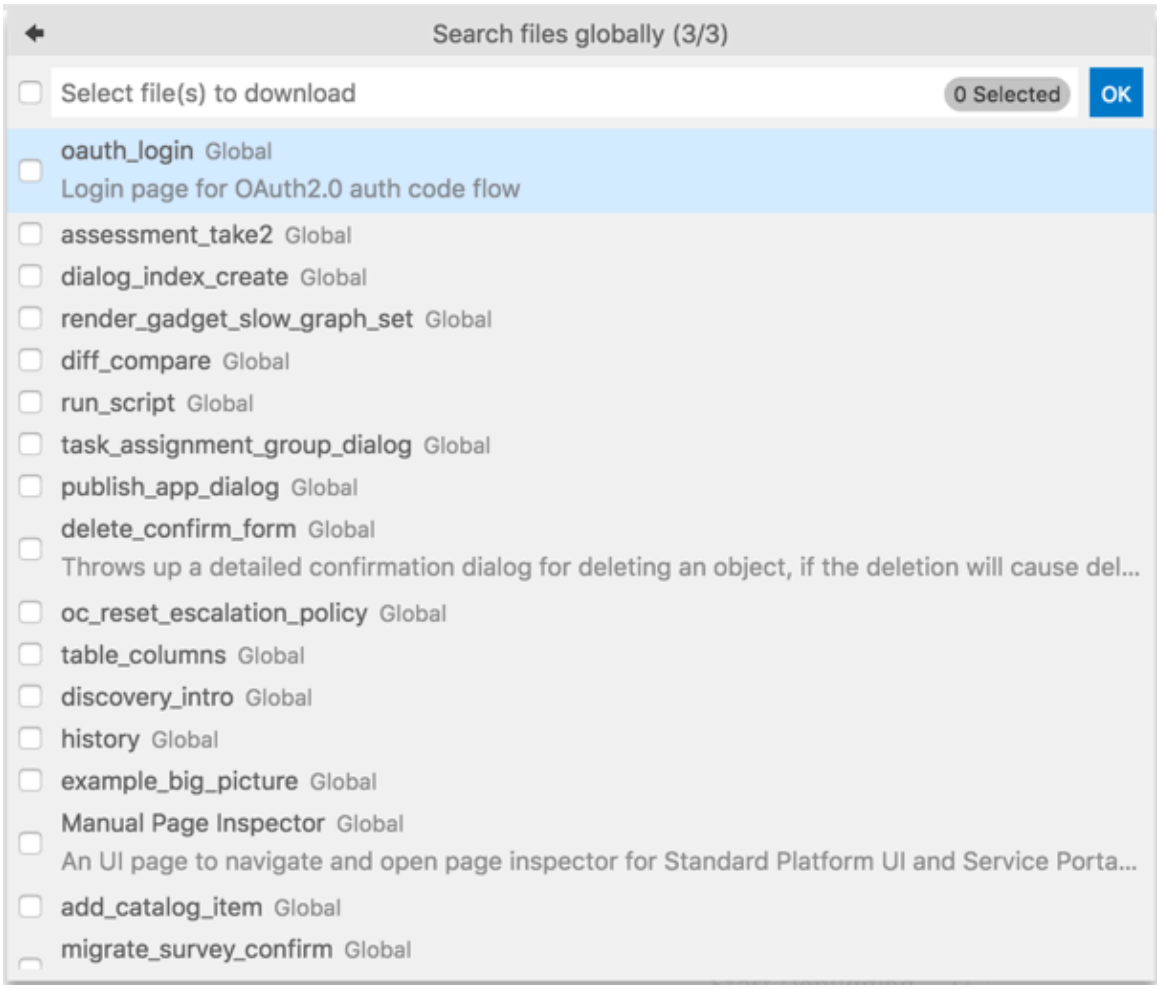
Select a file type list



5. Select files from the **Select files to download** list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Select files to download list



You can select and download multiple files at a time.

The selected files are downloaded to the `scratch` folder of your application. You can edit the selected files.

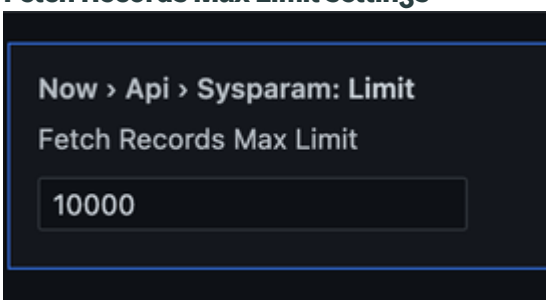
6. View more than 10,000 results from your search:

   a. Open settings by selecting Control+, in Windows or Command+, on MacOS.

   b. Enter `now` into the search bar.

   c. Enter the number of records you want to see in the Fetch Records Max Limit settings.

Fetch Records Max Limit settings



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Related topics

Create a file in VS Code

Run background scripts using VS Code

Run scripts from your ServiceNow instance in VS Code using the ServiceNow Extensions for VS Code.

Before you begin

Role required: admin

Procedure

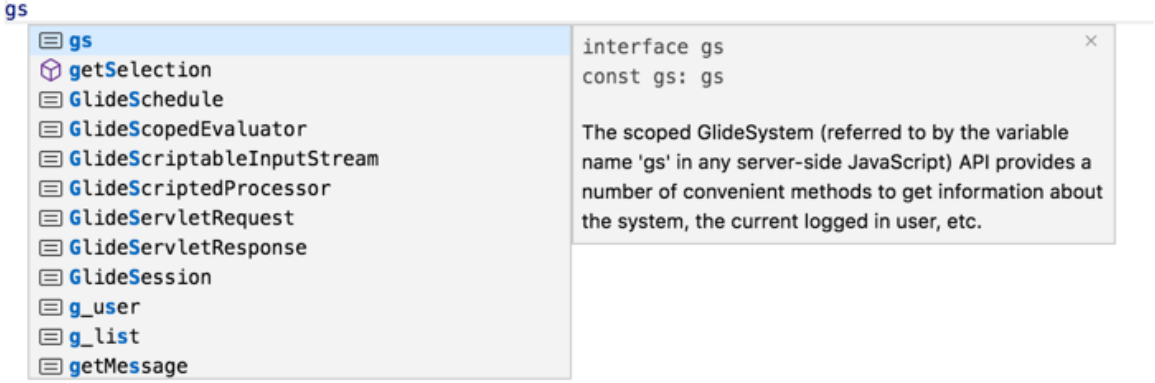
1. While writing the script in the VS Code IDE, select the script in the text file and right-click on the script.
2. Select **Run Background Script- Current** or **Run Background Script- Global** from the context menu.
   The background scripts runner tab opens in VS Code and shows the location where the script is executed in the server.

IntelliSense in VS Code

The IntelliSense code editing feature in the ServiceNow Extensions for VS Code extension provides intelligent code completions which provide auto-completion options to make writing code faster and more efficient.

When you start scripting in the VS Code, the IntelliSense feature shows auto-completion options suitable for your code context at the top of the list.

IntelliSense for GlideSystem



The auto-completion options available are:

- Auto-completion of methods and properties for client and server scripts.
- Auto-completion of methods and properties for standard JS APIs
- Auto-completion of Jelly tags and attributes.
- Code completion in commonly used standards.

The extension supports snippets (templates) for commonly used code patterns and loads snippets defined by the user in Macros table of the Syntax Editor on an instance. Snippets appear in IntelliSense when you type `^Space` along with other suggestions. Alternatively, snippets can also be inserted using the **Insert Snippet** command from the command palette in VS Code.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Telemetry in ServiceNow Extensions for VS Code

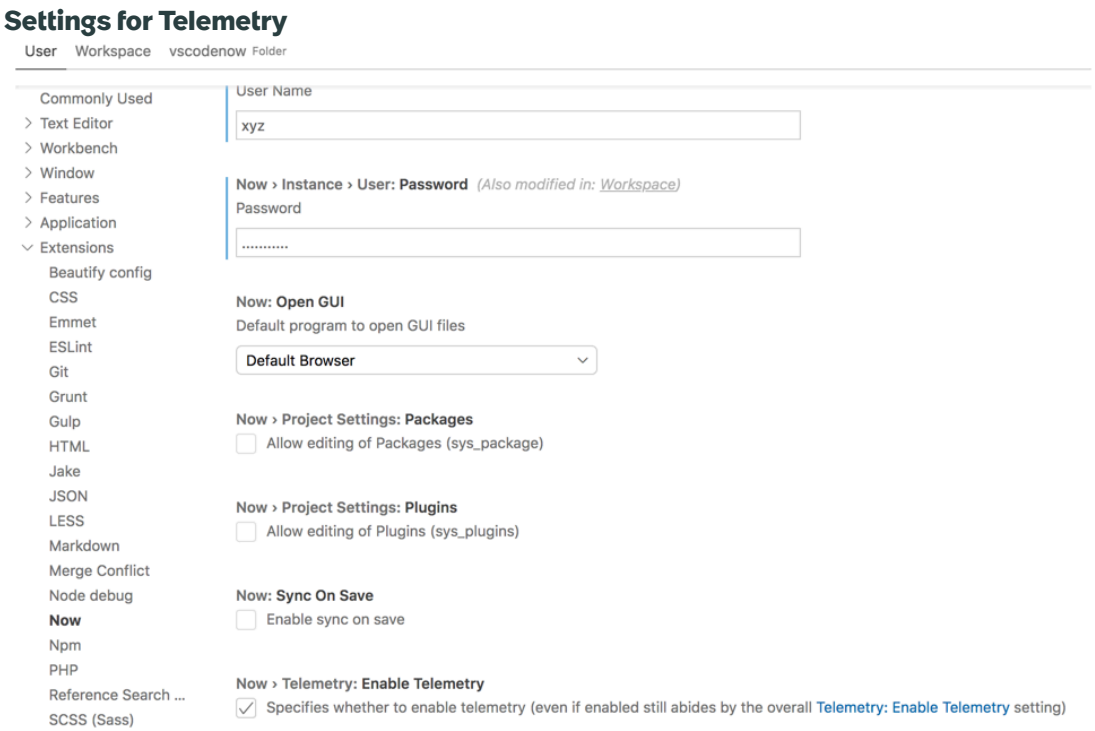
ServiceNow Extensions for VS Code collects information on its various usage patterns such as the commands run with the help of Now Telemetry.

Now Telemetry helps in the following

- Debug issues
- Analyse most used functions or features
- Capture user actions
- Improve the extension

Note: Telemetry collects anonymous information related to usage and does not capture any personal information such as name and email ID of the users. Data is accessible to ServiceNow only and is not shared with anyone

Telemetry is enabled by default for the ServiceNow extension. If you do not wish to send usage data, you can set the `now.telemetry.enableTelemetry` user setting to false. Alternatively, select **File** (Windows) or **Code** (Mac) and navigate to **Preferences** > **Settings** > **Extensions** > **Now** and clear **Now Telemetry** check box to disable the feature.



Note: The usage data will be sent until you disable the setting. The extension abides by the global telemetry setting `telemetry.enableTelemetry`. If that is set to false, Now telemetry is disabled.

Exploring professional development

ServiceNow provides a single mobile and web application development platform to quickly build business applications to power digital transformation. Learn how anyone can automate, extend, and build digital workflow apps across the enterprise with the ServiceNow AI Platform.

This content covers the application development stages to help developers easily build applications on the ServiceNow AI Platform.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Plan your application development: Think through the application before building it.
- Build your application: Create and configure the application and its components.
- Build applications with AI: Create applications with help from agentic AI.
- Validate app functionality: Perform functional testing and write test cases.
- Deploy your app: Move the application into production.

Plan your application development

The planning process is the most important step in building an application. Thinking through design and configuration before building has both immediate and long-term benefits for the application.

Fit for the ServiceNow AI Platform

Not every application idea is a good fit for the ServiceNow AI Platform. During the planning stage, consider what makes an application a good or a poor fit for the ServiceNow AI Platform. Keep the good fit characteristics in mind while planning the application.

Good fit	Poor fit
• Simple forms • Task management • Request management • Spreadsheet-driven processes • Repeatable processes • Third-party integrations • Orchestration of multiple systems • Single experience from functions in multiple systems • Web and mobile access to the same apps and data simultaneously	• Unstructured data • Unrepeatable processes • Content that needs graphics processing • Streaming audio or video • Highly customized user interface

Plan before you build

Essentially, an application is a digital program that supports user tasks. Some actions you take when building an application might be irreversible. Be aware of these actions and plan for them in advance.

Agentic AI

Create applications with help from agentic AI. For more information, see Build applications with AI.

Application scope

One of the first major decisions to make when creating an application is: Should the application be in a private scope or a global scope?

By default, applications are created in their own private application scope. Applications in a private application scope restrict access to their application artifacts so only application artifacts

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

in the same scope have full access to create, modify, remove, or run application data. Scoped applications can use source control integration and delegated development. Globally scoped applications cannot use delegated development.

Create custom business applications in scope unless:

- The application has to delete global data.
- The application needs to change application access settings on multiple default tables to function.
- The application needs to access APIs only available in the global scope. Creating a globally scoped passthrough script include would not be enough for this requirement.

Note: Globally scoped passthrough is a script include created in a global scope that is accessible from the private scope. The passthrough gives access to a global API that is not accessible by default from a private application scope.

For more information, see Application scope and Understanding Application Scope on the ServiceNow AI Platform (Whitepaper).

Instance selection

Proof of Concept (PoC) application builds can and should be built in a separate instance from a regular development instance. The instance can be a sandbox instance or a Personal Developer Instance (PDI) from the Developer Site. The PDI naming format is **dev12345.service-now.com**.

If using an instance with a different scope namespace, rebuild the PoC applications in the organization's development instance. Do not import the applications into the organization's development instance. The scoped namespace for the applications will not match the scoped namespace for the company's development instance.

Applications the organization intends to use (i.e. production apps) should be created in the organization's developer instance, so the application can follow the organization's testing and deployment process.

Naming decisions

The application name matters. ServiceNow suggests a scope based on the application's name. Application file names are appended to the scope to uniquely identify application resources in an instance. Scope is in the format: x_[company code]_[application_name] with a maximum of 18 characters. For example, an application name **Legal Request** has a suggested scope of **x_acme_legal_reque**.

All application files within the application inherit the scope, so carefully consider what the value should be. The application name can always be changed.

Manage app development

Govern app development on the ServiceNow AI Platform.

Source control

To enable simple, future proof collaboration for applications, use source control integration. Source control allows an organization to adopt industry standard tool sets while following modern application development and team management paradigms.

Source control platforms, such as GitLab and GitHub, align an organization with current and future best practice trends in ServiceNow application development and management. However, some organizations may not be ready to adopt the current source control integration.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Branch merging is not currently supported.
- Corporate Git repositories behind a firewall cannot be directly integrated without additional firewall configuration.

Delegated development

Delegated Development allows designated users without the admin role to develop or deploy applications on the ServiceNow AI Platform. Users with the application-specific admin role or the system-level admin role can delegate application development to designated developers at the application level. Delegated developer roles have fewer privileges than admin roles but are still allowed to develop applications.

Note: Delegated development is only available to privately scoped applications.

For more information about building applications, see Build My First Application.

Build your application

Build and define the data, design elements, and logic that make up your application.

Agentic AI

Create applications with help from agentic AI. For more information, see .

Define and build the data model

After planning is complete, define and build the data model. Create one or more tables with fields, load the table with demo data, and verify access controls to the data.

Agentic AI

Create applications with help from agentic AI. For more information, see Build applications with AI.

Create an application

Create or open an application record.

- Create: If creating an application directly, use Guided App Creator to create the application. Guided App Creater allows you to create tables from spreadsheets, roles, and security rules. Design options are also available.
- Open: If developers do not have the admin role, the ServiceNow System Administrator needs to create the application and grant developers a delegated development role. Developers then use Studio to open the application for editing.

Note: The application scope and table name are sometimes referred to as the internal names for these objects and cannot be changed once they are created. However, the application name and table label can be changed. Application users see the internal names in the URL only. If possible, internal names should always be consistent with what the user sees.

Build the data model

Create tables and fields on the tables to support the application's data model.

Note: Consider creating applications with help from agentic AI. For more information, see Build applications with AI.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ServiceNow automatically adds five fields to each new table. The new fields contain auto-populated information about the table.

Fields added to every table

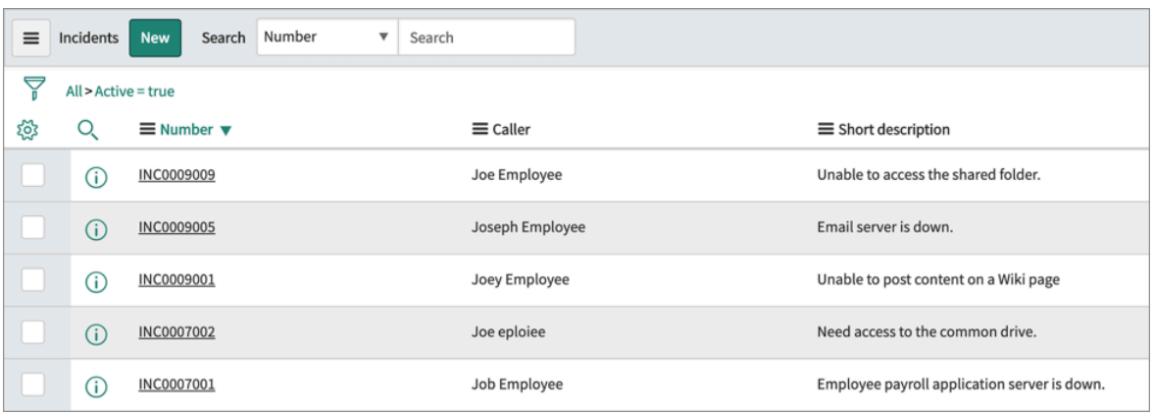
Field name	Database name	Description
Created by	sys_created_by	User who created the record.
Created	sys_created_on	Date/time when the record was created.
Updated by	sys_updated_by	User who last updated the record.
Sys ID	sys_id	Unique identifier for the record. It is unique throughout the instance.
Updates	sys_mod_count	Numeric field that counts the number of updates to the record since record creation.

New tables can extend an existing table to inherit fields and functionality from the table being extended. Add to and modify the components of the extended table. The most commonly extended ServiceNow table is the task table. For more information, see When to create a new table vs. when to extend and Exploring ServiceNow AI Platform® tables.

Add fields to the table to support the data model required by the application. ServiceNow has many different field types with built-in validation. Select the field type that best fits the field's data.

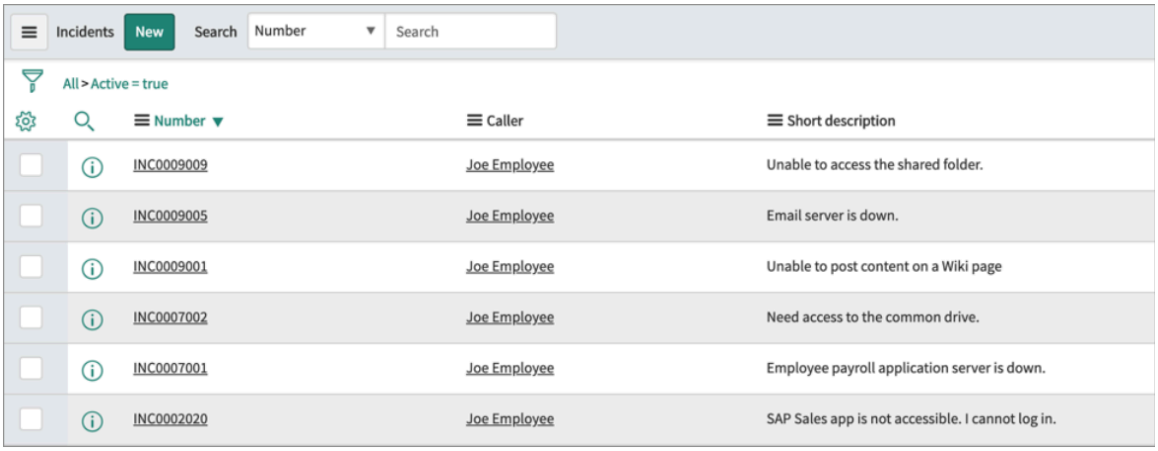
Note: String (plain text) fields are the easiest to configure. However, because users can enter anything, string fields can result in bad and inconsistent data that is difficult to use.

In the example, a string field type is used for a user's name. Notice the Caller field is different for each Incident record, but the caller may be the same person. Do not use a string field type for a user's name in tables.



Instead, use a reference field type that references the User table instead of a String field. Users then need to select a single consistent record in the Caller field.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Reference fields ensure consistent data by normalizing date in another table in ServiceNow. ServiceNow has over 2000 baseline tables available to reference. The Appendix lists some commonly used tables for building an app.

While a reference field can normalize data, other fields can be used for specific types of data. Some common field types are:

Field type	Descriptions
Integer	Stores number values and can be used in calculations.
Currency	Holds a currency value and will show values in the currency of the logged in user.
Phone number	Includes validation and formatting for E164-compliant phone numbers.
Reference	Displays a record from another table and helps to normalize data.
Choice	Displays a select box with a predefined list of choices. Choice lists should include fewer than ten items.
Date	Stores a date value selected with a date picker. Use Date if you do not need a specific time.
Date/Time	Stores date and time values selected with a date and time picker. Use Date/Time to compare specific times or if the exact time is important.
String	Holds freeform text. Use String if no other field type matches the values stored in the field.

Note: Field types should not be changed after a field is created.

Choice lists or reference fields

Choice lists and Reference fields both offer users a way to choose a value from a list. Choice lists are name/value pairs. Users select from the names and the field stores the value of the selected choice. Scripts use the value. Add and remove name/value pairs from the choices to manage the list of options.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Reference fields point to a table. Manage choices in the table. The value stored in the reference field is the sys_id of the referenced record.

Choice lists do not require a reference table and are easier to configure than reference fields. Use Choice lists when the field has ten or fewer options and the options will not change. Consider using a reference field and table when:

- The field requires more than ten choices.
- The choices will regularly change.
- Someone other than an administrator needs to manage the choices.
- The value of the field has an impact on decision logic. For example, decision tables in Flow Designer.
- The data has multi-level dependencies between different fields that can lead to complex and unwieldy choice field combinations.
- The choices require more than a name/value pair. For example, referencing a user record gives the referencing table access to other user details, such as email and department.
- A table already exists that includes the data needed for the field.

When using reference fields, review the tables available in the instance to reference before creating a table. If creating a new table, check the list of exempt tables in section 2 of the Custom Table Guide. If appropriate, extend the new table from one of these.

Note: Before creating new fields on an extended table, check for an existing field inherited from the base table that has a similar purpose. If a field is found, override the extended table's label.

Secure data

Data security is one of the most important and overlooked aspects of creating an application. ServiceNow automatically configures access control for a new or selected role during the table creation process. Only users with the role can access the table to read, create, write, and delete.

Use access control rules to configure table and column-level security in the ServiceNow AI Platform. To properly configure access to an application, developers should understand how access controls work and the order in which access controls are evaluated. Apply multiple access controls that together make an Access Control List (ACL).

Self-Paced Training: Securing Applications

Documentation: Access control list rules

When considering security:

- Protect tables, UI pages, property pages, and other content with the appropriate access controls and roles.
- Limit the use of GlideRecord queries in access control scripts. GlideRecord queries can affect performance.

Beginning with the Orlando Platform Subscription model, customers are charged by how many tables a user can access, regardless of whether the user does access the table. Configure ACLs to restrict access to a table to ensure that only the users that need access to a table can access the table.

Note: Consider making any auto-populated fields read only. If the system is populating the data, a user should not be able to.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Alternately, secure data on the ServiceNow AI Platform with before-query Business Rules. Before-query Business Rules run before the database query and are limited to controlling read access to a record. Only use before-query Business Rules when necessary. Some considerations when deciding to use Access Controls or before-query Business Rules:

- GlideRecord queries will bypass read access controls on a table and will be restricted by before-query Business Rules on a table.
- When access controls restrict read access to records in a list, ServiceNow shows a message saying that access has been restricted for the records. With before-query Business Rules, the number of records in the list total matches the number of records shown to the user. The user receives no indication that some records have been hidden from the list.

Review the user query Business Rule on the User [sys_user] table for reference.

Note: Before-query Business Rules do not take the place of ACLs. Denying users access to a table via before-query Business Rule will still count the table against the subscription model. Use Access Controls to prevent the table from being counted for the users in the Platform Subscription model.

Encryption

The ServiceNow AI Platform also provides various encryption solutions at the application tier, database tier, and hardware tier. Learn more in the Data Encryption Whitepaper.

Note: Set up security before configuring any interfaces or business logic. Since security affects the data available to interfaces and business logic, waiting until the end of the application build process may cause rework and issues.

Manage data

With the data model (tables and fields) created and security set up, add data into the application's table(s).

Import sets

To populate tables in ServiceNow from an external platform, use import sets and transform maps. To retrieve data from an external source on a regular basis, use a scheduled import.

Self-Paced Training: Importing Data into ServiceNow

Guidelines for Import Set Performance:

- Break up large amounts of data into smaller sets for faster imports. Consider 100,000 rows to be a baseline and break up anything larger than that into sets of 100,000. For example, importing 10 sets of 100,000 will finish quicker than one set of 1,000,000 records. Also consider using Concurrent imports with larger numbers of records.
- Importing large data sets simultaneously can put load on an instance. Stagger large imports so the imports do not overlap.
- If possible, deselect the Run business rules check box on the transform map table to avoid running Business Rules and other logic during an import. Consider using an onComplete transform script to run business logic, such as calculations, at the end of an import rather than on each record as Business Rules do.
- Use default functionality for imports. Whenever possible, avoid writing custom scripts. For example, use the coalesce functionality rather than writing a custom coalesce script.
- Avoid GlideRecord queries in an import set.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Make sure any fields configured to coalesce are indexed.
- If replacing a system with a requirement to import historical data, import only the historical data required for the application. Consider storing historical data in a data lake.

Inbound integrations

In addition to Import Sets, ServiceNow includes APIs to accept data from external platforms.

To push data directly into an application table from another system, use Web Service Import Sets instead of writing directly to application tables using the REST Table API or SOAP APIs.

To handle data transactions that are more complex than writing data to a table, such as sending an attachment or ordering a catalog item, review the available APIs to see if one exists that supports the required logic.

Use a Scripted REST API or Scripted SOAP Web Service to build REST or SOAP endpoints, respectively. Scripted REST API and Scripted SOAP Web Service endpoints can execute ServiceNow server-side code when the endpoint is consumed by an external system.

Self-Paced Training: REST Integrations

Create design elements

After the application's data model is created, secured, and populated with data, create the design elements to access that data.

Note: Consider creating applications and experiences with help from agentic AI. For more information, see Build applications with AI.

Primary interfaces

The primary way for users to interact with a data model is through forms and lists or through mobile.

Forms and lists

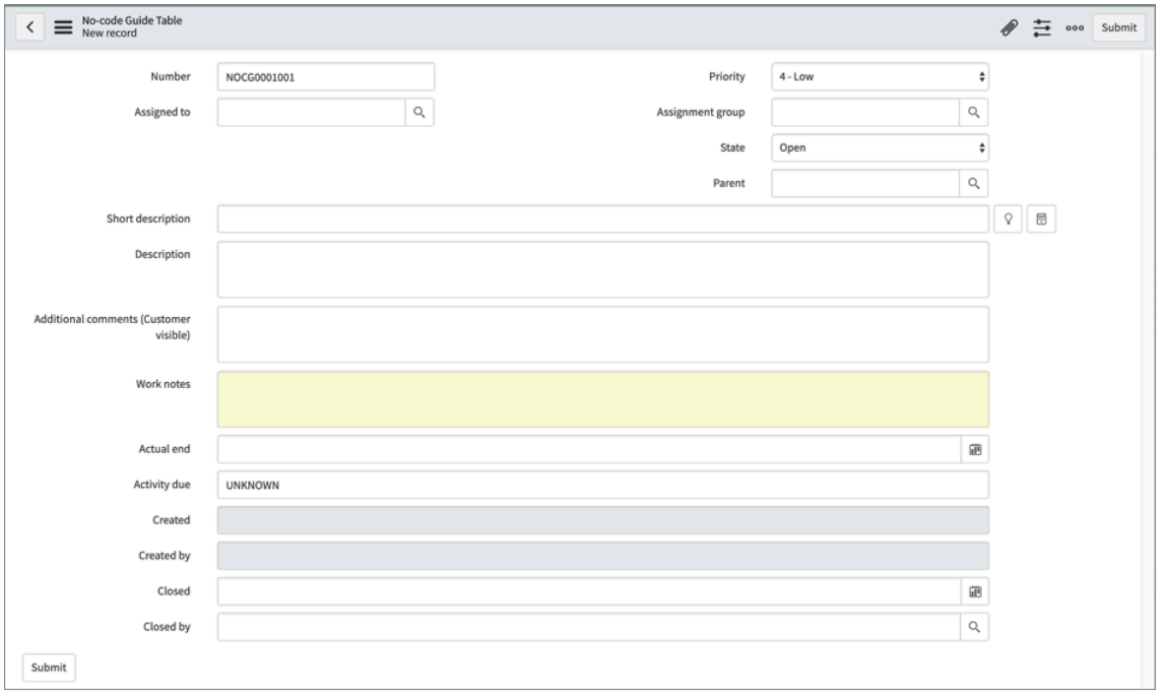
The standard method of accessing data in ServiceNow is through the default forms and lists. A form displays information from one record in a data table and a list displays a set of records from a table. When configuring forms and lists:

1. Keep the number of fields on a form to a minimum. The more fields on a form, the longer the form takes to load resulting in a poor user experience. Use form views to create different sets of fields for different situations.
2. Use form sections to logically group fields together and to keep users from scrolling. The top section of the form should contain the fields that are always needed or used, while the other form sections contain less frequently utilized fields.
3. Make sure fields appear in the right order. For example, the start date field should always come right before an end date field.
4. Use seven or fewer columns in a default list. Users can add more by personalizing their lists.
5. Avoid using a reference field as the first item in the list view, because it is shown as hyperlinked text. Clicking on the reference field will redirect the user to the referenced record instead of the list record, resulting in a poor user experience.

This example shows a poorly designed form.

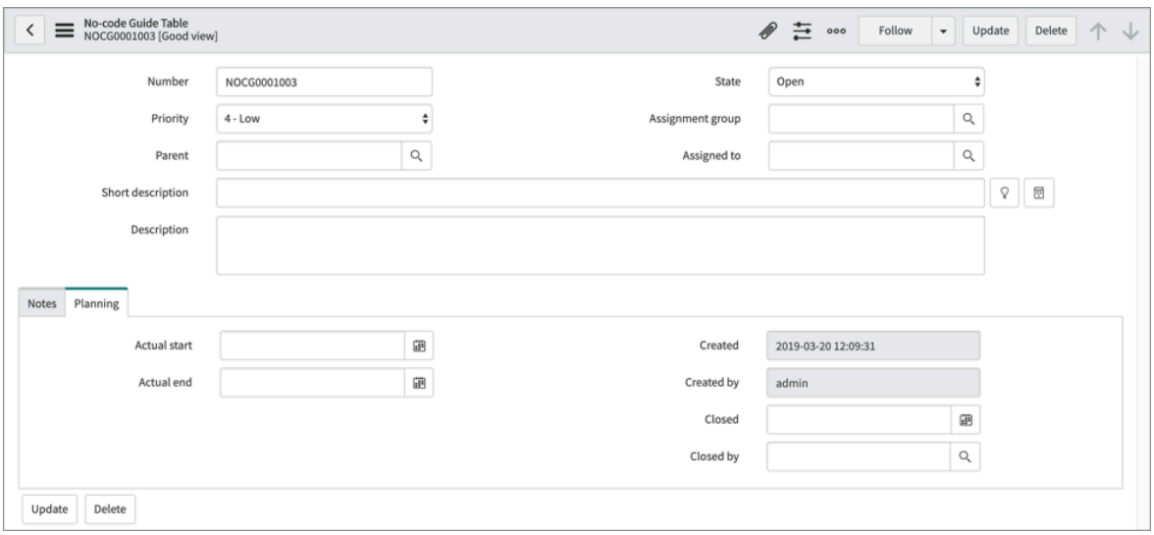
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- The form has no sections. Users need to scroll through the entire form to see all the fields.
- Similar fields are not grouped together. For example, Assignment group and Assigned to are on different sides of the form.



This example shows a well-designed form.

- Fields are grouped together logically.
- The form has been broken into sections for easier viewing and data entry.



Mobile

If users will interact with the application on their mobile devices and will need native iOS or Android functionality, such as geolocation or offline access to application data, use the ServiceNow Agent app. Use Studio to create a mobile application for the application.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Create applets in the mobile application. Applets are tiles within an application. Include a main screen, such as a map or a list, and various details screens, such as an activity stream or related lists. Applets should have focused experiences.

Note: Individual applets can be secured by role as well as made available in offline mode.

Self-paced training: Mobile Applications

Other resources: Mobile Resources

Self service

Your application may need a way for end users to be able to access your data model, so there are self-service options available.

Service Portal

If the application has Requestor or Self-Service users, use Service Portal to provide a friendly web experience.

To give self-service users the ability to easily create application records from the Service Portal, create a record producer. A record producer can provide a better end-user experience than a regular form. Talk to your ServiceNow Administrator about the appropriate catalog and categorization to make the record producer accessible through the Service Portal.

Alternatively, create a Service Portal for your app if the following is true:

The application needs different branding, navigation, or user experience than an organization's current Service Portal.

OR

The organization does not have an existing Service Portal.

AND

The application needs more functionality than the default portals provide.

AND

The application requires a more customized user experience than the default forms and lists can provide.

OR

The application needs more control over branding and themes than the default interface provides.

Tip: Do not try to reuse any existing service portal pages in an application. Create new pages and then reuse components in your pages, such as widgets and headers.

Widgets

Widgets are what define the portal content. The base system widgets provided with Service Portal can be used, or developers can build custom widgets to fit business needs.

Considerations for creating custom Service Portal widgets for an application:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Start from an existing widget instead of creating a widget from scratch. To protect existing widgets from accidental modifications, all baseline widgets are read-only.
- When developing a widget, use the preview pane to quickly test the widget's behavior. Always test the widget on a portal page before releasing a widget to production.
- Use third-party debugging tools when debugging browser-based applications. For example, the ng-inspector Chrome extension for Angular JS.
- Avoid the use of $rootScope.$broadcast(). Instead, use $rootScope.$emit() to publish an event to the rootscope.
- Use widget options to make widgets more easily reusable. The widget option schema defines the user-configurable fields.
- For field types not supported in the option schema, create an extension table to store a custom widget option schema.
- Make use of Angular Providers, which are reusable components that can be injected into multiple widgets. To ensure quick loading widgets and a high performing portal, create Angular Providers instead of overloading your client controllers with persistent data and additional logic. With Angular Providers, you can maintain data for the lifetime of your Service Portal and reuse components and data objects across multiple widgets.

For more information, see Service Portal training.

Virtual Agent as an application design element

Consider Virtual Agent, ServiceNow's conversational bot platform. Use ServiceNow Virtual Agent (VA) to build and design bot conversations to help users quickly obtain information, make decisions, and perform common work tasks.

Some considerations when building VA conversation topics:

- If adding VA to a portal, create topics that focus on particular use cases rather than a broad topic.
- Read topics out loud to keep the topic from sounding too robotic or being too verbose.
- Do not try to design the whole interaction at once. Instead take an iterative approach when building the bot and test the conversation frequently.

For more information, see Virtual Agent.

Notifications

After the data model is defined and users are able to interact with the application, determine how the application should communicate with users. Configure notifications to alert users to application-related events, share information in the knowledge base, and add translations so users can interact with the application in their native language.

The ServiceNow AI Platform can notify users by email, SMS text message, or push notifications.

Triggers: Event Driven Notifications vs Table Updates

Configure notifications to trigger based on updates to a table or based on an event. Use event-based triggering when triggering requirements cannot be easily implemented in the notification conditions or when the notification is triggered from a Workflow. Creating specific events also enables easier notification debugging.

Use the Notification activity in a workflow or the Send an email action in Flow Designer for simple notification use cases. For complex or critical notifications, trigger an event from a workflow or a Flow Designer flow and configure a notification to fire off that event.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Notification Content

ServiceNow notifications support static and dynamic content using email templates, email scripts, and notification variables.

Notification variables add dynamic information to the body of a notification, such as field values from the base record. The variables also support dot-walking, which allows field values from any related records to be included in the content without scripting.

For example, use the URI_REF variable to point to the record that originated the email.

Use Scripting for email notifications or dot-walk from the base record to include dynamic content that is not available in the record. Use the mail script API to set notification details, such as the recipient and sender addresses, etc.

Note: Scripts entered in the notification body using <mail_script> tags may not always work correctly. Always create email scripts at System Notifications > Email > Notification Email Script and include them in the notification body with ${mail_script:script_name}.

Create Email templates for content used in multiple notifications. Adding the content to an email template enables administrators to create reusable content for the subject line and message body of email notifications.

Configure recipients

ServiceNow emails can be sent to users, groups, or individual email addresses. When sending to groups, check the **Include members** field on the group record for the notification to be sent to the group email and all members individually.

Subscription-based notifications - Select the **Subscribable** option on the notification to allow recipients to pick and choose the emails they want to receive.

For a subscription-based notification, the Mandatory option can be set to true for the recipients to receive the notification regardless of their individual preferences. Optionally, configure Email unsubscribe links in the outgoing email to allow users to remove themselves from the notification.

ServiceNow uses email watermarks to correctly process user responses to notifications. Email notifications automatically include watermarks unless the **Omit watermark** option is selected in the notification. Omit watermarks only when no email response is expected from the recipients.

The Troubleshooting Outbound Email knowledge base article provides troubleshooting steps for the most common notification issues. Log in to the HI portal (https://hi.service-now.com) to access the article.

Translations

When using one of the Internationalization plugins, most of the fields in the instance are automatically translated. However, customizations are not translated automatically, and need to be translated by hand. In this case, it is best to locate the individual untranslated strings, and insert those translations manually.

To implement a new language, activate the plugin for the new language. Then export the existing values from any of the Translation tables.

For example, export all **sys_choice** records. Provide the export to a translator to provide translations for the **Label** field in the provided file. The **Value** field should not change. Update the **Language** value to the new language. Then, upload the translated values back to the **sys_choice** table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: To insert values, make sure you are setting all the necessary fields; table, element, language, label, value, and sequence. Set the dependent value and hint where applicable. This logic is the same for all translation tables.

For more information, see Language Internationalization Support

Overview of reporting and analytics for developers

Most applications that you create have some level of reporting requirements. Reports should be actionable to drive change.

"Reporting" generally refers to showing the data inside facts tables like Incident [incident]. You can also create key performance indicators (KPIs) to track changes in this data over time, through the Performance Analytics application.

Currently the ServiceNow AI Platform® is in transition between two user interfaces for showing this information:

- The older Core UI technology. This UI includes the Reporting application, which only shows data directly from tables, and PA Widgets, which show data from Performance Analytics indicators. Both reports and PA widgets can be placed on Core UI responsive dashboards. For more information, see Reporting, dashboards, and Performance Analytics in the Core UI.
- The newer Platform Analytics technology. This UI includes Data Visualizations, which let you report on data from any source. These visualizations can be placed on Platform Analytics dashboards, along with Platform Analytics filters. For more information, see Platform Analytics experience.

  All Platform Analytics objects are rooted in the Next Experience UI Framework and are available to developers. However, a non-developer can still build their own objects through the Platform Analytics experience, without using UI Builder. For more information, see Platform Analytics experience.

Note: Although "reporting" is a general term, this documentation usually uses "report" and "reporting" to refer to the Core UI Reporting application and "visualization" or "visualize" to refer to Platform Analytics data visualizations, for disambiguation.

Follow these guidelines when building reports or data visualizations:

- Creating reports or visualizations on large tables can negatively impact performance. Be sure to filter by a date range or other limiting criteria rather than showing all records on the table.
- Grouping by fields that contain many possible values can negatively impact performance.
- If loading a report or visualization gives a Long running transaction timer message, consider adding more data filters to reduce the load time.
- If you need to export a report, data visualization, or dashboard on a regular basis, schedule the export and email.

In Platform Analytics, you have several possibilities for showing multiple data visualizations on one page:

- Create the data visualizations and the dashboards entirely inside the Platform Analytics experience. This approach does not require a developer role or special technical knowledge, and probably should be explored before you try a more complex solution.
- Create data visualization components in a generic UI Builder page, along with filters and other components. This approach gives you the most freedom as a developer, but also requires the most configuration.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Create the data visualizations and the dashboards inside the Platform Analytics experience, but then place the dashboards inside UI Builder pages using the Dashboard page template. This approach lets you use the convenience and special features of the Platform Analytics experience to create dashboards, data visualizations, and filters in your own experiences/ workspaces. This approach also gives you the freedom to customize the page configuration partially. For more information, see Creating Platform Analytics pages in your own workspace.
- Create a technical dashboard and populate it inside UI Builder. This approach is almost the same as creating your own UI Builder page from scratch, but the dashboard is available in the dashboard library, has dashboard details, and can be shared like other dashboards. For more information, see Technical dashboards.

Build form and business logic

The next step in designing an application is to build logic. Logic includes form logic (what people can and cannot see/use on a form) and business logic (rules that govern what happens to data when it is entered).

Agentic AI

Create applications with help from agentic AI. For more information, see Build applications with AI.

Scripting and modifications

Before writing any code, be aware of the impact on upgrades and the adoption of new ServiceNow features. Particular care should be taken when modifying baseline artifacts and processes.

Consider the following before scripting:

- Assess the requirement. Is the logic critical to the functioning of the app?
- Determine if ServiceNow can be configured to fulfill the requirement without code.
- Leverage options, such as Flow Designer, Virtual Agent, and UI Policies to take advantage of platform capabilities without writing code.
- Low and no-code approaches to logic are easier to debug and upgrade.

Examples of when scripting is appropriate:

- Building Flow Designer actions
- Creating a Scripted REST API
- Creating logic for scoped applications in Script Includes
- Customizing and creating widgets for Service Portal

Evaluate the business requirement and consider a no-code route before using a scripted solution.

Be aware of ServiceNow enhancements. For example, in the orlando release, Virtual Agent conversations have more no-code options than London. Read release notes and other publications. Get certified and stay current with your certifications.

To better understand when to customize, review the Innovate at Scale Success Playbook on the Customer Success Center.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Modifying default behavior

In the past, one of the strategies used was to copy the artifact to update and to deactivate the original. The copy/deactivate approach is no longer recommended due to the following issues:

- Developers cannot tell if a deactivated artifact was upgraded without research.
- Two files, the original and the copy, need to be maintained. Maintenance doubles each time a customization is made.
- With each release, the customized record becomes older.
  - Customers do not receive the enhancements included in a new release.
  - A new release may rely on the original record being updated.
  - Developers may make more changes to compensate for the original record being inactive.

A script where only the **Active** flag is changed will be updated, but the script does not appear on the skipped list. With the copy and deactivate strategy, a developer has less visibility into customizations and cannot easily assess or revert to the baseline version.

Rather than copying and deactivating the original artifact, edit the artifact directly. The ServiceNow Upgrade Engine will add the latest version to the version history and report that the artifact was skipped. Developers can see a new version is available with the upgrade.

Form logic

Controlling what users see when they visit a form can increase productivity and responsiveness. For example, users should only see fields that are useful to them. Users may only need to see certain fields based on what is configured on the form. Apply form logic to control what is visible, read-only, and mandatory on a form.

Agentic AI

Create applications with help from agentic AI. For more information, see Build applications with AI.

Determine when to control user access

The following question will help direct you to the right decision for when to control user access to information: Is this a suggestion or enforcement? A suggestion makes the form easier to complete whereas enforcement forces the user to do something in order to complete the form.

UI policies are useful for conditional suggestions like showing and hiding fields or adding field messages based on another field's value, while Data policies and Business Rules are better suited for doing conditional enforcement like making a field mandatory.

The best user experience is to utilize both suggestion and enforcement together.

For more information, see the UI Policy article in the Client-side Scripting Module.

Build UI Policies and Data Policies to handle client-side activities before scripting any client-side logic. Use of Client Scripts to validate user input and provide feedback while the user is completing the form.

Some general practices for client scripting are:

- Optimize for performance by using asynchronous GlideAjaxover client-side GlideRecordor multiple getReference()calls.
- Keep the isLoadingcheck in onChangeclient scripts.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Keep the newValuecheck and add a newValue != oldValuecheck.
- Use all client-side scripts possible before making a server call with GlideAjax. Server roundtrips can impact performance.

Some client scripting practices to avoid are:

- Global Client Scripts or Global UI Scripts: Global scripts will run on every page load and introduce browser load delay.
- DOM Manipulation: Using document object model manipulation against default UI elements introduces upgrade risk and maintainability issues. The exception is using DOM manipulation against the DOM in pages authored in the same scoped application, like UI Pages or Service Portal widgets.

Business rules and script includes

Business rules are server-side actions that can be run during CRUD (Create, Read, Update, Delete) operations on instance records.

Note: Consider creating applications with help from agentic AI. For more information, see Build applications with AI.

Some good practices when using Business Rules are:

- Keep Business Rules small and specific.
- Avoid modifying base system Business Rules.
- Use Script Includes instead of global Business Rules.
- Use scripting only when necessary.
- Store reusable script logic in a script include.
- Use queries to limit records processed within a Business rule.
- Avoid client-callable Business Rules to improve efficiency when running client scripts.
- Always use a condition with Business Rules to control when the Business Rule runs. Running Business Rules with conditions can also aid in debugging. Business Rules rarely run with no conditions.

Business Rules can be configured to run before or after a database operation. They can also be configured to run asynchronously and also before displaying a form or executing a query.

Value	Runs	When to Use	Example
Before	Synchronously before the database operation	Set or update values on the current object as part of the save operation. Validate and abort execution if required.	A developer wants to set the state of the current record based on another input in that record.
After	Synchronously after the database operation	Trigger events and notifications after the database update to access the previous object or to make something occur in sequence. Update related records other than the base table being updated to access the previous object or to make something occur in sequence.	A developer wants to cascade values from the current record down to child records.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Value	Runs	When to Use	Example
Async	Asynchronously executed as a separate process after the database operation is completed	The process triggered by the rule may take a while to run. When the user who triggered the operation does not need the output right away. Trigger events, notifications, or related record updates when access to the previous values of the record or a specific sequence of actions is not required.	A developer needs to trigger an external process that may take a while or update a large number of records.
Display	Executed every time the corresponding form is displayed	Used to make server-side objects available to client-side scripts.	A developer wants to write information about a user associated with the current record to the g_scratchpad object to use in a client-side script.

Note: current.update()should not be used in any Business Rules. Using current.update()triggers an additional database operation, which could cause duplicate notifications, recursive loops, etc.

Use Script Includes to store JavaScript functions and classes for use by server scripts. Each Script Include defines either an object class or a function that can be reused among any server-side scripts. For more information, see Script includes.

Store any code that might need to be used elsewhere in a Script Include. Call the Script Include from a Business Rule, UI Action, workflow script, Scripted REST API, etc. Instead of calling a Business Rule from a UI Action or a UI Action from a Scripted REST API, put the code in a Script Include and call the Script Include from both places.

Keeping functions in a Script Include allows testing of the function before deploying the function in other scripted areas, thus reducing overall development and testing time.

For more information, see Classic Business rules.

Flow Designer

Flow Designer is a ServiceNow AI Platform feature that enables rich process automation capabilities in a consolidated design environment. Flow Designer enables process owners to use natural language to automate approvals, tasks, notifications, and record operations without having to code.

Flow Designer and IntegrationHub

For any new process flow requirements, ServiceNow recommends using Flow Designer over the legacy workflow for almost all circumstances.

Flow Designer and Business Rules

You should use Flow Designer instead of Business Rules unless:

- Business logic needs to run in a specific sequence with other Business Rules. For example, new business logic needs to run after one Business Rule but before another.
- Logic needs to execute immediately before or after writing to the database in the same thread.
- The logic only calls a Script Include.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

When designing a flow, follow these design principles:

- Single Purpose: Each flow should have a singular goal.
- Reusability: Design with reusable sub-flows in mind (approval is a great example).
- Clarity: The language and layout of a flow should make each action's purpose clear.

Start with a white board design of a business flow. Then build the flow action by action to align with the process. More than one flow may be required for a for a single process to keep to the design principles.

Use the following practices when working with Flow Designer:

- Use records, not SysIDs. Provide a guided experience with inline documentation.
- Learn how to use template objects to work with both static and dynamic inputs.
- Avoid passing around blobs of data unless absolutely necessary.
- Only pass information to a flow that the flow is going to use.

Use the following practices when working with Flow Designer Actions:

- Always create actions under the scope of the application's spoke, if applicable.
- Set access to Accessible from all scopes in actions to be able to reuse actions across other apps and scopes in the future.
- Set **Protection** to Read-only to avoid any unwanted edits to the actions by users.
- Make sure inputs have a specific type.
- Ensure that **Mandatory** is selected where required.
- If using a **choice** input type, use a default value.

Use the following practices when working with IntegrationHub:

- Create one spoke per integration system. Only put actions for a single system in a spoke.
- When creating the scoped app for the spoke, use a version naming convention that makes sense.
- Use a connection alias instead of an inline connection. The Base URL will be automatically extracted.
- Use connection attributes under the Alias to pass the version in a REST step giving future flexibility for versioning in the resource path.
- Use Save as Attachment to save the content in the response instead of creating another step to save the data.
- If the Alias is dynamic, make Alias one of the inputs and use the data pill to provide the Alias.

Use the following practices in Flow Designer and IntegrationHub for Error Handling:

- Create a Script Include to handle errors.
- Write short and understandable error messages.
- Incorporate all of the possible error messages the API returns.
- Ensure that the outputs from the integration step are validated before using them.
- Fail Early: If the inputs are not available, do not call the integration.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Self-Paced Training: Flow Designer

Self-Paced Training: IntegrationHub

Validate app functionality

As the application is built, validate that it works as expected.

Unit testing

Unit/Story testing ensures requirements specified in a story are validated before closing the story. A Story/Unit is a smallest testable portion of system or application that can be configured and executed.

When the configuration of the story is complete, developers need to unit test the features not only in the context of that particular story, but also other related stories that share components with the current story.

As a good practice, developers need to assign the story to process owner or designated stakeholder to validate the story configuration meets expected outcomes before closing the story.

ServiceNow's Automated Test Framework (ATF) is primarily meant for automating functional testing of applications but in few cases can be used to automate unit testing of configurations that involve Script Includes and Business Rules.

System testing

System testing is performed on a complete system when development is completed. Test the overall interaction of components and integrations with other applications within scope. System testing is performed by the QA/Testing team, but developers need to collaborate with the QA team and process owners to ensure test cases provide comprehensive coverage. Developers will be responsible for remediation of issues found during System Testing.

Automated Test Framework

Automated Test Framework (ATF) should be leveraged for automating functional system testing of ServiceNow applications to reduce testing time and costs and make testing repeatable and UI independent. When creating test cases, follow these guidelines.

When creating tests:

- Use parameterized testing to avoid duplicate test cases.
- Follow a Test naming standard.
  - `<app initial>: <functionality that is being tested>`
  - CSM: Resolve case
- Describe each test's use case in its description. For example: Sample that tests use case.
- Develop tests on a Development instance and promote/run the test on a Test instance.
- Clones wipe out tests. Use one of these options to preserve tests:
  - Bundle tests in a scoped app and upload the app to GIT.
  - Save tests before the clone.
  - Promote tests to prod instance, but DO NOT EXECUTE THE TESTS IN PROD.
- Create self-contained tests.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Create new server-side or REST test steps any test steps are missing. For example: Email body verification.
- Use server-side test step whenever possible and when screenshots are not important.
- Start with the Impersonatestep.
- Be aware of browser throttling.
- Use the Test Logs and Test Transactions to troubleshoot test errors.

When creating test suites:

- Follow a test suite naming standard. For example: ITSM INT: Use cases.
- Describe the suite.
  - Test suite description: "This is a sample test suite to test plugin/application".
  - Provide any additional information possible in the description.
- Organize test suites by feature areas.

User acceptance testing

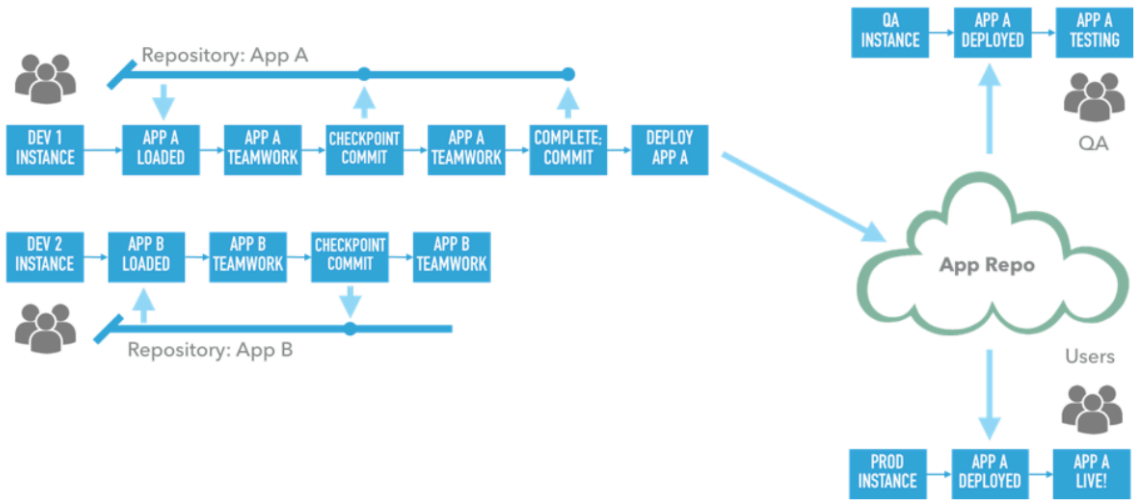
User Acceptance Testing (UAT) is a test conducted to evaluate the application's compliance with the business requirements and assess whether the application is acceptable for delivery. Users, customers, or other authorized stakeholders perform acceptance testing. Developers will be responsible for remediation of issues found during System Testing.

Deploy your app

Once the application is built and validated, the application needs to be moved to the production environment. Applications can be moved through an application repository or by using Update Sets. Applications should be deployed to test environments prior to moving to production.

Application Repository (App Repo)

Publishing an application to the App Repo makes this version of the application available to all of an organization's ServiceNow instances. Use the App Repo to deploy an application to QA / Test instances (for testing) and finally to Production (Prod) instances.

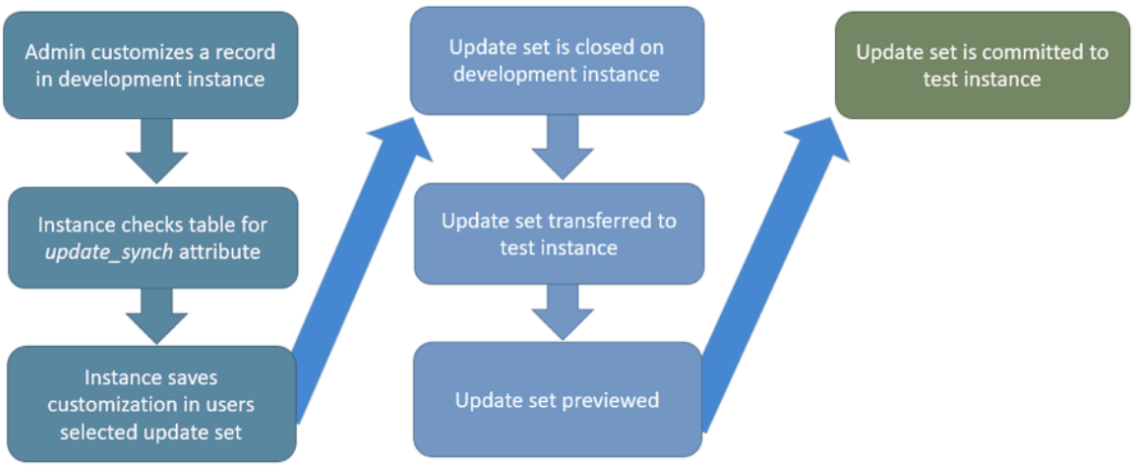


For more information, see Publishing an application to the application repository and Installing applications, plugins, and products.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Update Sets

If the application repository cannot be used to deploy applications, use Update Sets instead. The diagram shows the lifecycle of an Update Set to deploy a customization from the development instance to the test instance.



Practices that lead to a quality development and release process:

- Always move customizations from the bottom of the stack up.
  - Ensures down-stack instances match up-stack instances.
  - Customizations introduced mid-stack can be overwritten by future pushes from down-stack.
  - Common scenarios include:
    - Fixes need in test or prod – always push them from dev up
    - Common Prod admin customization, such as choice lists – always push updates from dev up
- Always review updates contained in an Update Set before transferring.
  - Look for updates associated with other development efforts and updates associated with testing.
  - Watch for system properties and integration end-points changes. Ex: pushing sys_properties change that directs all email to test email account
  - Move updates to a “scrap” Update Set rather than deleting the update.
- Always test after pushing to ensure that all desired customizations are captured and applied as expected.
- In situations with multiple, parallel releases, ensure communication and coordination between the development teams.
- Avoid experimenting on the development instance, because customizations can be captured accidentally and migrated by other team members.
- Do not capture development in the Default Update Set.

List all user Story numbers with a short description in the **Description** field of an Update Set. Include all manual steps that are required to deploy the Update Set.

Some typical examples of manual steps that are needed for a deployment that are not captured in an Update Set:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Plugin activation.
- Transfer of tables that are not tracked in the Update Set (typically, starting with “x_” or “u_”).
- Creation of database indexes on the tables. The index creation is not tracked via Update Set and needs to be done manually.

Update Set management

Be sure the correct Update Set is selected when working on a story or a defect and check the records in the Update Set daily.

Do not manually move sys_update_xml records between Update Sets. The only exception is to move a record to the Default Update Set.

Update Sets capture configuration information but not task or process data. For example, Update Sets track Service Catalog item definitions and related configuration data like variables and variable choices. However, the orders (requests, items, catalog tasks) placed in testing are not tracked by Update Sets.

Be aware of Update Set DOs and DON’Ts:

- To remove a specific sys_update_xml record from the current Update Set, move the record to the Default Update Set and populate the **sys_update_set.comments** field of the record with the reason for moving the record to the Default Update Set.
- Never move customization records from one Update Set to another Update Set.
- Never delete an Update Set unless the Update Set has been merged successfully into a new Update Set.
- Always use data extracts or Import Sets to move data from one instance to other (and not Update Sets).

Update Set batching

Batch Update Sets to preview and commit Update Sets in bulk.

Dealing with multiple Update Sets can lead to problems, including committing Update Sets in the wrong order or inadvertently leaving out one or more sets. Avoid these problems by grouping completed Update Sets into a batch.

The system organizes Update Set batches into a hierarchy. One Update Set can act as the parent for multiple child Update Sets. A given set can be both a child and parent, enabling multiple-level hierarchies. One Update Set at the top level of the hierarchy acts as the base Update Set.

Previewing or committing the base Update Set previews or commits the entire batch. The system determines the processing order, and checks for collisions based on the dates the changes were recorded and on their sequential ancestry. Their ancestries are the specific instances in which the changes in the Update Sets took place.

Update set batching can be applied to releases, where an empty parent Update Set is created for the release, and actual Update Sets are included in the release as children.

Advantages of using Update Set batching are:

- Individual Update Sets can be removed from the release at the last moment.
- Batching is similar to merging, except batching allows updates to be removed.
- Batch Update Sets are easy to deploy. Only the Parent Update Set needs to be processed.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For more information, see System update sets.

What to do next

Now that the app is deployed, think about how to improve and enhance it. Here are some suggestions for determining where to go next:

- The people using the application day-to-day will be the best source of feedback. Talk to them about what new features or changes they would like to see.
- Determine if additional related process flows can be automated through Flow Designer.
- Determine if new IntegrationHub spokes can be leveraged for new integrations.

Builder library

The ServiceNow AI Platform provides a robust set of builder tools that enable you to quickly develop applications for your organization.

The following builder tools enable you to quickly design and build user experiences, interface components, data models, and more when developing applications on the ServiceNow AI Platform.

Builder tools

Tool	Description	Use case(s)
Exploring Decision Tables	**Configure application decision logic** Use decision tables to help resolve a complex decision that depends on multiple factors. Decision Builder enables developers to decouple decision logic from their code by creating and maintaining decision rules.	Determine a customer discount rate for a product given two inputs (number of units ordered, and whether or not it was paid for in cash).
Flow Template Builder	**Configure a standard template for specific app workflows** Flow Templates guide flow authors to create flows for common use cases. Use this builder to define flows, actions, and flow template variables.	Create a flow template to use as the basis for all request approvals within an IT application.
Integration with Performance Analytics	**Optimize processes through integration with Performance Analytics** Use Process Optimization with Performance Analytics to perform data extraction from an indicator and analyze processes associated with the KPIs, such as time to resolve.	Visualize the execution of processes or a customer journey, show insight into adoption, activity transitions, performance, and user experiences or identify bottlenecks and see their impact on your key performance indicators, enabling you to resolve root-cause issues.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Builder tools (continued)

Tool	Description	Use case(s)
Mobile App Builder	**Configure Mobile apps** The Mobile App Builder is a configuration tool to build and manage screens and records that make up workflows within ServiceNow mobile apps. The organizational layout and navigation options in the Mobile App Builder facilitate a faster and more intuitive creation of ServiceNow mobile applications.	Streamline mobile configurations by organizing records in a hierarchical layout for quick access and reference, supporting hierarchical tree navigation, and enabling users to visualize how records relate to each other in the overall mobile app configuration.
Mobile Card Builder	**Configure and manage card templates to display information for mobile interfaces** Use Mobile Card Builder to create or modify card templates used in the ServiceNow mobile applications for iOS and Android. The ServiceNow mobile interface uses cards to display information about records on your instance.	Create cards for launcher screen record sections, calendar screens, form screens, Genius search results, list screens, and map screens.
NLU Workbench	**Configure Natural Language Understanding (NLU) models to interpret user-expressed intent** ServiceNow® Natural Language Understanding (NLU) provides an NLU Workbench and an NLU inference service that you can use to enable the system to learn and respond to human-expressed intent. By entering natural language examples into the system, you help it understand word meanings and contexts so it can infer user or system actions. NLU workbench is where you create, manage, and test NLU models. Use NLU models to apply ServiceNow® Natural Language Understanding on your instances. Create, manage, test, and publish NLU models with the NLU Workbench.	Configure a Virtual Agent Designer conversation flow to consume an NLU model so that agent chatbots can better understand user statements in the conversation.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Builder tools (continued)

Tool	Description	Use case(s)
Platform analytics for workspaces	**Configure performance analytics that capture historical trends in data through snapshots** Integrate data visualization and analytical functionality into the workspace experience. Subscribe to notifications about behavioral process changes. Explore KPIs and get answers and insights on Platform Analytics.	Configure the ability for users to ask natural language questions and be shown the relevant analytics in a visualization, view any Next Experience dashboard that are owned by a user or shared with a user, create dashboards and data visualizations from workspace runtime, or drill down into KPIs, apply filters to slice data, or apply aggregations and statistical tools.
Reports	**Build and distribute reports that show the current state of instance data** ServiceNow® Reporting enables you to create and distribute reports that show the current state of instance data. Reporting functionality is available by default for all tables, except for system tables.	Show the number of open incidents at an organization that also displays the priority of each incident at a current point in time. A variety of different types of reports could be configured to show this information, including bar, pie, time series, multidimensional, scores, statistical, etc.
Service Portal Designer	**Create a self-service portal for your users** Service Portal enables you to build a mobile-friendly self-service experience for your users. It interacts with parts of the ServiceNow AI Platform, so users can access specific ServiceNow AI Platform features using Service Portal. This tool is used to create service portal pages to organize content, ensure responsive mobile optimization, and design meaningful portal user experiences for your customers. A page houses containers and rows, which then contain widgets. By manipulating the layout of the page, and the widgets within it, you can construct your desired user experience. Widgets are what define the content in your portal. You can use the base system widgets provided with Service Portal, clone and modify widgets, or develop custom widgets to fit your own needs.	Configure a mobile-friendly experience for a company's employees to engage with IT and request services like new emails, new employee computer setup, etc.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Builder tools (continued)

Tool	Description	Use case(s)
Table Builder	**Create application tables and table-related forms and configure display logic** Table Builder enables you to create, design, and administer tables and forms visually from a single user interface.	Create an online food order form in an food delivery application for customers. The form might capture the food ordered and related delivery options. The form would store and capture the information entered in a designated application data table that you configure for this purpose.
Themes in Theme Builder	**Customize themes and branding** Theme Builder enables you to tailor the visual experience for your users, helping to update the look and feel to be more like your brand. Next Experience extends the theming model used in UI Builder.	Configure a dark theme for your company's brand that makes use of a primary color and secondary color, specifies what color to use for screen elements like divider lines, and contains a new brand logo.
UI Builder	**Design application user Interfaces** UI Builder is a web user interface builder. Use UI Builder to build pages for workspaces or custom web experiences (generated by App Engine Studio) that use Next Experience Components and custom web components.	Create a new standard record page variant to show specific incident records, edit the standard record page to show a list of problems for a user, or build a landing page for a Next Experience app you are designing.
Virtual Agent Designer	**Design virtual agent conversation blueprints** The Virtual Agent Designer is a diagram tool for creating and managing topics, which are blueprints for conversations between a virtual agent and user. You can design topics that help your users resolve common work issues or guide them through self-service tasks.	Create and configure a conversation flow that enables end users to see the status of incidents they submitted and leave a comment in the incident.
Workspace Builder	**Create custom workspaces for your application** Workspace Builder is a streamlined, no-code environment that enables you to create a custom workspace from within App Engine Studio or ServiceNow Studio quickly and efficiently.	Configure a Service Desk workspace that will handle request and fulfillment of IT services within your organization.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Create custom components using ServiceNow CLI

Develop custom components using the Next Experience UI Framework and the ui-component extension.

Components are reusable building blocks that you use to create a custom user interface. The Next Experience Design System comes with a set of customizable components that you can drag into your custom UI. Develop your own components if you can't find what you need in the Next Experience Design System library.

To see the Next Experience Design System library, visit the ServiceNow® Developer Site.

Benefits of creating custom components

Developing custom components lets you:

- Personalize a UI according to your agent, customer, and company needs.
- Make your employees more effective and reduce context switching with quick access to important data and information.
- Accommodate your company's unique omni-channel environment using APIs to consolidate your data.

For example, you might want to create a component that displays the cases associated with an SLA, or that tracks the active chats in a particular queue. You can use the Next Experience UI Framework and the ui-component extension to develop the component you need, and access data from your platform using the Http Effect API. You can also query platform data using GraphQL by creating a custom schema. For more information, see Scripted GraphQL.

What to know before you begin

Before you start designing and building your component, make sure you have:

- Some general knowledge of web component concepts, development, and design.
- JavaScript knowledge to define the component behavior.
- Knowledge of Node Package Manager (npm).
- The most recent version of Node.js installed on your local machine. For more information, see, Node.js.
- The ServiceNow CLI installed on your machine.

Next Experience UI Framework

The Next Experience UI Framework is a JavaScript framework that lets you extend your apps and build web components that are reusable throughout your applications. Using the Next Experience UI Framework lets you:

- Create a single component to use in multiple places across your applications.
- Encapsulate the component's scope to prevent conflicts with other code.
- Add properties, slots, and actions to your component, allowing users to customize the component every time they use it in a Workspace.

For more information, see the ServiceNow® Developer Site.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ui-component extension and development flow

The ui-component extension is an extension to the ServiceNow CLI that lets you develop custom components using the Next Experience UI Framework. With the ui-component extension, you can:

- Create the set of files required to develop a component, or project scaffolding.
- Start a local development server to test your component.
- Build a component project and deploy it to a ServiceNow instance.

For more information, see the ServiceNow® Developer Site.

Application scope

When you deploy a Next Experience UI Framework component, it deploys into a scoped application on the instance. You can provide an application scope for the component to use as a namespace identifier. Use the namespace identifier guidelines for application development on the instance. For more information, see Application scope.

When reserving an application scope, follow these requirements:

- Maximum: 18 characters.
- Case: snake case.
- Format: `x_customerprefix_componentname`, where:
  - `customerprefix` is the value in the *`glide.appcreator.company.code`* system property on your instance.
  - `componentname` is the value provided in the component's `name` parameter when you created the project.

If you do not provide an application scope when creating your component project, the Now CLI creates one for you.

Alternatively, you can add a value to the `scopeName` parameter in the `now-ui.json` file.

Reference guide

To see the Now CLI reference guide, visit the Developer Site.

Develop a component for Virtual Agent

Create a custom Virtual Agent component to gather input or display information in the Virtual Agent client interface.

Types of Virtual Agent components

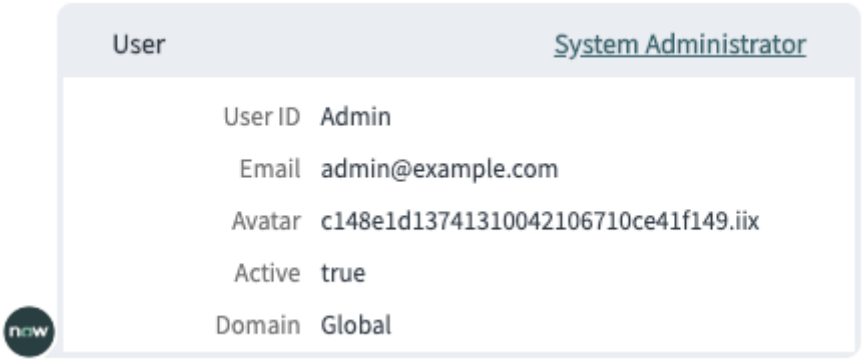
To develop a component for Virtual Agent, add specific properties and actions to interact with the Virtual Agent client interface. The properties required depend on the type of component you are creating.

Response component

A response component only provides information to the user, and does not gather input or handle user interaction. For example, a card

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

control that does not require user input and is only in the conversation



once.

Add a property to your response component to handle the data sent by the Virtual Agent server.

Output component properties

Property	Description
controlData	Initial data that the Virtual Agent server sends to your component as the topic runs. Data type: JSON Object

Input component

An input component displays information and/or gathers user input. It includes the same property as the output component to handle data sent by the server, but has more possible states and requires user interaction.

Input component properties

Property	Description
controlData	Initial data that the Virtual Agent server sends to your component as the topic runs. Data type: JSON Object
responseValue	Data sent to the component from the user's response, either from the client directly, or from the server if there is a refresh. Only use in components that require user input. Data type: JSON Object
forceCloseControl	Flag that indicates whether the component can accept input. When true, the control closes and the user cannot interact with it. Monitor changes on the client to update this value. Only use in components that require user input. Data type: Boolean

Use this action to emit data from user interaction.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Input component actions

Action	Description
VA_CONTROL#VALUE_SENT	Response data from the client to send to the server. Only use in components that require user input. Data type: JSON Object

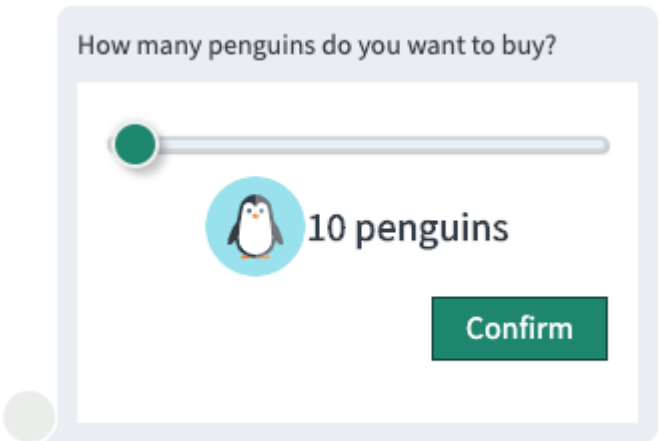
Input component states

Because they accept data, input components must handle multiple states. The state flow is generally as follows:

1. Virtual Agent shows the custom component in the waiting for input state.
2. The user interacts with the component to provide input.
3. The component closes and sends the `responseValue` property to the server.
4. The server runs the server-side logic and sends the component with the user's input back to the client.

Waiting for Input

The initial state of a component waiting for user interaction. The `controlData` property is set, but the `forceControlClosed` property is false. This example shows a slider component in the waiting for input state.



In this example, if the user has not provided a `responseValue` and the control is not closed, the slider and the input button display.

```
const {controlClosed, sliderVal, sliderMin, sliderMax}
 = state;
return (<div class={{"slider-chat": true}}>
    {responseValue ? null :  <Fragment>
    <div class={{"slider-label": true}}>{label}</div>
    {controlClosed ? null : <Fragment>
        <div class={{"slider-container": true}}>
                    <input on-change={onSliderChange}
 type="range" min={sliderMin} max={sliderMax}
 value={sliderVal} class={{"slider": true}} />
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
            <div class={{"slider-value": true}}>
                    {unitIcon &&
                     <div class={{"unit-icon":
 true}}><img src={unitIcon} /></div>
                     }
                     {sliderVal} {unitName}
                </div>
        <div class={{"button-container":true}}>
            <now-button variant="primary"
 label={buttonText || 'Confirm'} />
         </div></div></Fragment>}
```

Handling Input

In the previous example, the slider uses a `now-button` component, which the user clicks to confirm input and send it to the server. When the user clicks the button, the `VA_CONTROL#VALUE_SENT` action fires with the `responseValue` payload.

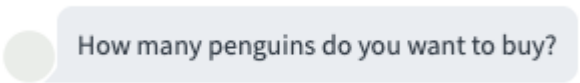
```
'NOW_BUTTON#CLICKED': (data) => {
    const {updateState, dispatch, state: {sliderVal}} =
 data;
    updateState({controlClosed: true});
    dispatch('VA_CONTROL#VALUE_SENT', {
        value: {
            sliderVal
        }
    });
}
```

Closed

A component that is closed can no longer accept user input. Components are generally closed because:

- The user responded to the component. The component closes and the conversation continues.
- The user ended the chat. The server does not wait for a response.

For example, the slider component only renders the original prompt when in the closed state.



Sending response

After the user responds, the control is rendered again on the user's side of the conversation with the value of the `responseValue` property.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



For example, the slider control uses this snippet to render the response.

```
{responseValue && <div className="response-container
 slider-value">
   {unitIcon &&
   <div className="unit-icon"><img
 src={unitIcon} /></div>
   }
   {returnVal} {unitName}
</div>}
```

Adding the component to Virtual Agent Designer

After developing the component and deploying it to your instance, add it to Virtual Agent Designer using a custom control and definition. For more information, see Virtual Agent custom controls.

Add properties to communicate with Virtual Agent

To develop a component for Virtual Agent, add specific properties and actions to interact with the Virtual Agent client interface. The properties required depend on the type of component you are creating.

Before you begin

- Set up your environment. For instructions, see the ServiceNow® Developer Site.
- Set up your component project. For instructions, see the ServiceNow® Developer Site
- Develop your component. For instructions, see the ServiceNow® Developer Site

Role required: virtual_agent_admin or admin

Procedure

1. In your component code, add the Virtual Agent properties to interact with the Virtual Agent client interface.

Virtual Agent component properties

Property	Description
controlData	Initial data that the Virtual Agent server sends to your component as the topic runs. Data type: JSON Object

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Property	Description
responseValue	Data sent to the component from the user's response, either from the client directly, or from the server if there is a refresh. Only use in components that require user input. Data type: JSON Object
forceCloseControl	Flag that indicates whether the component can accept input. When true, the control closes and the user cannot interact with it. Monitor changes on the client to update this value. Only use in components that require user input. Data type: Boolean

2. If creating an input component, use the `VA_CONTROL#VALUE_SENT` action to send user values to the server.

Virtual Agent component actions

Action	Description
VA_CONTROL#VALUE_SENT	Response data from the client to send to the server. Only use in components that require user input. Data type: JSON Object

What to do next

Test a component for Virtual Agent.

Related topics

Create custom components using ServiceNow CLI

Test a component for Virtual Agent

Test your Virtual Agent custom component before deploying it to your instance.

Before you begin

- Set up your environment. For instructions, see the ServiceNow® Developer Site.
- Set up your component project. For instructions, see the ServiceNow® Developer Site
- Develop your component. For instructions, see the ServiceNow® Developer Site
- Add properties to communicate with Virtual Agent

Role required: virtual_agent_admin or admin

About this task

Virtual Agent components must be tested within the Virtual Agent client chat to ensure that the component responds correctly to user input. You can set properties in your component project to test your component in a mock Virtual Agent test client tool.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Add a dependency on the test client tool to your component project.
   a. Open the `<component-name>/package.json` file in your component project.
   b. Add `"@servicenow/sdk-ci": "1.0.8"` and `"@servicenow/library-translate": "^18.0.0"` to the `devDependencies` object.

Example
Here is an example `package.json` file with the correct dependencies.

```
"dependencies": {
    "@servicenow/ui-renderer-snabbdom": "brazil",
    "@servicenow/library-translate": "^18.0.0",
    "@servicenow/now-button": "brazil",
    "@servicenow/now-dropdown": "brazil",
    "@servicenow/sass-generic": "brazil",
    "@servicenow/cli-archetype": "brazil",
    "@servicenow/sdk-ci": "1.0.8"
}
```

2. Add sample properties to use as the initial state of the component in your test.
   a. Add a `<component-name>/example/sampleProps.json` file with initial properties to render in your test.

Example
Here is an example `sampleProps.json` file with initial properties set for a slider component.

```
{
 "label": "How many penguins do you want to buy?",
 "defaultValue": 10,
 "sliderMin": 20,
 "sliderMax": 80,
 "unitName": "penguins",
 "unitIcon":
 "https://image.flaticon.com/icons/svg/141/141836.svg"
}
```

3. Update the `example.js` file to open the component through the test client tool using the sample properties that you created.
   a. Open the `<component-name>/example/element.js` file in your component project.
   b. Add statements to import `@servicenow/sdk-ci` and the sample properties file you created.
   c. Add the following statement, replacing `<component-name>` with your component's name to create the test tool with initial data from your sample properties.

```
const el =
 document.createElement('tool-ci-custom-control-tester');
el.componentTagName="<component-name>";
el.initialExampleData=sampleProps;
document.body.appendChild(el);
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example

Here is an example `example.js` file that opens the component using the test client tool.

```
import '../src/now-chat-control-slider';

import '@servicenow/sdk-ci';

import sampleProps from './sampleProps.json';

const el =
 document.createElement('tool-ci-custom-control-tester');
el.componentTagName="<component-name>";
el.initialExampleData=sampleProps;
document.body.appendChild(el);
```

4. Run the development server command to view your component in a test browser.

```
$ snc ui-component develop [--entry entry --open --port port
 --host host]
```

Pass in values for these parameters.

Name	Description
entry	Path to the test module in your component project. Default: `example/index.js`.
open	Opens the default browser and navigates to the test page. Default: false.
port	Port where the development server runs. Default: 8081.
host	Host address to use if you want your local development server to be accessible externally by others. Typically set to 0.0.0.0

Example

```
$ snc ui-component develop --entry example/hello.js --open
 --port 3000
```

The component opens in the test client tool. You can see the initial data provided in the **Custom Control JSON Input Data** field and the component's output in the **Custom Control Return Data** field.

What to do next

If your component is working as expected, deploy it to your instance. For instructions, see the ServiceNow® Developer Site.

After developing the component and deploying it to your instance, add it to Virtual Agent Designer using a custom control and definition. For more information, see Virtual Agent custom controls.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Related topics

Create custom components using ServiceNow CLI

Add a component to Agent Workspace

Use custom components to create a custom Workspace interface to fulfill the specific need of your company's agents.

Communicating with customers through multiple channels can be time consuming. To be efficient in these omni-channel interactions, your agents need a single view of customer information to reduce context switching between multiple tools. By developing custom components for Workspace, your team can bring communications from multiple channels into one interface.

Adding components to a Workspace

Once deployed to your instance, you can add components to Workspace in these ways.

Add a component to a Workspace modal

Use a UI action to launch a custom component in a modal so an agent doesn't have to navigate to a different screen to accomplish a task.

Add a component to a Workspace landing page using UI Builder

Use the UI Builder to create custom landing pages for your agents. UI Builder is a drag-and-drop tool that lets you visually arrange workspace components.

Configure properties in the `now-ui.json` file to use deployed components in the UI Builder. For instructions, see the ServiceNow Developer Site.

Add a component to a Workspace record view

You can add custom or standard components to the component area in the Workspace record view.

Decision Tables

Use decision tables to decouple decision logic from your code by creating and maintaining decision rules.

Important: Try creating decision tables in Workflow Studio instead. Workflow Studio integrates workflow authoring, configuring, and monitoring into a single page experience. For more information, see Create decision tables in Workflow Studio.

Use decisions tables to help you reach outcomes that depend on multiple factors. In these tables, each factor is a decision input. For example, if you are trying to make a decision about car insurance coverage, your inputs might include the location where the insured person lives, the age and accident history of the insured person, the car make, the car model, and the car year. This logic can save time and present a more organized, readable format than using a script. Decision tables provide a single point where you can create, view, and modify decisions.

To interact with Decision Tables in script, use the DecisionTableAPI - Scoped, Global.

Components of a decision

Decision Table [sys_decision] records

Decision table records represent a single decision. In this record, you specify which table you want to use for your answers. This record also contains related lists where you can create your decision inputs and decisions.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Decision inputs [sys_decision_input] records

Decision input records represent your inputs that are used to obtain answers in a decision. These inputs can include a variety of types, including strings, references, true/false inputs, and dates. Each decision input has a specified input type and can be assigned a default value.

Decision [sys_decision_question] records

Each decision record represents a decision that is based on your inputs. Use the condition builder on the decision record form to create a condition that is based on the inputs for this decision. Then, you can select an answer record for this decision. The answer record can be any record from the table that you have defined in the Decision Table record. You can select the same answer record for more than one decision.

Answer [sys_decision_multi_result_element] records

Answer records represent answers that are reached using different decision input values. Answers records can be records on any table, but you need to choose the table when creating your Decision Table record. You could, for example, use the sys_choice table to use existing choice records. From the example about insurance coverage, you could create a table where each record contains details about the levels of insurance coverage.

Decision tables in the classic environment

In the classic environment, you can develop a table to use for your decision answer records. You then associate an answer record to each decision on your decision table. This answer record is returned when the decision is reached based on your inputs.

Note: Workflow Studio decision tables provide a new intuitive interface to create decision tables. For more information, see Workflow Studio.

Create a decision table to resolve complex decisions

Create a decision table to help you resolve a complex decision that depends on multiple factors.

Before you begin

Role required: decision_table_admin

Procedure

1. Create or select a table to use for your decision answer records.
   You will associate an answer record to each decision on your decision table in later steps. This answer record is returned when that decision is reached based on your inputs. For information on creating a table, see Create a table.
2. Ensure that one field on the table that you use for your decision records has been set as the display value.

   This field is used as a label when you display decision answers in Workflow Studio.

   You can use an existing table or create a new table to use for your decision records.
3. Create a record on this table to each answer in your decision.
   If you have modified an existing table, you may need to customize the form to show the new fields that you have added.
4. Create a decision table record.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. Navigate to **System Definition** > **Decision Tables**, and select **New**.

b. On the form, fill in the fields.

Decision Table form

Field	Description
Name	Descriptive label for this decision.
Application	Application scope of this decision. This field defaults to the current scope.
Answer table	Table that this decision uses for answer records. Select the table that you created in the previous steps.
Accessible From	Option to limit the availability of this decision to the current scope. Select **All application scopes** or **This application scope only**.

c. Right-click the form header and select **Save**.
The form refreshes with the Decision Inputs and Decisions related lists.

5. Create decision input records.

a. In the Decision Inputs related list, select **New**.

b. On the form, fill in the fields.

Decision Input form

Field	Description
Name	Name of the decision that is associated to this decision input. This field is automatically populated.
Application	Application scope of this decision input. This field defaults to the current scope.
Type	Type of data that is used for this input. When you use this decision in a flow, you can only use data pills that match this type.
Active	Check box to activate this input. This field is selected by default.
Label	Descriptive label for this input.
Read only	Check box to make this decision input read-only.
Column name	Column name for this input. This field is automatically populated when you give the **Label** field a value.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Mandatory	Check box to make this decision input mandatory.
Display	Check box to indicates that this decision input is the display value for reference fields.
Choice	Select a method for users to see a list of suggested values: ▪ List menu without -- None -- ▪ List menu with -- None -- ▪ Suggestion field type If a choice is used, define your choices in the Choices related list at the bottom of the form.
Default Value	Default value for this input.

c. Select **Submit**.
Your changes are saved and the decision table record reopens.

6. Create decision records.

a. In the Decisions related list, select **New**.

b. On the form, fill in the fields.

Decision form

Field	Description
Label	Descriptive label for this decision table.
Application	Application scope of this decision. This field defaults to the current scope.
Order	Order in which the flow evaluates decisions. The order can be important in flows that use the **First decision that matches** option and ends after the first match is found.
Answer	Answer that is used when the conditions in this decision are met.
Default Answer	Check box to enable this decision as the default for your decision table.
Condition	Conditions that are needed to reach this decision. The fields available for your condition are the Decision inputs that are associated with this decision table.

c. Select **Submit**.
Your changes are saved and the decision table record reopens.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

What to do next

With Workflow Studio, you can add your decision to the **Make a decision** flow logic. For more information about the **Make a decision** flow logic, see Make a decision flow logic.

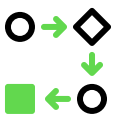
Table Builder

Table Builder enables you to create, design, and administer tables and forms visually from a single user interface.

Announcements

The Table Builder for App Engine application has been combined with the standalone Table Builder application. Some features such as schema view, spreadsheet view, flows, and PDF extractor that previously were only available in the Table Builder for App Engine application are now available in Table Builder.

Get started

Explore	**Use**	Reference
		
Learn about Table Builder concepts and features.	Manage tables, forms, and related policies and rules using Table Builder.	Get details about Table Builder components like field properties and properties for policies and rules.

Get help with Table Builder

Contact your company's Customer Admin to unlock or add user accounts, perform restores or zBoots, and more.

Following is the intro video about the table builder.

https://player.vimeo.com/video/1138041032?h=eb605503b7&badge=0&autopause=0&player_id=0&app_id=58479

- Ask or answer questions in the Table Builder community forum
- Search the Known Error Portal for known error articles
- Contact Customer Service and Support

Exploring Table Builder

Learn about Table Builder features that enable you to design tables, forms, and flows visually using a single user interface.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Learn more about Table Builder	Additional ServiceNow resources
You can use Table Builder to configure form elements such as layouts, fields, field configuration, and UI policies.	
	 Table Builder: Creating a custom order form

Key features

Table Builder includes these key features:

- Edit tables by adding additional fields or changing existing field properties for a particular application or globally.
- Visually customize form views by using a simple drag-and-drop editor interface. You can add and arrange sections, fields, and annotations as needed from a table that you select.
- Configure UI Policies for your forms from one interface or view these other related policies and rules:
  - Access control rules
  - Client scripts
  - Business rules
  - Workspace view rules
- Preview your forms throughout the editing process so that you can be sure that any changes function well in your environment.
- Integrate with App Engine Studio to promote a unified experience for designing your forms and managing data.

Accessing Table Builder

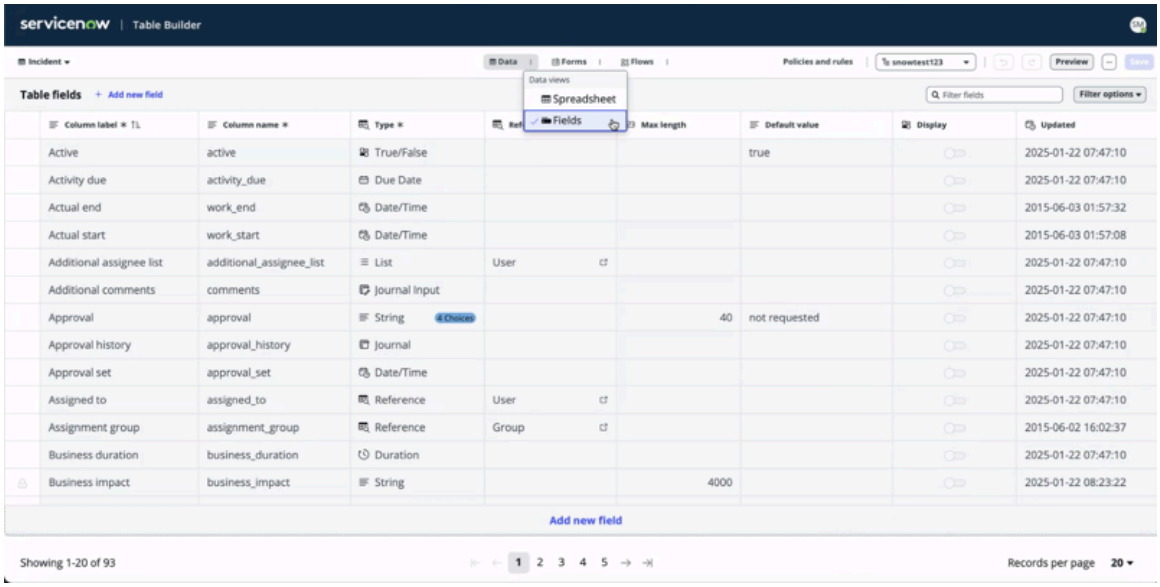
You can access Table Builder in several ways.

Launching Table Builder using naming shortcuts

You can use naming shortcuts to open tables from the app navigator.

Enter `tablename.builder`, `tablename.view`, `tablename.sheet`, or `tablename.flow` in the search to access the table directly. For example, searching `incident.builder` in the search opens field view of the incident table in the Data tab.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Naming conventions user inputs and their behavior

User Input	Behavior
<tablename>.builder	Opens the table in the Table Builder with the **Field** option selected in **Data** tab.
<tablename>.view	Opens table in Table Builder in the **Form** tab.
<tablename>.sheet	Opens a table in Table Builder with the **Sheet** option selected in **Data** tab.
<tablename>.flow	Opens a table in Table Builder in the **Flow** tab.

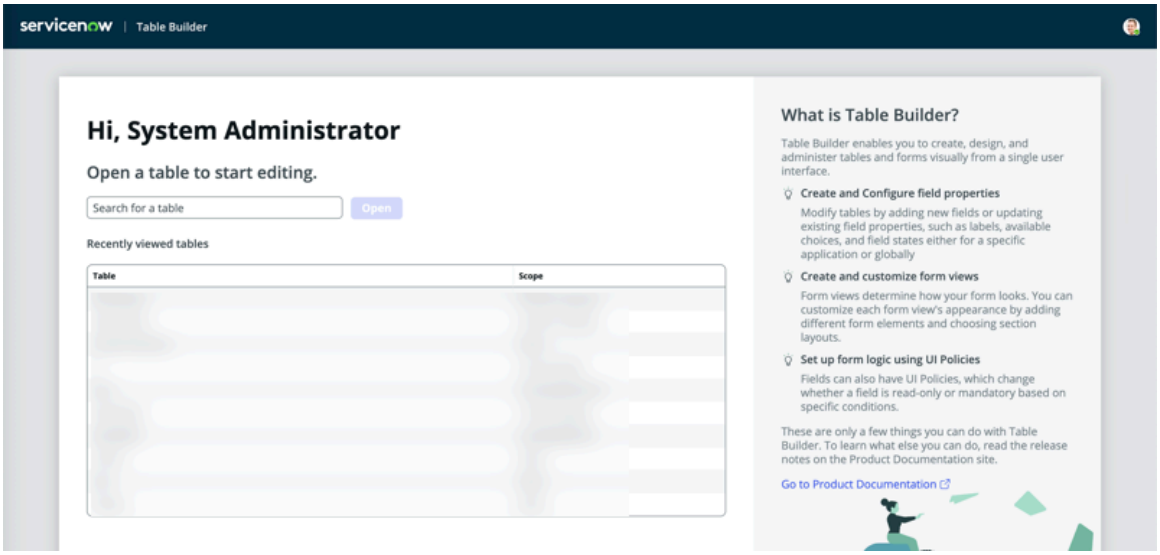
Launching Table Builder from the App Navigator

Table Builder is accessible through the application navigator. This feature helps you to interact with Table Builder more efficiently.

1. Navigate to **All** > **Table Builder**.
2. Search for the table that you want to open.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

3. Select **Open**.



Launching Table Builder from App Engine Studio

Table Builder can be currently accessed from within App Engine Studio.

See Editing data in App Engine Studio for instructions on editing application data.

1. Open an application in App Engine Studio.

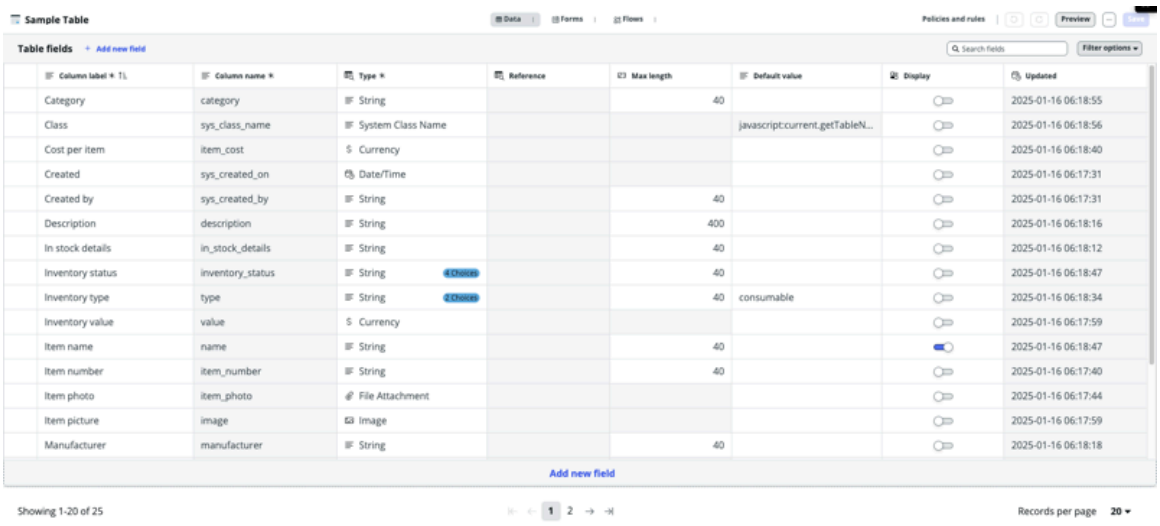
   Note: A data table must first be present in the application so that you can access Table Builder.

2. From the application home, select the menu icon (•••) next to a table, and then select **Edit**.

   Table Builder opens with the **Data** tab selected as shown in the following example.

   Note: To edit form views for the selected data table, navigate to the **Forms** tab.

Data tab



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Launching Table Builder from UI Builder

Table Builder can be accessed from within UI Builder by selecting **Edit form view** at the bottom of the Config panel, for a selected form object.

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**, and then select a form component on a page.

   See Work with components in UI Builder for more information on adding components.

2. Select **Edit form view** at the bottom of the Configuration pane.

   Table Builder opens with the **Forms** tab selected, where you can customize your form views for the selected table. To edit table data, navigate to the **Data** tab as shown in the following example.

Forms tab

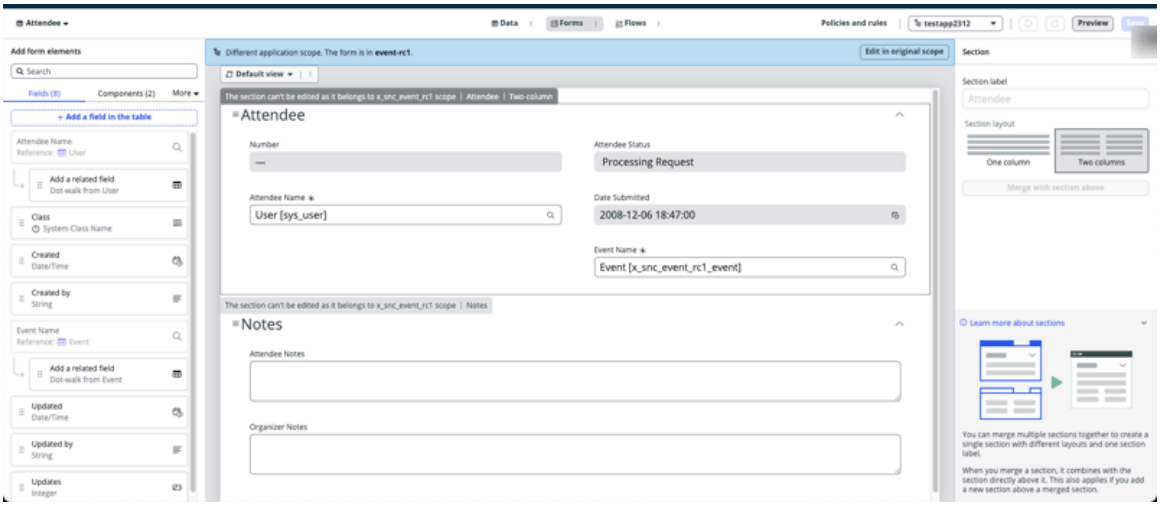


Table Builder permissions

To use Table Builder, you must either:

- be an administrator, or
- have the following role permissions in ServiceNow AI Platform, or
- have the relevant AES user role and delegated developer permissions.

For more information, see Delegate developers using AES.

Note: When creating a table, assign at least one role with **Read** permission. Setting read permission helps you to preview and access the table.

Table Builder permissions

Permission	Access
Personalize Form (personalize_form)	• Required to access **Forms** tab. • Required to customize fields on the form. • Doesn't grant dictionary access.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Table Builder permissions (continued)

Permission	Access
	• Doesn't grant Policies and Rules access. • Doesn't grant Tables access.
Personalize Dictionary (personalize_dictionary)	• Required to access **Data** tab. • Required to access field configuration. • Required to change dictionary fields. • Doesn't grant Policies and Rules access.
Personalize Rules (personalize_rules)	Enables access to the **Policies and Rules** tab.
Personalize Choice (personalize_choices)	Enables you to configure choices for a field.
Flow Designer (flow_designer)	Required to access the **Flows** tab. Enables you to configure flows. To work with Table Builder flows, you must also have personalize_form and personalize_dictionary permissions (or related delegated developer permissions).

Accessing Form Builder

Form Builder is an application that is accessed most often through Table Builder on the **Forms** tab. You can also access Form Builder through the ServiceNow AI Platform® and the **Related Links** section of any table.

Accessing Form Builder from Table Builder

In Table Builder, access Form Builder on the **Forms** tab. For more information, see Forms in Table Builder.

Accessing Form Builder from the ServiceNow AI Platform

Customize any ServiceNow AI Platform record or form using Form Builder. On the form context menu, select **Configure**, and then select **Form Builder**.

Accessing Form Builder from Related Links

You can also access Form Builder from the Related Links section of any table. Open Form Builder to make edits to the form that you use to populate records for that table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Accessing Form Builder from UI Builder

You can access Form Builder directly from the UI Builder stage. Move your mouse device to a form component that has a set **Table** value and **View** value, then select the **Edit Form** button. Your changes appear on the form after closing Form Builder.

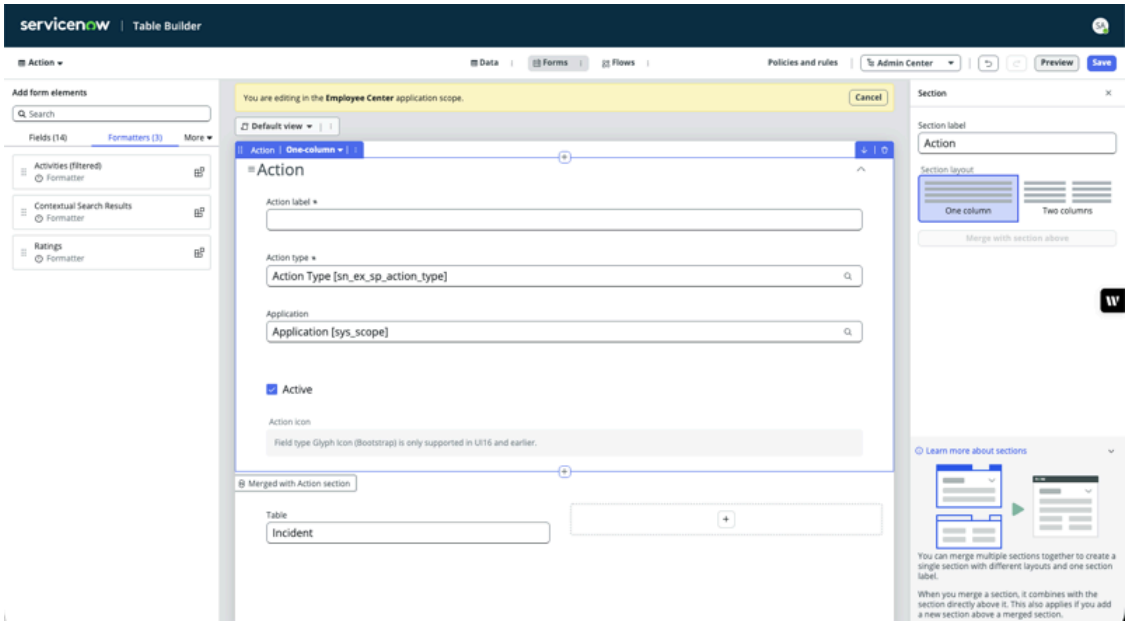


Table Builder workflow and navigation

You can start using Table Builder to manage your data tables, forms, or flows by reviewing this high-level workflow.

Learn about the high-level actions that you perform when you are using Table Builder.

Basic workflow

The following table outlines the basic actions that you take when working in Table Builder.

Basic workflow

Action	Description
Choosing a table or form to work with	To choose a data table or form to work with, launch Table Builder as instructed in Accessing Table Builder and choose the table or form to work with. **Note:** While using Table Builder in App Engine Studio, the table selected here is chosen when you click on a table in your app. However, if Table Builder is launched from UI Builder, you can choose a new table to work with from the drop-down list that displays the name of your currently selected table.
Selecting a domain and application scope	Select a domain and application scope to work within from Table Builder navigation.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Basic workflow (continued)

Action	Description
	Note: While using Table Builder in App Engine Studio, the application scope selected is the app you are working in. However, if Table Builder is launched from UI Builder, you can choose an application scope to work with from the drop-down list in the top right side of your screen (e.g., Global). See Domain separation and Table Builder and Using an application scope with Table Builder.
Modifying table properties	Select **Data** tab to edit table properties directly. For more information, see Data in Table Builder.
Customizing form views	Select **Forms** tab to create and customize new views or configure the form elements for a selected view. For more information, see Forms in Table Builder.
Configuring policies and rules	Select **Policies and rules** tab to configure and control how data is displayed for your form views. For more information, see Policies and rules in Table Builder.
Previewing a form and managing your changes	At any point in the process, you can preview the changes you've made by selecting **Preview**. For more information, see Preview your form. Save your form view changes at any point in the process by clicking **Save**. Your form view is ready and can be accessed by other applications in the ServiceNow AI Platform. **Note:** To undo ( ) or reapply ( ) changes you make, select the Undo or Redo icons at the top of your screen.

Navigational elements

The following table shows and describes the basic navigational elements shown in Table Builder.

Basic navigation elements

Navigational element	Description
Table selection list (Problem)	Choose an option to customize the policies and rules, table properties, fields, or form views for the selected table.
Select domain list	Choose an option to customize the policies and rules, table properties, fields, or form views within the selected domain (for example, the Global domain).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Basic navigation elements (continued)

Navigational element	Description
()	
Select scope list ()	Choose an option to customize the policies and rules, table properties, fields, or form views within the selected application scope (for example, the Global application scope).
Undo () Redo ()	Select the undo () or redo () icons in the navigation bar to revert to a previous editing state.
Advanced table properties menu ()	Choose an option: • **Advanced view** to manage the advanced properties in the selected table. For more information, see Table administration . • **Form designer** to launch the selected table in Form Designer. For more information, see Using the Form Designer . • **Delete table** to delete the selected table. **Note:** This option only appears on the Data tab.
Preview ()	Select **Preview** to review the selected form as how it is displayed in the ServiceNow AI Platform. For more information on previewing, see Preview your form.
Save ()	Select **Save** to commit your changes to your selection.
Search fields and filter options ()	Use the search box to filter the list of field columns by keyword or use the **Filter options** list to show inactive fields or to hide extended fields from the list (if a table has been extended from an existing table).
PDF extractor ()	If a PDF is associated with the selected data table, select this element to launch the PDF extractor tool and view the PDF. See Use a PDF to create data tables.

Domain separation and Table Builder

Domain separation enables you to separate data, processes, and administrative tasks into logical groupings called domains. You can control several aspects of this separation, including which users can see and access data.

Support level: Standard*

The support level is Standard but has some exceptions or special conditions.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Includes all aspects of **Basic** level support.
- Business logic: The service provider (SP) creates or modifies processes per customer. The use cases reflect proper use of the application by multiple SP customers in a single instance.
- The instance owner must be able to configure minimum viable product (MVP) business logic and data parameters. This configuration is done per tenant, as expected for the specific application.

Sample use case: An admin must be able to make comments required when a record closes for one tenant, but not for another.

For more information on support levels, see Application support for domain separation.

Overview of domain separation

Table Builder enables developers to configure the layout and logic for each form view for tables in their application. When launched from UI Builder, Table Builder supports domain separation, which is the ServiceNow instance-wide multi-tenant architecture.

Note: Domain separation is supported for Table Builder when it is launched from UI Builder. Domain separation is not supported when launching Table Builder from App Engine Studio.

Table Builder enables developers in domain-separated environments to create forms while they are in the same browser window. Domain separation in Table Builder works similarly to an application scope that helps administrators to create or edit in a multi-tenant environment.

Standardization is the key principle to maintaining a stable, healthy, and scalable ServiceNow instance, where domain separation is installed. By having standardization, you have a common configuration that most of the instance operates by. When an instance has hundreds or thousands of domains, managing them successfully requires rigorous governance. Domain-specific configurations should be done only if they are deemed necessary by the instance owners. Most instances should follow the common instance configuration to provide a more uniform experience across the instance. It also lets instance owners minimize the code sprawl that slows the adoption of new ServiceNow features that are included as part of the release upgrades.

How domain separation works in Table Builder

Table Builder enables developers to switch the session domain to create domain overrides to the form layout, section layout, and form logic. System dictionary record changes (such as field metadata for a field label) do not require domain overrides. Form layout changes require domain overrides to both the form record and the section when the change is performed at a lower domain. When creating a domain override, section labels can't be edited prior to saving the override.

Note: UI policies can have overrides at lower domains, as well. An override indicator appears on the UI policy.

In the global domain, a developer can expand the domain scope to see all form views from different domains. Developers can see all form views and views with overrides that are indicated with an icon.

Note: Single section form records do not have a *sys_ui_forms* record attached to it. Instead, these records only have an attached *sys_ui_section* record until a second section is created. If a single section form has existing domain overrides, and the developer adds another section to the form, the existing domain overrides will break.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Related topics

Domain separation for service providers

Using an application scope with Table Builder

Selecting an application from the application scope list in Table Builder enables you to customize form views for the users of the selected application in the ServiceNow AI Platform.

Application scoping protects applications by identifying and restricting access to application files and data. As an administrator, you can specify what parts of an application are accessible to other applications, which helps to protect data and application files. In Table Builder, you and your developers can set an application scope when you are working with a table or form.

Changing the scope in Table Builder also changes the scope in the ServiceNow AI Platform®, and changing the scope in the ServiceNow AI Platform® will change the scope in Table Builder.

When you are creating a form, you must be aware of the scope that you are working in. Choose the correct application scope for a table, form, or policy. The **Select scope** list is near the **Select Domain** list. The scope defaults to the scope that the user is in within the ServiceNow AI Platform®. If your users change to a different scope while in a page, they are notified that they are in a different application scope from the one that the form was created in.

Using Table Builder

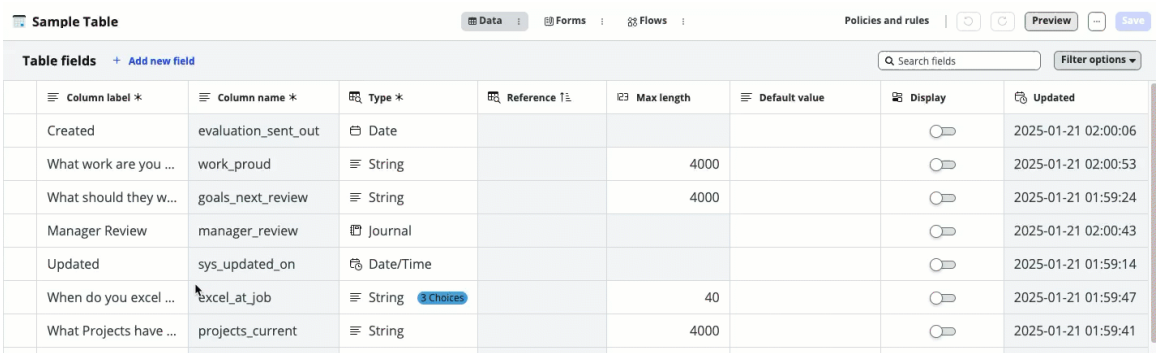
Use Table Builder, to design and customize tables, forms, and form elements for users from a single interface without needing to navigate between existing ServiceNow AI Platform tools.

Data in Table Builder

With Table Builder, you can modify the table properties and manage the fields that are available for a selected data table.

Table Builder is a tool for editing data tables. You edit a table on the **Data** tab by editing the columns of the table.

Table Builder



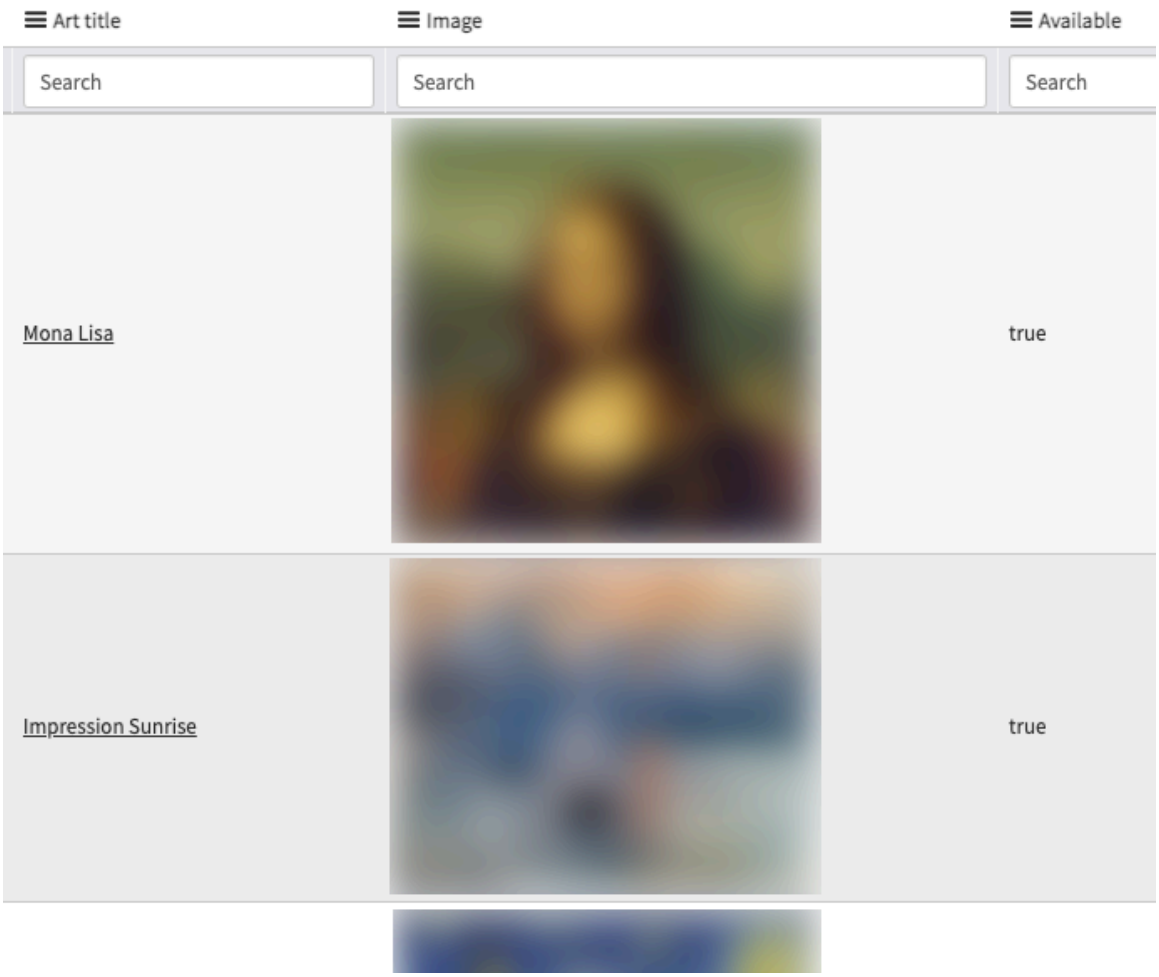
In Table Builder, each table column is represented as a row. For example, "Art title," "Image," and "Available" appear as column headers on the actual table. However, when you edit the columns in Table Builder, they appear as rows.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example: Columns in Table Builder

Column label * ▲	Column name *	Type *
Art title	art_title	String
Image	image	Image
Available	available	True/False

Example: Actual data table shown with two records

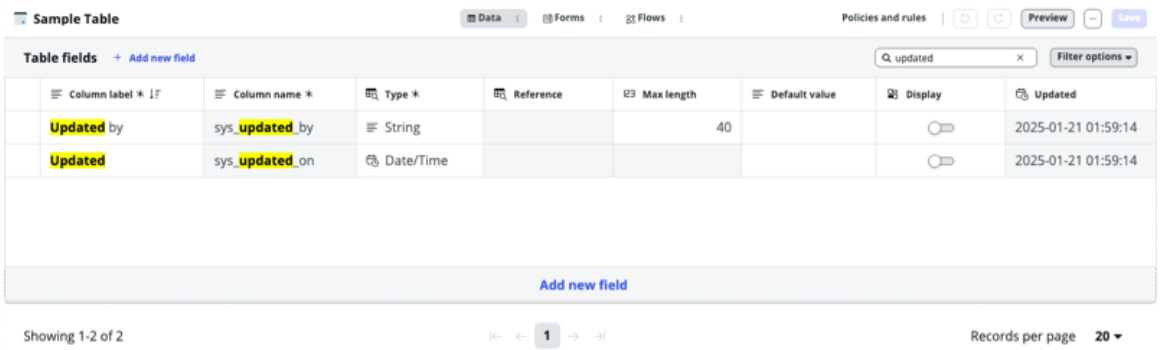


To easily find a specific table column to edit in the default Fields view, type the column name into the search box at the top right. As you type, the list of columns below is filtered to match the entered keyword.

Note: Use the **Filter options** list to show inactive fields or to hide extended fields from the list (if a table has been extended from an existing table).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Search for a table column



As you edit the table columns in Table Builder, think of each row as a field on a form. For example, if you change a column label, use a label that describes the data that you expect the user to enter in the form field. Change the column type to support this kind of form entry.

Modify table properties and table structure in **Fields** view, where you can perform the following actions:

- Add a new column to a table. See Add a table column in Table Builder.
- Delete a table column. See Delete a table column in Table Builder.
- Edit table properties. See Edit table properties in Table Builder.
- Change a column label. See Change a column label in Table Builder.
- Change a column name. See Changing a table column name in Table Builder.
- Change a column type. See Change a column type in Table Builder.
- Add a default value to a table. See Add a default value to a table column in Table Builder.
- Select a column as table display name. See Select a column as the table display value in Table Builder.
- Delete a table. See Delete a table in Table Builder.

Work with table structure and your data records in spreadsheet format within **Spreadsheet** view, where you can perform the following actions:

> **Note:** This feature is only available if your licensing entitles you to "exclusive low code capability" and you have Table Builder for App Engine installed. Contact your Solutions consultant for more information.

- Add a new data record. See Add a new data record.
- Edit individual data records. See Edit individual data records.
- Sort your data records. See Sort your data records.
- Filter your data records. See Filter your data records.
- Edit column field properties. See Edit column field properties.
- Add new columns. See Add new columns.
- Duplicate an existing column. See Duplicate an existing column.
- Reorder columns. See Reorder columns.
- Set column visibility. See Set column visibility.
- Pin columns. See Pin columns.
- Delete a table column. See Delete columns.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

You can also view the table relationships for your table in **Schema** view. See Schema view in Table Builder.

Note: This feature is only available if your licensing entitles you to "exclusive low code capability" and you have Table Builder for App Engine installed. Contact your Solutions consultant for more information.

Spreadsheet view in Table Builder

Use **Spreadsheet** view in Table Builder to work with your application data records and data structure in a spreadsheet format.

Overview of the Spreadsheet view

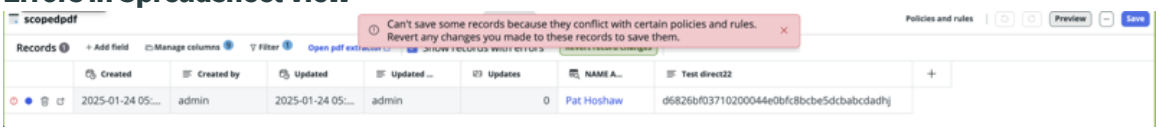
Spreadsheet view in Table Builder, enables you to review your application tables and their accompanying data in a familiar spreadsheet format. You can add columns, delete columns, and change column properties as well as review and update individual data records within those columns from a single screen.

Saving data records in Spreadsheet view

Sometimes errors occur when trying to save data records (data conflicting with applicable policies or business rules). When this happens, a warning popup displays along with a filtered list of erroneous data records.

Correct the data in each record individually, or select **Revert record changes** to revert any changes you made to the erroneous records.

Errors in Spreadsheet view



Use the following procedures to edit table field structure and any imported data records for a selected table using **Spreadsheet** view in Table Builder

Add a new data record

Add a new data record within Table Builder **Spreadsheet** view.

Before you begin
Role required: personalize_dictionary or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

Procedure

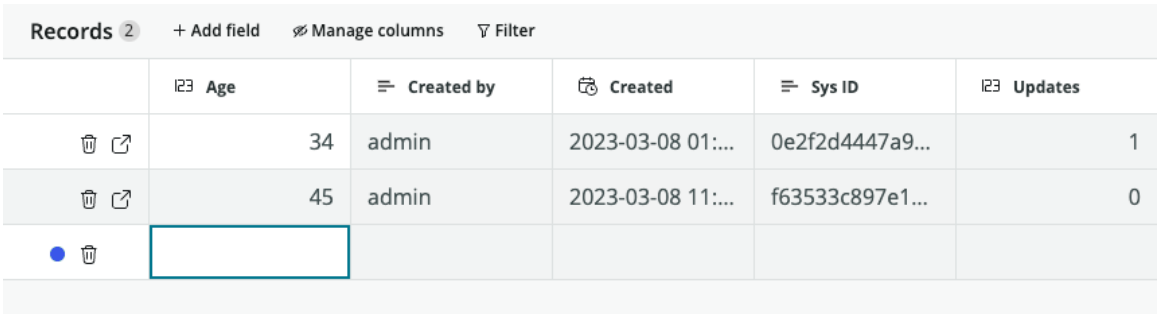
1. Navigate to the **Data** tab.

   Note: Spreadsheet view displays by default. You can also access Spreadsheet view from the Additional actions menu ( ).

2. Select the **Add new record** icon ( ).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add new record



3. Make your changes to the data record.
4. Select **Save**.
   See Saving data records in Spreadsheet viewfor details on troubleshooting errors.

Edit individual data records

Edit individual data records within Table Builder **Spreadsheet** view.

Before you begin

Role required: personalize_dictionary or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

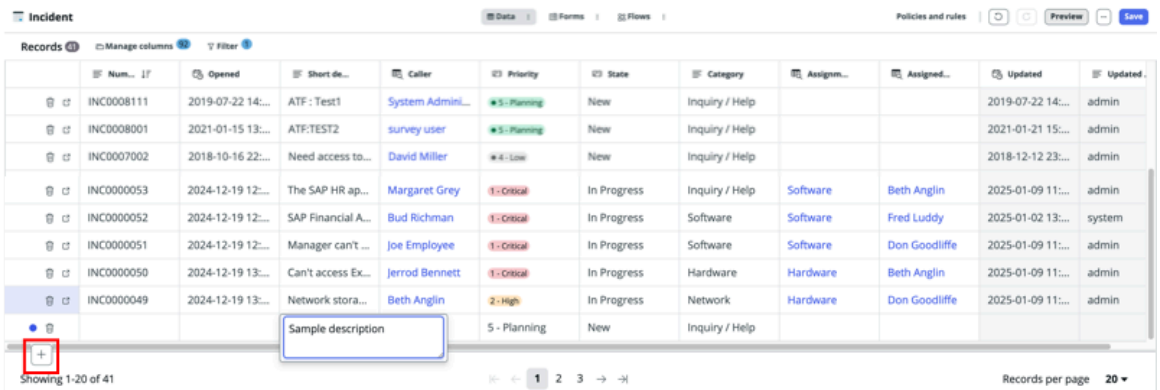
Procedure

1. Navigate to the **Data** tab.

   Note: Spreadsheet view displays by default. You can also access Spreadsheet view from the Additional actions menu ( ).

2. Click into any editable data record field to launch the data for editing.

Edit a field



3. Make your changes to the data.
4. Select **Save**.
   See Saving data records in Spreadsheet viewfor details on troubleshooting errors.

Sort your data records

Sort your data records by a specific field column within Table Builder **Spreadsheet** view.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

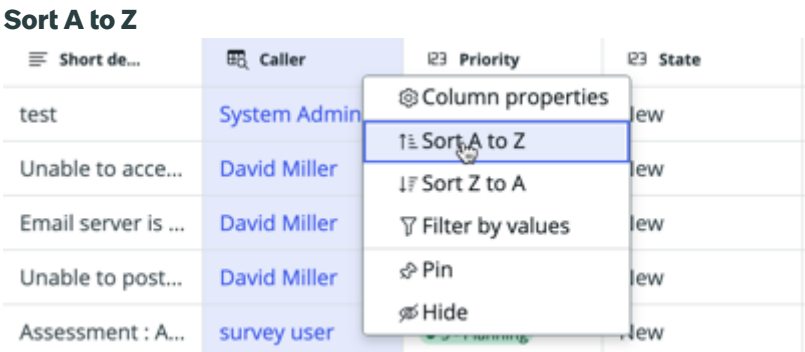
Role required: personalize_dictionary or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

Procedure

1. Navigate to the **Data** tab.

   Note: Spreadsheet view displays by default. You can also access Spreadsheet view from the Additional actions menu ( ).

2. Hover over a column header that contains the data by which you would like to sort the list of records, and then select the **Additional actions** icon ( ) to launch a column-specific list of actions (e.g., sort records by Approval Date in ascending order).
3. Select from the following options to choose whether to sort the data for the column in ascending or descending order.
   - Sort A to Z or Sort Oldest to Newest (ascending order)
   - Sort Z to A or Sort Newest to Oldest (descending order)



The list of records will then display in the order you chose.

Show matching data records

Show a filtered list of data records that contain matching data for a selected field within Table Builder **Spreadsheet** view.

Before you begin

Role required: personalize_dictionary or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

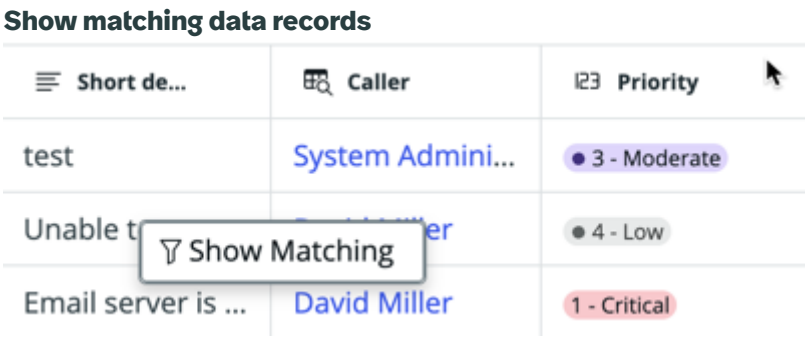
Procedure

1. Navigate to the **Data** tab.

   Note: Spreadsheet view displays by default. You can also access Spreadsheet view from the Additional actions menu ( ).

2. Select and hold (or right-click) the data record in a cell to create a filtered list of records that contain the matching data, (e.g., show all records where **Short description** is "Can't read email".).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



A filtered list of records with matching data displays and a new filter query is added and accessible when you click the **Filter** button at the top of the grid.

Note: To remove the filtering you just added, select the **Filter** button, and then delete the new filter condition from the list. See Filter your data records.

Filter your data records

Display only the data records you want to see in your list by setting up a conditional query within Table Builder **Spreadsheet** view.

Before you begin

Role required: personalize_dictionary or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

Procedure

1. Navigate to the **Data** tab.

   Note: Spreadsheet view displays by default. You can also access Spreadsheet view from the Additional actions menu ( ).

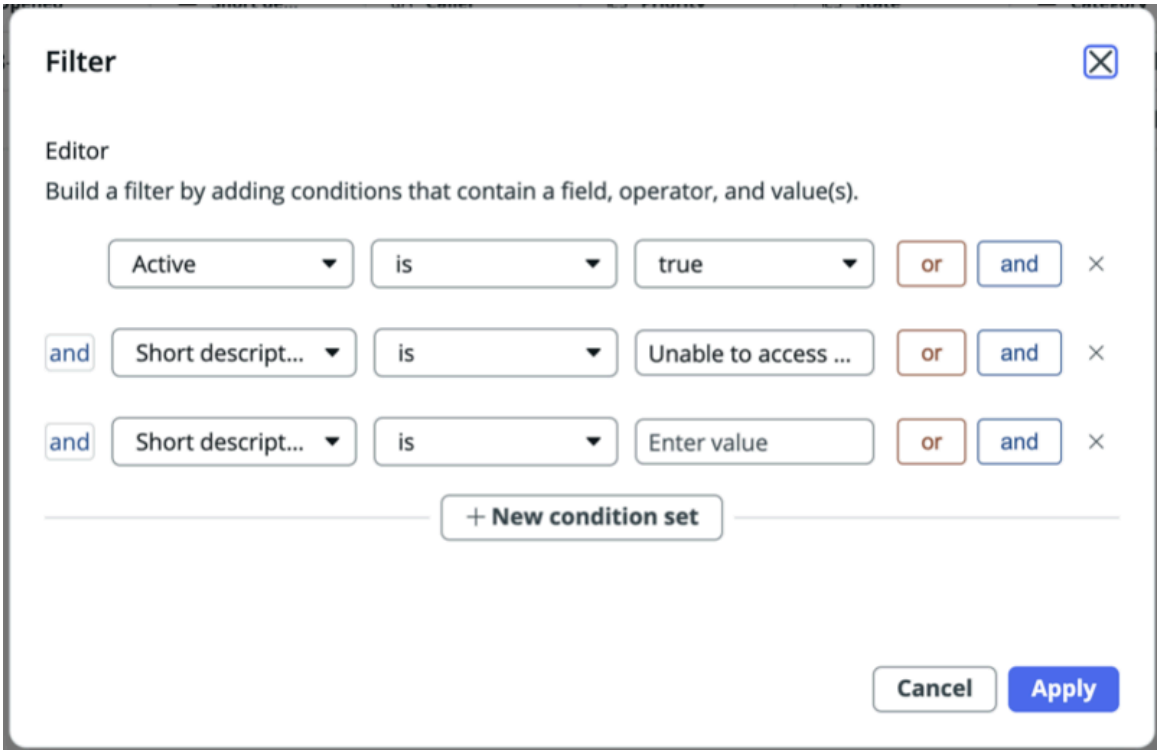
2. Select **Filter** above the spreadsheet to launch a filter query.
   You can also select **Filter by value** from a specific column's **Additional actions** menu ( ) to pre-select that column's data for your query.

   Note: When a number appears to the right of the **Filter** button, it means that there are existing filters set for the view.

   The Filter window displays (along with any filter criteria that has already been set). In this case, an existing filter is set to display records where **Age** is 35.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Filter data



3. Enter your filtering query to specify which data records you want to display.
   - Select the **New condition set** button to enter a new filter condition which is connected by an OR statement.
   - Select the **or** or **and** buttons to extend the existing filter set with the desired boolean.
   - Select the **Delete** icon to remove a filter condition row.
4. Select **Apply** to apply the filter condition to your view.
   The filtered list of records will then display based on the criteria you chose.

Edit column field properties

Edit field properties such as the field label, type, attributes, and default values within Table Builder **Spreadsheet** view.

Before you begin

Role required: personalize_dictionary or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

Procedure

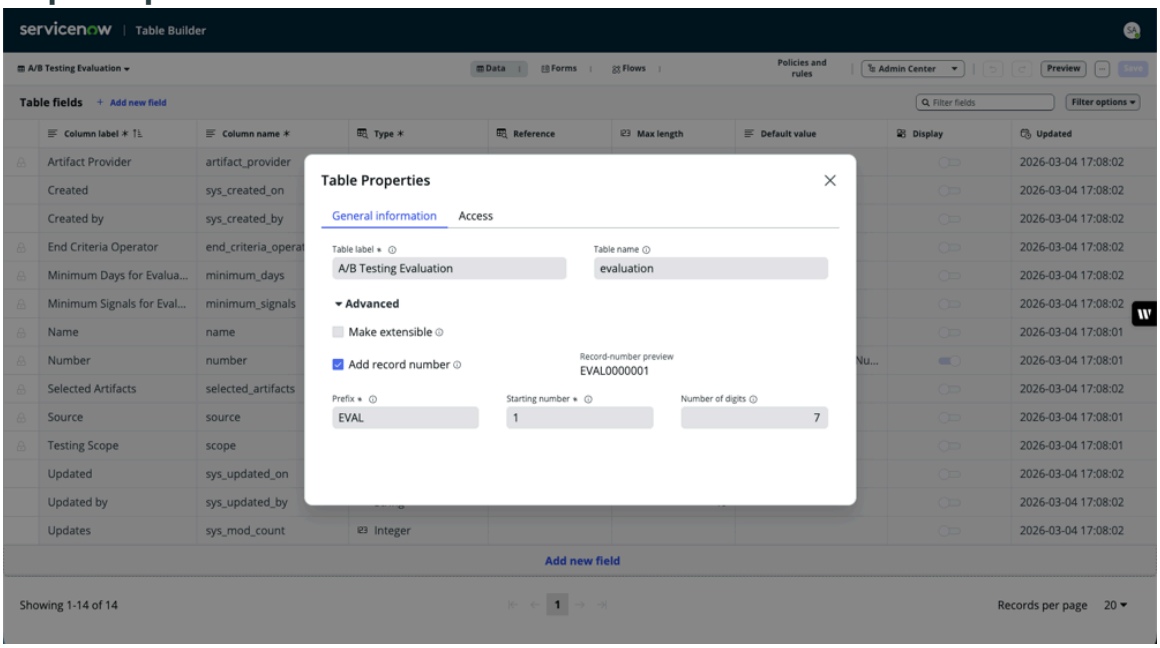
1. Navigate to the **Data** tab.

   Note: Spreadsheet view displays by default. You can also access Spreadsheet view from the Additional actions menu ( ).

2. Hover over the desired column header, and then select the **Additional actions** icon ( ) to launch a column-specific list of actions.
3. Select **Column properties**.
   The Properties pane displays for the selected field column.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties pane



4. Edit the desired field property for the column.
   See Field configuration in Table Builder for more information on field properties.
5. Select **Save**.

Add new columns

Add new field columns to a selected table within Table Builder **Spreadsheet** view.

Before you begin

Role required: personalize_dictionary or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

Procedure

1. Navigate to the **Data** tab.

   Note: Spreadsheet view displays by default. You can also access Spreadsheet view from the Additional actions menu ( ).

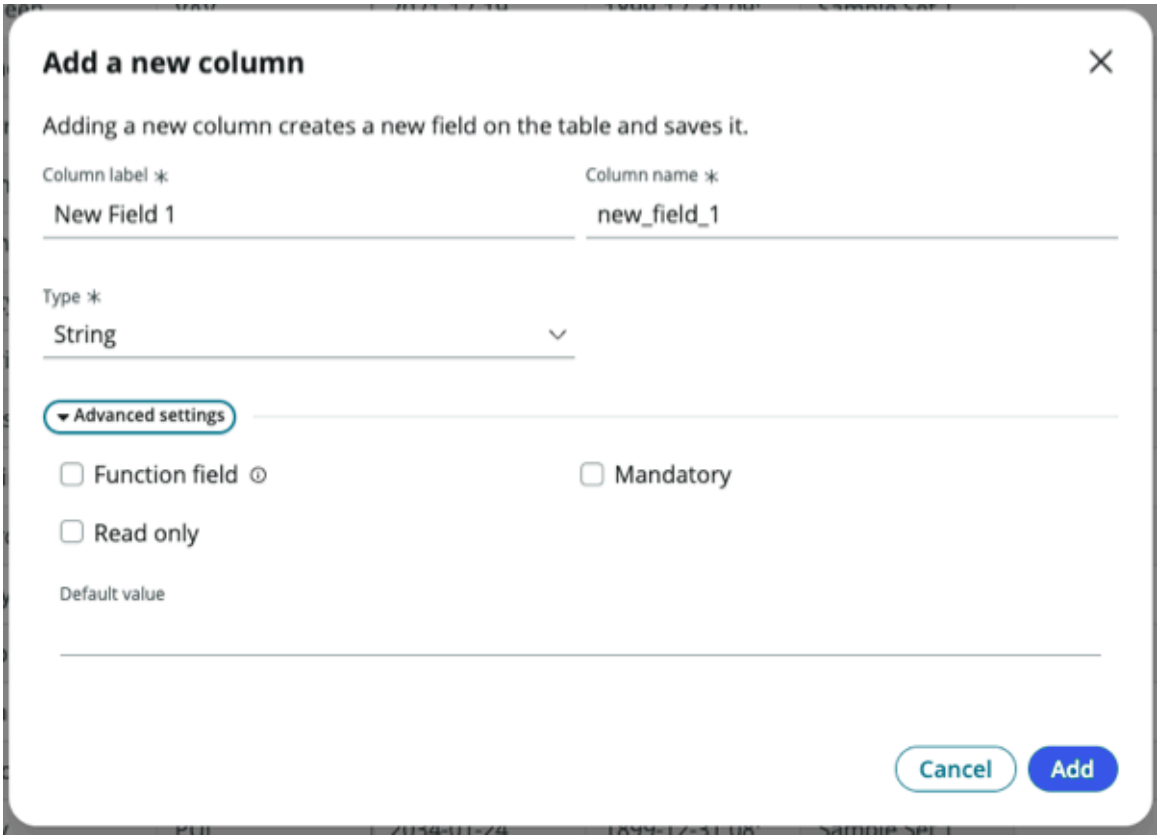
2. Select the **+ Add field** button.

   Note: You can also select the **Additional actions** icon ( ) to launch a column-specific list of actions, and then select **Insert column left** or **Insert column right** to insert the new column in a specific place within the table.

   The Add a new column window displays.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add new column



3. Enter the following information for your new field column.

Field	Description
Column label	Unique label for the column.
Column name	Database name for the column.
Type	Type of information that the column contains. For example, to contain plain text in the col umn, select **String**. Depending on the type that you select, fill in the additional fields to further define the table column. For example, if you select **String**, de fine the character limit of the string input. Or, if you select **Choice**, define the choices that users can choose from. See Field types.

4. **Optional:** Select **Advanced settings** to expand it and enter any additional properties for your new field column.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Function field	(Optional) A function field generates a value based on simple computations of other fields and constants. The **Function definition** field then displays, where you define a function.
Mandatory	(Optional) Option that you can select so that the data that your users enter in the field is required before they can save their changes to the form.
Read only	(Optional) Option that you can select so that the field is not editable by the form users.
Default value	(Optional) String or number that represents the default value of the field. This will be auto-populated in all rows of your column once it is created.

5. Select **Add** to add the new field column to your table.

Duplicate an existing column

Duplicate an existing field column within a table shown in Table Builder **Spreadsheet** view.

Before you begin

Role required: personalize_dictionary or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

Procedure

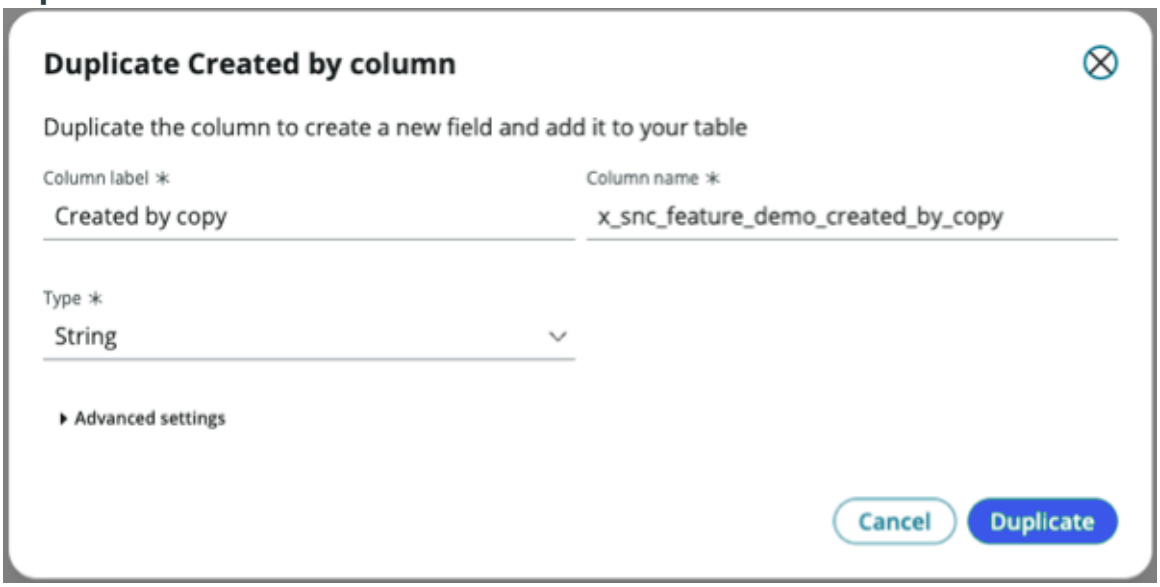
1. Navigate to the **Data** tab.

   Note: Spreadsheet view displays by default. You can also access Spreadsheet view from the Additional actions menu ( ).

2. Hover over the header for the column to duplicate, and then select the **Additional actions** icon ( ) to launch a column-specific list of actions.
3. Select **Duplicate column**from the menu.
   The text "copy" is automatically appended to the existing column label.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Duplicate column



4. Edit the **Column label** to the desired display name for the duplicated field column.
5. Edit the **Type** for the field column as desired.

   See Field types.
6. Select **Duplicate**.
   A new column displays in the view.

Reorder columns

Change the order in which your columns are displayed within Table Builder **Spreadsheet** view.

Before you begin

Role required: personalize_dictionary or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

Procedure

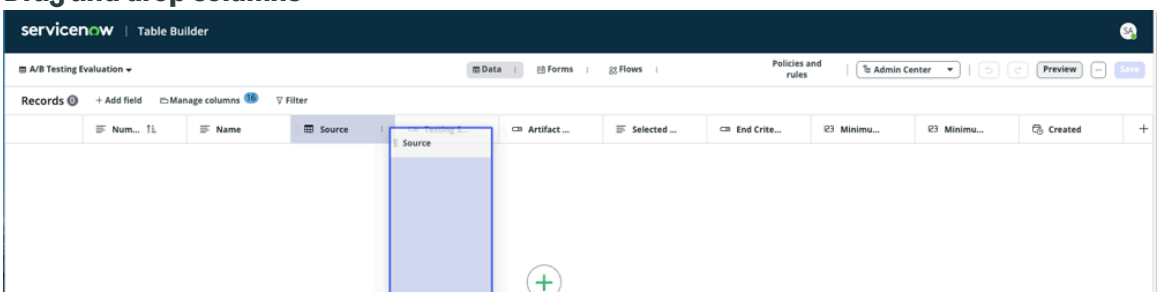
1. Navigate to the **Data** tab.

   Note: Spreadsheet view displays by default. You can also access Spreadsheet view from the Additional actions menu ( ).

2. Click to select the **Column handle** icon ( ) for a column and then drag the column to the desired locations in the table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Drag and drop columns



Note: If your table contains many table columns, select the **Manage columns** menu to view a compact list of column names, and then visually drag and drop within this list to order your table columns.

Set column visibility

Choose whether to hide or show specific field columns within Table Builder **Spreadsheet** view.

Before you begin

Role required: personalize_dictionary or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

Procedure

1. Navigate to the **Data** tab.

   Note: Spreadsheet view displays by default. You can also access Spreadsheet view from the Additional actions menu ( ).

2. Choose one of the following options.

Action	Steps
Hide a column	Using the **Manage columns** button: **a.** Select the **Manage columns** button to launch a window that displays the list of available columns. **b.** Deselect the column you wish to hide. Only selected columns will display in your view. Using the column's context menu: **a.** Hover over the column header to hide, and then select the **Additional actions** icon ( ). **b.** Select **Hide** from the menu to hide the col umn from view.
Unhide a column	Using the **Manage columns** menu:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Action	Steps
	a. Select the **Manage columns** button to launch a window that displays the list of available columns. **b.** Select the checkbox to the left of the col umn you wish to unhide. Only selected columns are displayed in your view.

Note: A number displays to the right of the **Manage columns** button to show how many columns are active in the view.

Pin columns

Pin specific columns to always display on the left when you scroll to the right in a table with a large number of field columns within Table Builder **Spreadsheet** view. Pinned columns are saved for each individual user, not globally. The next time a user logs in, any columns they pinned will be shown.

Before you begin

Role required: personalize_dictionary or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

Procedure

1. Navigate to the **Data** tab.

Note: Spreadsheet view displays by default. You can also access Spreadsheet view from the Additional actions menu ( ).

2. Choose one of the following options to pin or unpin a field column.

Action	Steps
Pin a column	**a.** Hover over the column header to pin, and then select the **Additional actions** icon ( ). **b.** Select **Pin** from the menu to pin the column to the left side of your view.
Unpin a column	**a.** Hover over the column header to unpin, and then select the **Additional actions** icon ( ). **Note:** Pinned columns have a check mark to the left of the column in the list. **b.** Toggle **Pin** off to unpin the pinned column.

Note: You can also view which columns are pinned in the view by clicking the **Manage columns** button. Pinned columns will have a pin icon displayed next to the right of the listed column.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Delete columns

Delete a field column for a selected table within Table Builder **Spreadsheet** view.

Before you begin

Role required: personalize_dictionary or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

Procedure

1. Navigate to the **Data** tab.

   Note: Spreadsheet view displays by default. You can also access Spreadsheet view from the Additional actions menu ( ).

2. Hover over a column header you want to delete, and then select the **Additional actions** icon ( ) to launch a column-specific list of actions (e.g., sort records by Approval Date in ascending order).
3. Select **Delete column**.
   A confirmation window displays.
4. Select **Delete** to delete the field column from your table.

   Note: If there are data records stored for the column you want to delete, you will not be able to delete the column until you have removed the records stored for this field.

Fields view in Table Builder

Modify the table properties and manage the table fields by using the default Fields view on the **Data** tab in Table Builder.

Before you begin

- Launch Table Builder. For more information, see Accessing Table Builder.
- (Optional) Choose a domain to work within (if not global). For more information, see Domain separation and Table Builder.
- (Optional) Choose an application scope to work within (if not global). For more information, see Using an application scope with Table Builder.

Role required: personalize_dictionary or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

Procedure

Use the following procedures to edit tables using **Fields** view in Table Builder.

Add a table column in Table Builder

Store more information in a table record by adding a table column.

Before you begin

Role required: personalize_dictionary or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

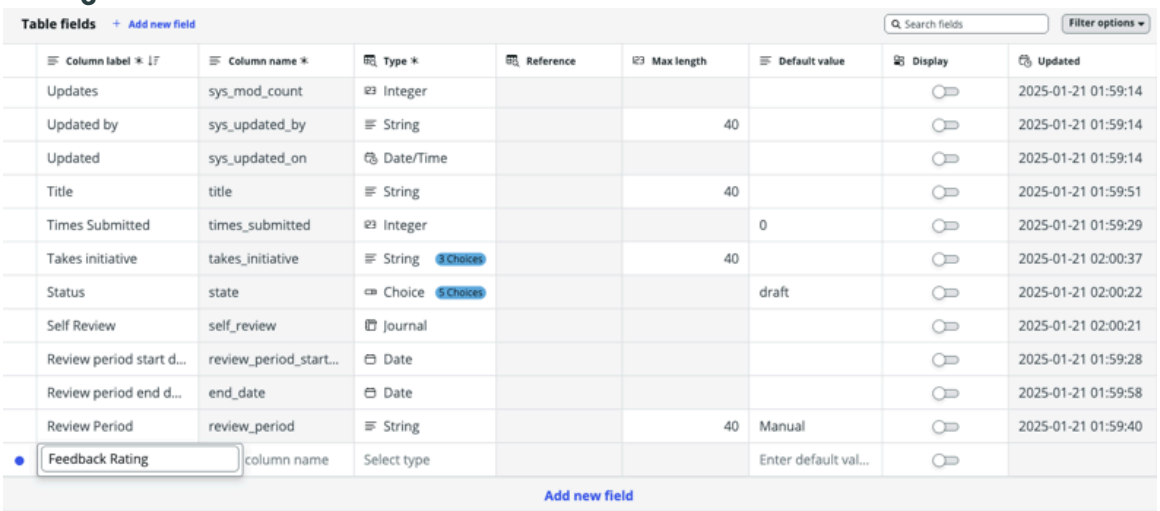
Procedure

1. Navigate to the **Data** tab.

   Note: Fields view displays by default. You can also access the Fields view from the Additional actions menu ( ).

2. Select **+ Add new field**.
3. On the resulting blank row, fill in the desired properties for the new data table column.

Adding a new field to a table



Field	Description
Column label	Unique label for the column.
Column name	Database name for the column.
Type	Type of information that the column contains. For example, to contain plain text in the col umn, select **String**. Depending on the type that you select, fill in the additional fields to further define the table column. For example, if you select **String**, de fine the character limit of the string input. Or, if you select **Choice**, define the choices that users can choose from. See Field types.
Reference	Table that is associated with the column. This field applies only if the column type is **Refer ence**.
Max length	Maximum number of characters that users can enter in the field.
Default value	Value that populates the field automatically af ter a new record is created.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Display	Option to set the column as the display value for the table. A reference field shows the dis play value of the table to which it is referring. For example, the **Opened by** column of the task table refers to the user table. Because the display value of the user table is the user name, the **Opened by** field shows something like **Beth Anglin** or **Joe Employee**. When you select a display value, choose the table col umn that would act as an appropriate title for individual records. Only one column can act as the display value for a table.
Updated	Information about when the table was last up dated.

4. Select **Save**.

Result

If you preview the table, you can see the new column that was added to your table. For more information on previewing a table, see Preview your data in Table Builder.

Delete a table column in Table Builder

Delete a column from your table using Table Builder.

Before you begin

Role required: personalize_dictionary or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

Procedure

1. Navigate to the **Data** tab.

   Note: Fields view displays by default. You can also access the Fields view from the Additional actions menu ( ).

2. Next to the table field row you want to delete, hover over the leftmost column, and select the Open side panel icon ( ).

3. At the top right corner of the side panel, select the trash icon ( ) to delete the field.

   Note: Your table includes several default columns (including Created and Updated), that you can't delete. Also, you can't delete columns that are extended from another table or columns where data records are already present. To delete the column, you must first delete the existing data.

4. Select **Delete** in the confirmation dialog.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Result
If you preview the table, you can see that the table column is no longer used in the table. For more information on previewing a table, see Preview your data in Table Builder.

Change a column label in Table Builder

Modify the label that uniquely identifies the data that is stored in it to users.

Before you begin
Role required: personalize_dictionary or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

About this task
A column label is the text that's visible to your users when they're reading or updating application data. For example, if you create a form for requesters to fill in, each column label appears as a form field label.

Use a column label that describes the specific information that you expect users to enter. For example, to prompt a user to add comments, you would use the column label "Comments" instead of "Text".

Procedure
1. Navigate to the **Data** tab.

   Note: Fields view displays by default. You can also access the Fields view from the Additional actions menu ( ).

2. Change a column label in the Column label column, by updating the text in the corresponding row.

   For example, to change the label of the "Office location" column, you would select the cell that says "Office location" and enter a different label.

   The column label updates automatically after you select another cell.

3. Select **Save**.

Changing a table column name in Table Builder

Change the database name for the column. You may opt to change the column name manually if a reviewer finds that the column name in the database was causing issues in testing.

Before you begin
Role required: personalize_dictionary or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

About this task
Note: You can't change the column name for existing fields that have information in them saved already.

A column name is the text that an administrator uses to track column data in the database. A column name is created automatically after you enter a column label. In Table Builder, if you change the column label, the column name also changes automatically.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to the **Data** tab.

   Note: Fields view displays by default. You can also access the Fields view from the Additional actions menu ( ).

2. Change the column name in the Column name column by updating the text in the corresponding row.

   For example, to change the name of the "Office location" column, you would select the cell that says u_office_location and enter a different name.

   The column name updates automatically after you select another cell.

3. Select **Save**.

Change a column type in Table Builder

Change the type of field that will store information for rows in a table column. For example, if your table is tracking dates, you would want to select the Date column type.

Before you begin

Role required: personalize_dictionary or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

Procedure

1. Navigate to the **Data** tab.

   Note: Fields view displays by default. You can also access the Fields view from the Additional actions menu ( ).

2. Select a cell and clear the text from the **Type** column.
3. From the list, select a column type.

   You can search the list by entering new text. For example, to find all the string types, you would enter `string`.

4. If required, define the column type with more information.

   For example, if you select **Choice**, define the list of choices that users can select from.

   The column type updates automatically after you select another cell.

5. Select **Save**.

Edit field properties in Table Builder

Edit field properties such as the field label, type, and default values in Table Builder.

Before you begin

Launch Table Builder. For more information, see Accessing Table Builder.

Role required: personalize_dictionary or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to the **Data** tab.

   Note: Fields view displays by default. You can also access the Fields view from the Additional actions menu ( ).

2. Select a field you want to modify in the table.

   For information about basic field properties, see Field Configuration.

3. Next to the field you want to edit, hover over the leftmost column, and select the Open side panel icon ( ).

4. Make your changes to the field properties.

   For information about modifying a field's UI policies, see Modify a UI policy in Table Builder.

   Note: The **Config** tab on the right stays open even if you switch between the **Data** and **Forms** tabs.

5. Select **Save**.

Add a default value to a table column in Table Builder

Define the default value to populate a table column automatically after a user creates a record.

Before you begin

Role required: personalize_dictionary or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

Procedure

1. Navigate to the **Data** tab.

   Note: Fields view displays by default. You can also access the Fields view from the Additional actions menu ( ).

2. Add the default value of a column in the **Default value** column by updating the content in the corresponding row.

   You define the default value by selecting a cell and then entering the value. The value that you enter depends on the column type. For example, if the column requires a choice, you would select one of the choices that you defined in the Type column.

   The column name updates automatically after you select another cell.

3. Select **Save**.

Select a column as the table display value in Table Builder

Select a column value that will be displayed in the reference fields of other tables. For example, the Opened by column of the task table could refer to a specific column in the user table so it can display a user name.

Before you begin

Role required: personalize_dictionary or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

About this task

A reference field shows the display value of the table to which it is referring. For example, the Opened by column of the task table refers to the user table. Because the display value of the user table is the user name, the **Opened by** field shows something like "Beth Anglin" or "Joe Employee". When you select a display value, choose the table column that would act as an appropriate title for individual records.

Only one column can act as the display value for a table.

Procedure

1. Navigate to the **Data** tab.

   Note: Fields view displays by default. You can also access the Fields view from the Additional actions menu ( ).

2. Check that all selections are cleared from the **Display** column.
   To clear a selection, toggle off the switch as shown in the following example.

   Toggle off

3. From the Display column, select one row to act as the table display value.
   To select a row, toggle on the switch as shown in the following example.

   Toggle on

4. Select **Save**.

Schema view in Table Builder

Use **Schema** view in Table Builder to explore data relationships for your application data.

Before you begin

- Launch Table Builder. For more information, see Accessing Table Builder.
- (Optional) Choose a domain to work within (if not global). For more information, see Domain separation and Table Builder.
- (Optional) Choose an application scope to work within (if not global). For more information, see Using an application scope with Table Builder.

Role required: personalize_dictionary or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

Procedure

1. Navigate to the **Data** tab.

   Note: Spreadsheet view displays by default.

2. **Optional:** Select **Schema** from the additional actions menu ( ) to access Schema view.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

3. Navigate through the schema diagram by dragging the canvas with your mouse.
4. You can perform the following additional actions on this read-only screen.

Option	Description
Navigation overview controls	Select the **Overview toggle** icon ( ) in the lower right corner to toggle the navigational overview controls on or off. This navigational control enables you to perform the following actions: ○ When the schema diagram is zoomed in, you can drag the rectangle showing your view around by clicking and dragging it to focus it on the part of the diagram you want to review. ○ Select the **Zoom in** icon ( ) to zoom in. ○ Select the **Zoom out** icon ( ) to zoom out.
Show / hide table fields	The **Expand tables** toggle is toggled on by de fault to show all the fields within the displayed schema tables. To hide the fields within the displayed schema tables, select **Expand ta bles** to toggle it off.
Show / hide tables	Use the **Items selected** menu to view a list of application and non-application tables that are displayed in the schema diagram. You can select the table name to remove the selection checkmark, which hides the table from dis playing in the diagram. To show the table, se lect the table name again to toggle the check mark back on.
Set field options	Use the **Field options** menu to control how fields display in the schema diagram. ○ Select **Extended fields** to display extended fields within schema tables. ○ Select **Field type** to show the type of field for each field displayed in the schema ta bles.
Group fields	Use the **Group fields by** menu to group fields within the displayed tables (e.g., grouping fields by type). Choose from the following grouping options: ○ No group ○ Extended ○ Reference ○ Type
Sort fields	Use the **Sort fields** menu to sort fields within the displayed tables. Choose from the follow ing sorting options:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Description
	○ Alphabetically (A to Z) ○ Alphabetically (Z to A)
View legends	Select **View legends** to view information for in terpreting the displayed schema diagram.
Table-specific options	Select the **Additional options** icon (⋮) for a ta ble to control how fields in the selected table are displayed, grouped, and/or sorted similar to the top-level options described above.
View in Table Builder	Select the **View Table** icon (↗) to view the se lected table in Table Builder.

Preview your data in Table Builder

See what your table looks like after you've edited the table columns.

Before you begin

- Launch Table Builder. For more information, see Accessing Table Builder.
- (Optional) Choose a domain to work within (if not global). For more information, see Domain separation and Table Builder.
- (Optional) Choose an application scope to work within (if not global). For more information, see Using an application scope with Table Builder.

Role required: none

About this task

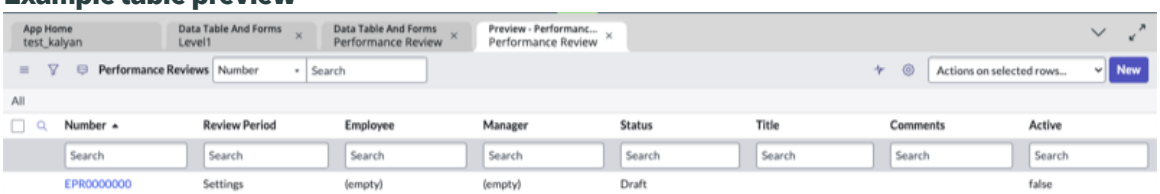
Each row in Table Builder represents a column from your table. Previewing the table enables you to see what the columns look like in their proper place. You can also test the experience of creating records on your table.

Note: The table which you are editing must have at least one role that is associated to it or preview is disabled.

Procedure

1. On the **Data** tab, select **Preview**.
2. From the new tab, review the table preview as shown in the following example.

Example table preview



What to do next

If your table doesn't work as expected, continue editing in Table Builder. Then, preview the table again.

Delete a table in Table Builder

Delete a table from your application so that it no longer appears in the app dashboard.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

Note: Parent tables can't be deleted until all existing child tables are deleted.

Role required: Users must have canDelete access to **sys_db_object** or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

Procedure

1. Navigate to the **Data** tab.
2. Select the menu icon (•••), and then select **Delete Table**.
3. In the dialog box, enter `delete` and then select **Delete**.
4. Select **Save**.

Result

The table is no longer available in the app dashboard.

Edit table properties in Table Builder

Change table properties such as the table label or other settings so that you can make a table extensible or add record numbers by using Table Builder.

Before you begin

Role required: none

Note: User must have canWrite access to *`sys_db_object`* or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

About this task

By editing table properties, you can relabel your table, make your table extensible, or add record numbers. You can also set application access settings.

Making a table extensible means that you are enabling new tables to share columns from your table. For example, if you created an "Office location" column in your table, you can allow new tables to use the "Office location" column also. For more information on table extensions, see Table extension and classes.

Adding record numbers means that a tracking number is created automatically for each new table entry. For example, a new entry to the Travel Requests table would get a record number like TRV1234567. You can use this number to find table records more easily.

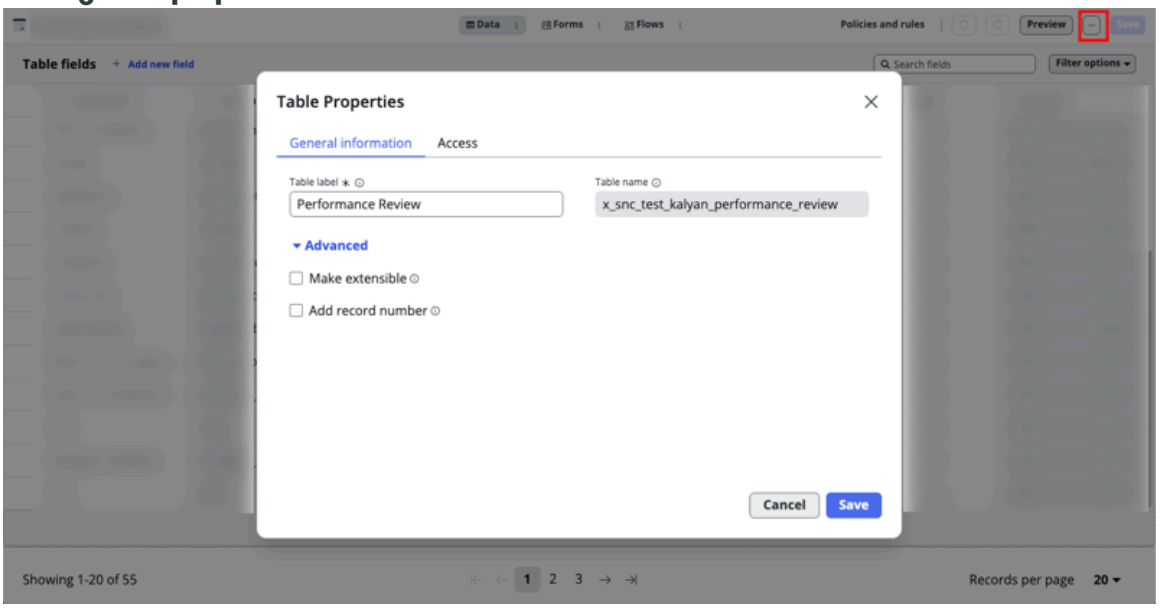
Application access settings determine whether script objects from other applications can access the table in your application. You can give these script objects access to read, create, update, or delete records on your table. Alternatively, you can disable access to your table from other applications. For more information on the application access controls, see Table design and runtime settings.

Procedure

1. In the header, navigate to the more options button ⋯.
2. Select **Properties**.
3. In the dialog box that appears, update the table properties as shown in the following example. For more information on these properties, see Table properties.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Editing table properties



4. To close the dialog box, select **Save**.
5. Select **Save**.

Forms in Table Builder

On the **Forms** tab in Table Builder, you can visually create, configure, and customize the different form views for your users using the form editor. The views that you define contain the elements that appear when a user opens that form or list.

Form View

A form view defines the elements that appear to a user when the user opens a form or list. Each form has a default view, but you may want to create several different form views for different groups of users.

In Table Builder, you can customize multiple views for a form for the different users who input data or view the form. You can also customize the default form view for everyone.

Explore the following major areas when you configure a form or list:

- Form view selection
- All form views list
- Form layout
- Form fields
- Form annotations
- Formatters
- Embedded lists

Basic form navigation

The **Forms** tab consists of the following four major areas:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Top navigation menu
- Add form elements panel
- Form editor
- Configuration panel

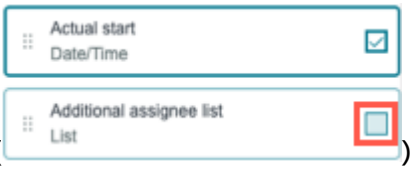
The following table contains a list of some of the other basic navigational elements within the **Forms** tab in Table Builder.

Navigational elements in the Forms tab

Navigational element	Description
Forms additional actions list ()	The Additional actions list (vertical ellipsis) on the **Forms** tab provides a quick way to perform any of the following actions: • Add a new form view • Duplicate the selected form view • Configure a related list • Configure a UI action • Navigate to a filterable, sortable list of all form views by selecting **All form views**. See All form views list. Changes to the form made in the editor are committed to the selected view. **Note:** The more options button is visible only when forms are accessed through ServiceNow Studio or App Engine Studio and not through ServiceNow AI Platform.
Form view list ()	Select a form view from this list or use the Additional actions menu (vertical ellipsis) to add a view. Changes to the form made in the editor are committed to the selected view.
PDF extractor ( )	If a PDF is associated with the selected data table, select this element to launch the PDF extractor tool and view the PDF. See Use a PDF to create data tables. **Note:** This feature is only available if your licensing entitles you to "exclusive low code capability" and you have Table Builder for App Engine installed. Contact your Solutions consultant for more information.
Form element search bar	Enter keywords here to filter the list of form elements.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Navigational elements in the Forms tab (continued)

Navigational element	Description
()	
Multiple selection check box 	When you point to a form element in the Add form elements panel, you can select this option if you want to select and drag multiple fields to the form editor at once.
Add form element icon ()	When you point to a form element or section, select this icon to add a new section or element above or below the existing element.
Move element icon ()	When you point to a form element, this icon displays on the left edge. Select this icon, and then drag the element to move it to the desired location.
Delete form element icon ()	When you point to a form element in the editor, you can select this icon to remove it from the form view.
Move section arrow icon ()	When you select a form section, you can select the up or down arrow to move it up or down on the form view.
Delete section icon ()	When you select a form section, you can select this icon to remove the section from the form view.
Open panel icon () Close panel icon (✕)	Selecting the Open panel icon expands the panel to the right. Selecting the Close panel icon collapse it.

Form view selection

Select a form view to work with from the form view list in the form editor.

Note: You can also select a form view from the Additional actions menu on the **Forms** tab.

See Choose a form view in Table Builder.

All form views list

Select **All form views** from the Additional actions menu to the right of the **Forms** tab to browse a filterable list of form view cards (vertical ellipsis). This page allows you to create complex filtering and sorting if you have a great deal of form views to navigate.

You have the following options for finding a desired form view if you have a large list of them.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

All form views list

Option	Steps
Search	Type a keyword in the **Search** box. The list of cards will be filtered as you type.
Sort form views	Select an option from the sorting list to sort the form view cards. • Form name (a to z) • Form name (z to a) • Updated by (a to z) • Updated by (z to a) • Updated (recent to last) - Default option • Updated (last to recent)
Filter forms by setting up filter criteria	**1.** Select the filter icon ( ). **2.** Add your filter condition criteria including any AND/OR logic. **3.** To add additional conditions, select **+ New condition set**. **4.** Select **Apply**.

Form layout (sections)

A view is composed of sections that group the data elements that you want displayed.

You can add multiple sections and change the layout of each section into one or two columns. A section is where you can group the data elements that you want to display for a form view. These elements include form fields, annotations, formatters, and embedded lists.

See Customize your form layout in Table Builder.

Form fields

When you select a table in Table Builder, the available fields are displayed in the form elements panel. You can create fields for the table or configure the fields and their properties.

By dragging these fields into the form editor, you can then visually arrange these fields on the form view you have selected.

See Add fields to a form layout in Table Builder.

See Modify field properties in Table Builder form editor.

Form annotations

A form annotation is an additional piece of information on a form, such as a line or paragraph of text.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For example, you may want to add instructional text for a particular section on the form. By using Table Builder, you can add a form annotation to the view that displays on-screen instructions on how a user should enter information.

See Create form annotations in Table Builder.

Formatters

A **Formatter** is a read-only element in a form that displays additional information about the selected table record. Unlike fields, formatters are not editable.

See Using formatters.

By using Table Builder, you can visually drag any of the displayed formatters onto the form editor and arrange them.

See Add formatters in Table Builder.

Embedded lists

You can use an embedded list to display the data for a selected related list on a form. A related list shows the records in the tables that have relationships to the current record. For example, for a form view that references a Problem table, you might want to display a list of locations that are impacted by a selected problem.

With Table Builder, you can visually drag a list from the **Embedded Lists** tab of the form elements panel onto the form editor, and then arrange them.

By adding an embedded list to a form, you enable your users to view or edit the related list without needing to navigate away from the form. Changes to the data in the list are saved when the user saves the form.

See Add embedded lists in Table Builder.

Choose a form view in Table Builder

Choose existing views, duplicate views, or create entirely new views for form users in Table Builder.

Before you begin

- Launch Table Builder. For more information, see Accessing Table Builder.
- (Optional) Choose a domain to work within (if not global). For more information, see Domain separation and Table Builder.
- (Optional) Choose an application scope to work within (if not global). For more information, see Using an application scope with Table Builder.

Note: To understand how to approach customizing your forms, review Table Builder workflow and navigation.

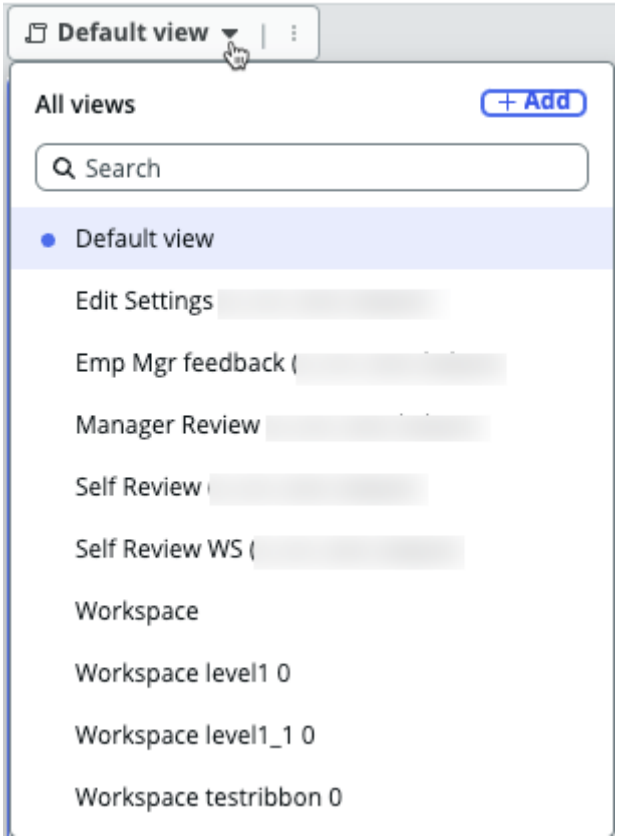
Role required: personalize_form or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

Procedure

1. Navigate to the **Forms** tab in Table Builder.
   The "Default view" is selected.

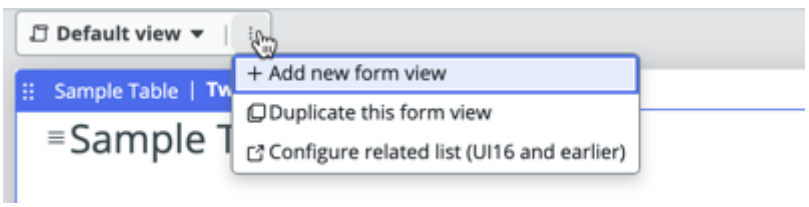
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

View list



2. You can modify the default view selected by using the View list or select another view. You can also add a view or choose one the other form view actions from the Additional actions menu.

Additional actions menu



   - Add a new view to customize by clicking **Add new form view** from the view list or by selecting an option from the Additional actions (⋮) menu.
   - Duplicate an existing view to customize by selecting a view and then choosing **Duplicate this form view** from the Additional actions (⋮ menu.
   - Configure a related list for the selected form view by selecting this option from the Additional actions (⋮) menu.

   Note: You can also choose a view from the Additional actions list to the right of the **Forms** tab or navigate to the **All form views** list to filter and sort a large list of form views. See All form views list.

3. With the view selected, you can modify the form view in the following ways:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Add fields to your form. See Add fields to a form layout in Table Builder.
- Customize the layout of your form (sections). For more information, see Customize your form layout in Table Builder.
- Add non-field elements that display data on your form. For more information, see Add formatters in Table Builder.
- Add instructional text or other text to your form. For more information, see Create form annotations in Table Builder.
- Add related lists to your form. For more information, see Add embedded lists in Table Builder.

Note: To customize the elements that display on a form, see Forms in Table Builder.

Customize your form layout in Table Builder

Visually design the look of the forms your users see by customizing form views in Table Builder.

Before you begin

- Launch Table Builder. For more information, see Accessing Table Builder.
- (Optional) Choose a domain to work within (if not global). For more information, see Domain separation and Table Builder.
- (Optional) Choose an application scope to work within (if not global). For more information, see Using an application scope with Table Builder.

Note: To understand how to approach customizing your forms, review Table Builder workflow and navigation.

Role required: personalize_form or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

Procedure

1. Navigate to the **Forms** tab in Table Builder.
2. Choose a view to work with.
   For detailed information on how to choose a view for a form, see Choose a form view in Table Builder.
3. Customize the form layout that displays in the form editor by performing the following actions.

Option	Action
Add form sections	**a.** To display a list of layout components that you can add to the form, select **Components** in the Add form elements panel. **b.** Select **Section**, and then drag the section to the desired location above or below any existing sections on the form. **c.** Enter a title for the section.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Action
	Note: You can also add a new section from within the form editor by pointing to the **+** icon at the top of an existing sec tion, and then selecting **+Add section** or by using the menu in the section head er.
Configure section layout (columns)	Change the layout of your section. Sections can consist of one column or be split into two columns. **a.** Select a section in the form editor. **b.** Change the column layout via one of the fol lowing methods: ▪ From the Section panel, select **One-col umn** or **Two-column**. ▪ From the section header menu, select **One-column** or **Two-column**.
Merge section with the section above	Merge two sections. Merging a section refers to combining two sections together with a sin gle section title. A merge is useful when you want to have a single section with two differ ent column layouts. **a.** In the form editor, select a section. **b.** Merge the selected section with the section above via one of the following methods: ▪ In the section header, select the Addi tional actions (**|**) menu, and then select **Merge with section above**. ▪ From the section panel, select **Merge with section above**.
Detach section	Detach a section. You can split two previous ly merged sections apart into separate ele ments again. **a.** In the form editor, select a previously merged section. **b.** Detach the selected section via one of the following methods: ▪ In the section header, select the Addition al actions (**|**) menu, and then select **De tach from section above**. ▪ From the section panel, select **Detach from section above**.
Remove section from form	**a.** In the form editor, select a section. **b.** Remove the selected section from the form view via one of the following methods:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Action
	▪ On the top right of the section header, select the Trash icon. ▪ To remove the section from the form view, in the section header, select the Additional actions (⋮) menu, and then select **Delete section**.

4. To rearrange the sections within the form, in the section header, select the section and then drag it to the desired location on the form or use the up and down arrow icons.
5. Select **Save**.

Add fields to a form layout in Table Builder

Visually add and arrange fields within a form layout to create a form that matches your requirements using Table Builder.

Before you begin

- Launch Table Builder. For more information, see Accessing Table Builder.
- (Optional) Choose a domain to work within (if not global). For more information, see Domain separation and Table Builder.
- (Optional) Choose an application scope to work within (if not global). For more information, see Using an application scope with Table Builder.

Note: To understand how to approach customizing your forms, review Table Builder workflow and navigation.

Role required: personalize_form or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

About this task

Procedure

1. Navigate to the **Forms** tab in Table Builder.
2. Choose a view to work with.
   For detailed information on how to choose a view for a form, see Choose a form view in Table Builder.
3. To display a list of the existing fields for the selected table, in the add form elements panel, select **Fields**.

   Note: You can also launch the Add form elements panel by selecting the Add (⊕) icon above any existing element in the form editor.

4. Customize the fields that display on the selected form view by performing one of the following actions.

   Note: Certain field types are only supported in Core UI. Fields are flagged with an exclamation icon (!) to indicate an unsupported type.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Action
Add an existing field to a form	**a.** To select a field, click a field in the list. **Note:** To select multiple fields, select the option that displays on the right of the field. **b.** In the form editor, drag one or more selected fields to the desired location and then move them around as necessary.
Add a new field to a form	**a.** Select **+ Add a field in the table**. **b.** In the **Column Label** field, enter a label. **c.** In the **Column name** field, enter the internal name for the field. **d.** Select the type of field from the **Type** list. To set additional properties for the field that you are creating, click **Advanced settings**. For information on field types and advanced settings, see Field configuration in Table Builder. **e.** To add the new field to the table, click **Add**. A confirmation dialog displays. You can click **Done** to close it or click **Add another one** to add more fields to the table. The new field is displayed at the top of the Fields list and can be dragged onto the form editor.
Remove field from form	**a.** In the form editor, click a field to select it. **b.** To remove the selected field from the form view, click the x icon to the right of the field. **Note:** If you are unable to delete a field, it may be because the table belongs to a different scope.

5. Select **Save**.

Modify field properties in Table Builder form editor

Modify basic field properties for the fields that display for users in a form view from within the **Forms** tab in Table Builder.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

- Launch Table Builder. For more information, see Accessing Table Builder.
- (Optional) Choose a domain to work within (if not global). For more information, see Domain separation and Table Builder.
- (Optional) Choose an application scope to work within (if not global). For more information, see Using an application scope with Table Builder.

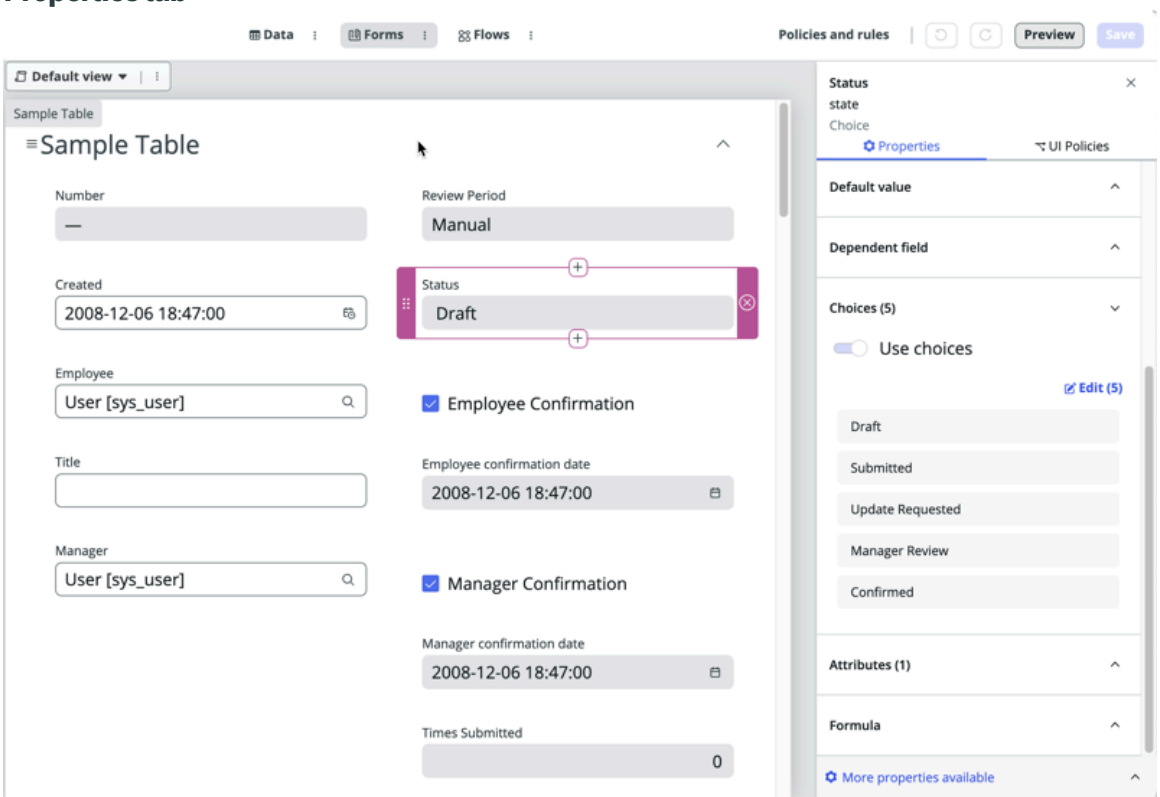
Note: To understand how to approach customizing your forms, review Table Builder workflow and navigation.

Role required: personalize_form or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

Procedure

1. On the **Forms** tab, choose a view to work with.
   For detailed information on how to choose a view for a form, see Choose a form view in Table Builder.
2. In the form editor, select a field to highlight it.
   The following example shows that the panel on the right displays the **Properties** tab. This tab shows the basic properties of the selected field.

Properties tab



3. Edit the basic field properties as needed for the selected field.
   The sections displayed vary depending on the field type. For a description about these fields, see Field configuration in Table Builder.

Note: Below the **Properties** tab, select **More properties available** to launch the full dictionary entry for the selected field.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

4. To view any UI policies that are set up for the selected field and make edits to that field, select the **UI Policies** tab.
   For more information about the policies and properties, see Policies and rules properties in Table Builder.
5. Select **Save**.

Create form annotations in Table Builder

Add instructional text and other design elements to your forms by using form annotations in Table Builder.

Before you begin

- Launch Table Builder. For more information, see Accessing Table Builder.
- (Optional) Choose a domain to work within (if not global). For more information, see Domain separation and Table Builder.
- (Optional) Choose an application scope to work within (if not global). For more information, see Using an application scope with Table Builder.

Note: To understand how to approach customizing your forms, review Table Builder workflow and navigation.

Role required: personalize_form or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

Procedure

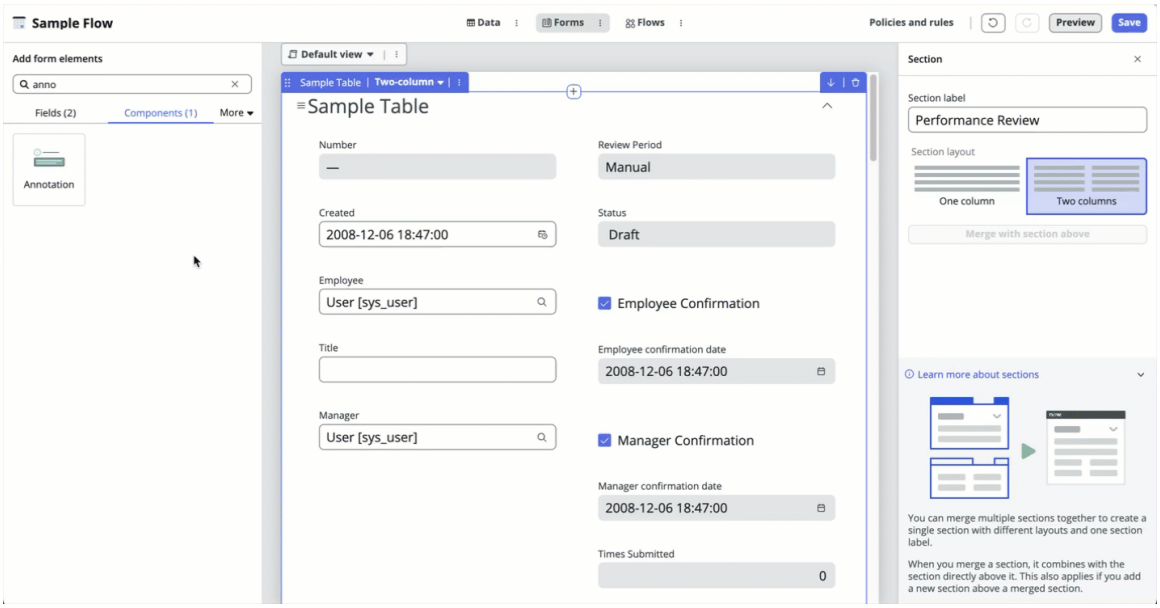
1. Navigate to the **Forms** tab in Table Builder.
2. Choose a view to work with.
   For detailed information on how to choose a view for a form, see Choose a form view in Table Builder.
3. To display a list of the available components that you can add to your form view, in the Add form elements panel, select **Components**.

   Note: You can also launch the Add form elements panel by clicking the Add (+) icon above an existing element in the form editor.

4. Select the **Annotation** elements, and then drag it to the location in the form editor where you'd like to position the annotation text as shown in the following example.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add annotation



5. Select the **Annotation type**.

Annotation type	Description
Info Box Blue	Blue box that surrounds the entered text.
Info Box Red	Red box that surrounds the entered text.
Line Separator	Line separator inside a section.
Plain Text	Text that is entered as plain text.
Section Details	Text that is entered as section details.
Section Plain Text	Text that is entered as plain text in a section.
Section Separator	Line separator between sections.
Text	Text that is entered as text.

6. Select an option for **Annotation text**.

Annotation text option	Description
Plain text	Text that is entered renders on the form as plain text.
Rich text	Rich text. When you point to the text box be low, an Edit rich text icon is displayed. Select this icon to launch a rich text editor window where you can adjust the basic properties for the text that you want to display on the form such as: ○ Font size ○ Font color or background color

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Annotation text option	Description
	○ Bold, italic, underline ○ Indentation ○ List type (bullet or ordered list) ○ Add hyperlink or remove hyperlink To show the HTML code that renders on the form, click **Show HTML**, and then click **Apply**.

7. Enter the desired annotation text in the field that displays in the Annotation panel.
8. Select **Save**.

Add formatters in Table Builder

You can add form elements that display non-field information to users by adding a formatter in Table Builder. For example, you may want to display a **Comments** field where users can add comments to a displayed form.

Before you begin

Note: Formatters are only supported in Core UI and earlier.

- Launch Table Builder. For more information, see Accessing Table Builder.
- (Optional) Choose a domain to work within (if not global). For more information, see Domain separation and Table Builder.
- (Optional) Choose an application scope to work within (if not global). For more information, see Using an application scope with Table Builder.

Note: To understand how to approach customizing your forms, review Table Builder workflow and navigation.

Role required: personalize_form or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

Procedure

1. Navigate to the **Forms** tab in Table Builder.
2. Choose a view to work with.
   For detailed information on how to choose a view for a form, see Choose a form view in Table Builder.
3. To display a list of the available components that you can add to your form view, in the Add form elements panel, select **Formatters**.

   Note: You can also launch the Add form elements panel by clicking the Add (+) icon above an existing element in the form editor.

4. Select a formatter, and then drag it to the location in the form editor where you'd like to place it.

   Note: Formatters in forms are only supported in Core UI and earlier interfaces.

   For information on how you can add a formatter to a form, see Formatters.
5. Select **Save**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add embedded lists in Table Builder

Add an embedded list using Table Builder to display data for a selected related list on your form. A related list shows the records in the tables that have relationships to the current record. For example, for a form view that references a Problem table, you might want to display a list of locations that are impacted by a selected problem.

Before you begin

Note: Embedded lists are only supported in Core UI and earlier.

- Launch Table Builder. For more information, see Accessing Table Builder.
- (Optional) Choose a domain to work within (if not global). For more information, see Domain separation and Table Builder.
- (Optional) Choose an application scope to work within (if not global). For more information, see Using an application scope with Table Builder.

Note: To understand how to approach customizing your forms, review Table Builder workflow and navigation.

Role required: personalize_form or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

Procedure

1. Navigate to the **Forms** tab in Table Builder.
2. Choose a view to work with.
   For detailed information on how to choose a view for a form, see Choose a form view in Table Builder.
3. To display a list of the embedded lists that you can add to your form view, in the Add form elements panel, select **Embedded lists**.

   Note: You can also launch the Add form elements panel by clicking the Add (+) icon above an existing element in the form editor.

4. Select an embedded list, and then drag it to the location in the form editor where you'd like to place it.

   Note: Embedded lists in forms are only supported in Core UI and earlier interfaces.

   For information on how you can add an embedded list to a form, see Embed a list within a form.
5. Select **Save**.

Preview your form

Previewing and saving your form in Table Builder.

Before you begin

- Launch Table Builder. For more information, see Accessing Table Builder.
- (Optional) Choose a domain to work within (if not global). For more information, see Domain separation and Table Builder.
- (Optional) Choose an application scope to work within (if not global). For more information, see Using an application scope with Table Builder.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: To understand how to approach customizing your forms, review Table Builder workflow and navigation.

Role required: personalize_form or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

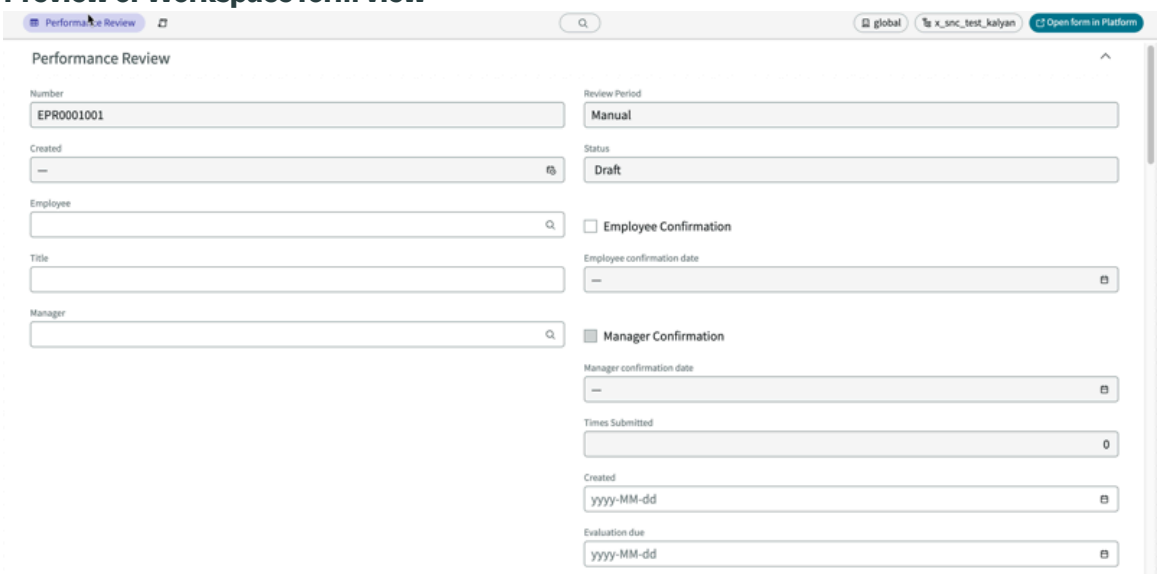
Procedure

1. Access Table Builder.
   - For information on viewing forms, see Launching Table Builder from UI Builder.
2. Select the **Forms** tab.
3. Select a form view.
4. Customize the form view as desired.
5. Save your changes by selecting **Save**.

   Note: Once you save your changes, you cannot undo them using the Undo function. The Undo icon ( ) is grayed out.

6. Select **Preview** to review your saved changes.
   The Preview window displays a record for your form. The Workspace form view is shown in the example.

Preview of Workspace form view



7. You can perform any of the following actions as you review your changes to better understand how your form will display for users.
   - When previewing a form, you can search for a new record to populate your form with by clicking the New record icon ( ). This will also let you add a new record if none exist so you can really see what the form will look like.
   - When previewing a form, you can open the form in the ServiceNow AI Platform by clicking the Open form in Platform icon (Open form in Platform).
8. When you are finished reviewing, close the browser window.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Policies and rules in Table Builder

Policies and rules define how information appears to users based on role and entries, as well as who can access tables and forms built in Table Builder.

Sections for working with policies and rules

The **Policies and rules** tab in Table Builder has several sections, but you can edit only UI policies within Table Builder. The other sections are read-only and contain links to manage their content in other parts of the ServiceNow AI Platform.

Administrative UI policy and rules sections

Section	Description	Editable within Table Builder
UI Policies	Dynamically change the behavior of information on a form and control custom process flows for tasks.	Yes.
Access control rules	Control access to the specified resource based on role, condition, and script criteria being met.	No.
Client Scripts	Create scripts that control how form fields appear based on defined criteria.	No.
Business Rules	Create business rules to accomplish tasks like automatically changing values in form fields when certain conditions are met.	No.
Workspace view rules	Define rules to control how users view workspaces based on criteria.	No.

To manage a rule or policy script outside of Table Builder, select for the entry. Or, you can view the full list of rules and scripts outside of Table Builder by selecting the **Full list** link for each section of the **Policies and rules** tab:

- **Full ACL list**
- **Full client scripts list**
- **Full business rule list**
- **Full workspace view rules list**

UI policies for tables and forms

Configure UI policies to adjust how forms appear based on roles and user input.

For example, you can use UI policies to make a field on a form read-only or required or hide certain fields depending on a user's role. Basic UI policies do not require any scripting. For more advanced actions, use the **Run scripts** option.

You can also use client scripts to perform all these actions, but for faster load times use UI policies when possible.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Create a UI policy in Table Builder

Configure UI policies in Table Builder to adjust how forms appear based on roles and user input.

Before you begin

- Launch Table Builder. For more information, see Accessing Table Builder.
- (Optional) Choose a domain to work within (if not global). For more information, see Domain separation and Table Builder.
- (Optional) Choose an application scope to work within (if not global). For more information, see Using an application scope with Table Builder.

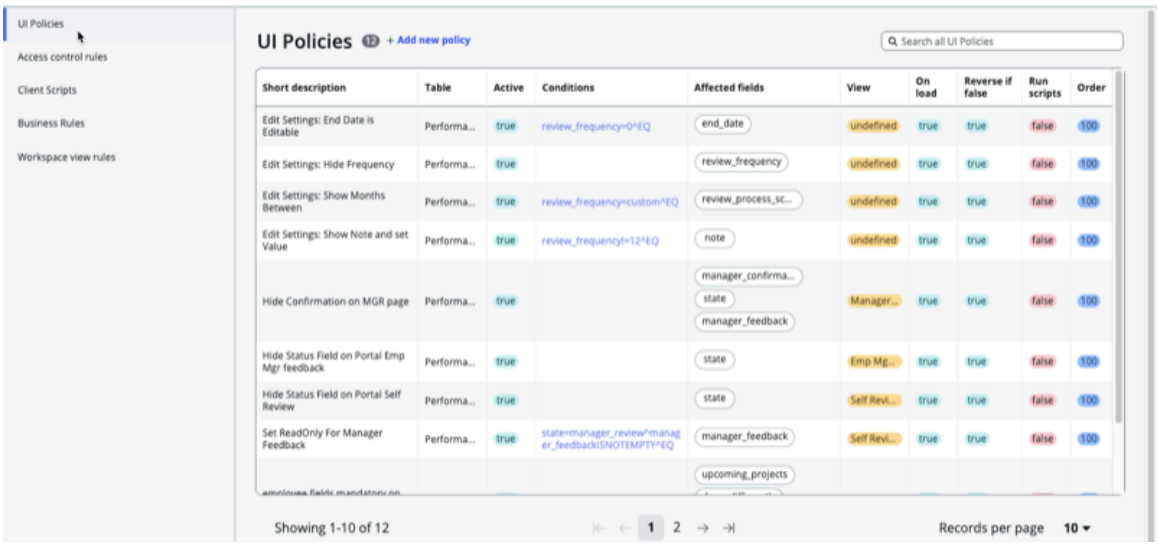
Role required: personalize_rules or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

About this task

You can create and edit UI policies only for applications that you own. For more information on application scope, see Domain separation and Table Builder.

Procedure

1. On the Table Builder screen, choose a table to work with.
2. Select the **Policies and rules** tab.
   Alternatively, you can add a UI policy for a field directly in the **Forms** tab of Table Builder in the right pane that appears when you select a field.
3. Select the UI Policies section to display a list of UI policies.
4. Select the **Add new policy** link to display the policy details.
   - If the link is inactive, you must change the scope that you're working with on the **Forms** tab.
   - If a field has policy and appears in the view, the active icon appears on the top corner of the field as shown in the following example.



5. On the form, fill in the fields for the Policy details section.
   For more information on field definitions, see Policies and rules properties in Table Builder.
6. In the When these conditions are met section, specify the conditions that, if fulfilled, cause the UI policy to be applied using the condition builder.
   To add multiple sets of conditions, select **New condition set**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

7. In the Do the following section, create field actions that should be performed when the conditions are met.
8. To finish adding the policy, select **Add UI policy**.

Result

After you add the UI policy, you can configure scripts and the related list actions for it. For more information on client scripts and UI policies, see Client scripts for UI policies.

Modify a UI policy in Table Builder

Edit UI policies to change how forms appear based on roles and user input in Table Builder.

Before you begin

- Launch Table Builder. For more information, see Accessing Table Builder.
- (Optional) Choose a domain to work within (if not global). For more information, see Domain separation and Table Builder.
- (Optional) Choose an application scope to work within (if not global). For more information, see Using an application scope with Table Builder.

Role required: personalize_rules or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

About this task

You can create and edit UI policies only for applications that you own. For more information on application scope, see Domain separation and Table Builder.

Procedure

1. In the UI policies section, on the **Policies and rules** tab, select the policy that you want to edit. You can also edit a field's UI policy directly in the form by selecting it in the **UI policies** tab of the right pane that appears when you select a field.
2. Change the UI policy fields and attributes as needed.
   For more information on field definitions, see Policies and rules properties in Table Builder.
3. Click **Update UI policy**.

Flows in Table Builder

You can use the Workflow Studio functionality that is integrated within Table Builder to manage your table's record-based flows. You can manage these flows by using the **Flows** tab.

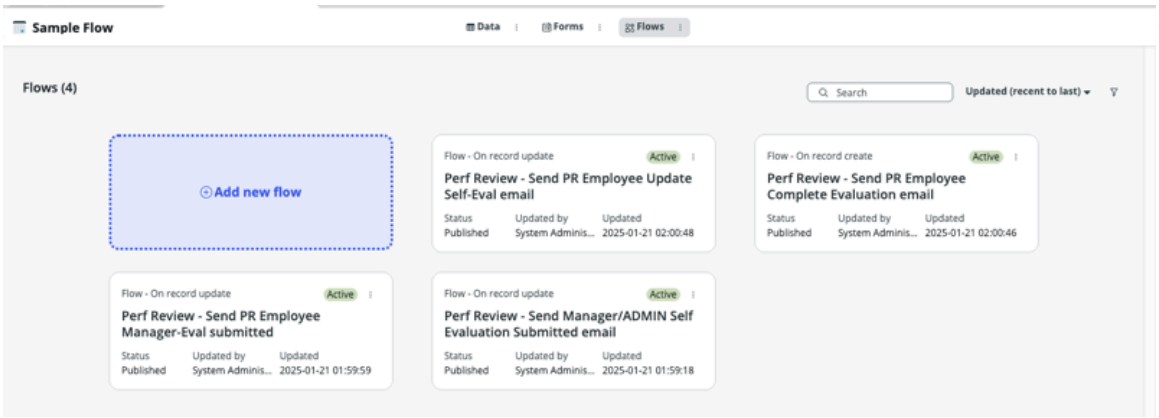
Workflow Studio is a ServiceNow AI Platform® feature that enables process owners to automate work. Build multi-step flows from reusable components without having to code.

The **Flows** tab within Table Builder is where you can view and work with record-based flows for a selected data table (that is, actions that are triggered when a table record is created or updated or both).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Flows tab

The following diagram shows flow cards in Table Builder.



Find a record-based flow

Use the **Flows** tab in Table Builder to search and filter a list of flows that are triggered when selected table records are updated, created, or both.

Before you begin

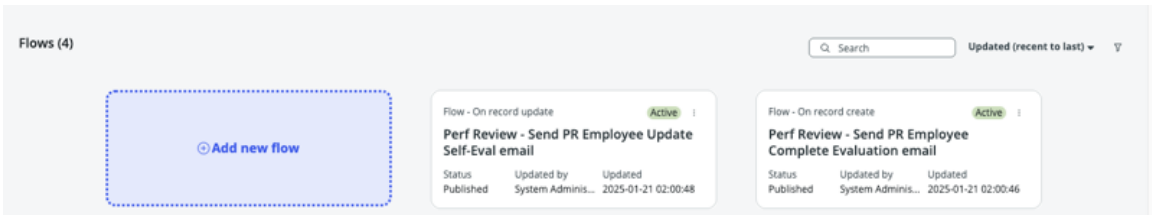
- Launch Table Builder. For more information, see Accessing Table Builder.
- (Optional) Choose a domain to work within (if not global). For more information, see Domain separation and Table Builder.
- (Optional) Choose an application scope to work within (if not global). For more information, see Using an application scope with Table Builder.

Role required: personalize_forms, personalize_dictionary, and flow_designer permissions (or relevant AES user role and delegated developer permissions). For more information, see Delegate developers using AES.

Procedure

1. Launch Table Builder as described in Accessing Table Builder.
2. Select the **Flows** tab.
   The flows triggered by the selected table are listed as flow cards.

Flow cards list



3. Filter or search the list of flow cards as needed to find the desired flow.

Find the desired flow card

Option	Steps
Search text	Type a keyword in the **Search** box. The list of cards will be filtered as you type.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Steps
Sort flows	Select an option from the sorting list to sort the flow cards. **a.** Flow name (a to z) **b.** Flow name (z to a) **c.** Updated by (a to z) **d.** Updated by (z to a) **e.** Updated (recent to last) - Default option **f.** Updated (last to recent)
Filter flows by setting up filter criteria	**a.** Select the filter icon ( ). **b.** Add your filter condition criteria including any AND/OR logic. **c.** To add additional conditions, select **+ New condition set**. **d.** Select **Apply**.

What to do next

- Select the flow card to edit your new flow using Workflow Studio. See Modify a record-based flow.
- Create a new record-based flow from an existing one. See Copy a record-based flow.
- Delete the flow. See Delete a record-based flow.

Add a record-based flow

Add flows that are triggered when a table record is updated, created, or both updated and created by using the **Flows** tab in Table Builder.

Before you begin

- Launch Table Builder. For more information, see Accessing Table Builder.
- (Optional) Choose a domain to work within (if not global). For more information, see Domain separation and Table Builder.
- (Optional) Choose an application scope to work within (if not global). For more information, see Using an application scope with Table Builder.

Role required: personalize_forms, personalize_dictionary, and flow_designer permissions (or relevant AES user role and delegated developer permissions). For more information, see Delegate developers using AES.

Procedure

1. Launch Table Builder as described in Accessing Table Builder.
2. Select the **Flows** tab.
3. Select the **Add new flow** button.
4. On the New Flow form, fill in the following fields for your new flow.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

New Flow form

Field	Description
Name	Name to uniquely identify your flow. The system computes the internal name of the flow from the name.
Description	Description of your flow.

5. **Optional:** To enter advanced options for your flow, select **Show advanced options** on the New Flow form.

Advanced options

Field	Description
Protection	Option that you can select if the flow is read-only. You can only select a value when you create the flow in an application scope that you own.
Run as	(Optional) Option to specify the user that runs the flow. You can select the system user or the user who initiates the session. Select the user who initiates the session option when updates should come from the user who triggered the flow. For example, use this option when you want the incident record comments to come from the user who started the flow. Settings for the Run as option in a flow don't apply to child subflows. To create a flow that can run with a personal OAuth token, select the user who initiates the session option. If the user who is running the flow has a personal OAuth token, the flow runs with that token. For more information about creating a personal OAuth token, see OAuth 2.0 credentials. When flows run as the user who initiates the session, the system limits the flow actions by user access control list (ACL) restrictions. Ensure that security restrictions don't prevent users who trigger the flow from performing flow actions. Flows that are run by the initiating user also respect user-specific settings, such as the date/time formats. **Note:** Inbound email flows ignore this setting and always run as the user who initiates the session. To test access controls for an inbound email flow, impersonate a typical inbound email user and manually trigger the flow.
Run with roles	Roles that the flow runs with. This option is only available when **Run as** is set to **user who initiates the session**.

6. Select **Continue**.
7. Choose the event that your flow will be triggered by.

Option	Description
When the record is created	Create a flow that is triggered when a record in the selected table is created.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Description
When the record is updated	Create a flow that is triggered when a record in the selected table is updated.
When the record is created or updated	Create a flow that is triggered when a record in the selected table is created or updated.

The flow that you just created is displayed in the flow diagramming view. See Flow diagramming view.

8. Select **Add node**, and then select one of the following options to modify your flow:

Option	Description
Action	Select the desired action. Workflow Studio includes Workflow Studio actions that are available to flows and subflows. Alternatively, a user with the action_designer role can create additional actions to add to flows. The Integration Hub and Spokes plugins install additional actions. To add draft actions from the More Actions menu, set **Show draft actions** to **true**. To view spokes that are available in the ServiceNow Store, set **Show store spokes** to **true** from the More Actions menu. **Note:** Under **Not Installed Spokes**, the system displays spokes that are available in the ServiceNow Store based on compatibility with the ServiceNow version and application dependency on Workflow Studio.
Flow Logic	Select an option to specify conditional or repeated operations.
Subflow	Select a published subflow and define the input values. In addition to adding a subflow as a flow action, you can enable the **Show triggered flows** option from the More Actions menu to select an activated flow and define the required inputs. Running a triggered flow ignores its trigger conditions and runs all actions.

Note: Alternately, you may toggle the Workflow Studio default view from the view selector, and then select **Add an Action, Flow Logic, or Subflow** in that view of Workflow Studio. For more information, see Flow diagramming view.

9. Continue modifying your flow in Workflow Studio.

Note: For detailed information on how to work with flows for your application, see Create a flow.

10. Select **Save**.
Workflow Studio saves a draft of the flow, trigger, and actions.

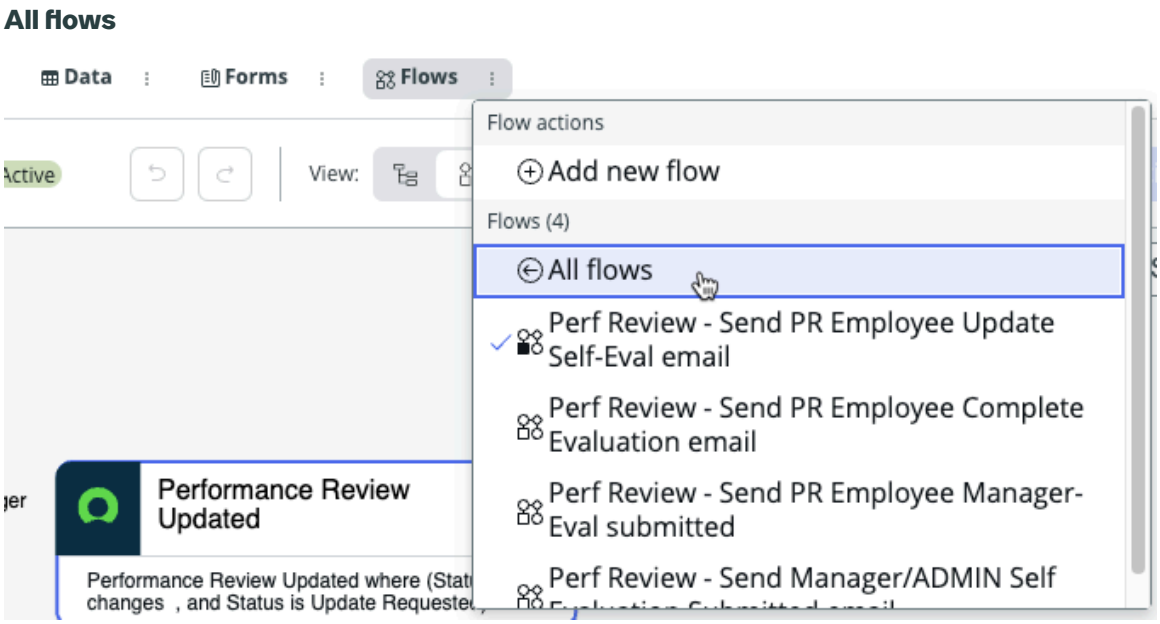
What to do next

Test your flow until you're ready to activate it. For more information on testing and editing flows, see Activate a flow.

Note: Your application can trigger only activated flows. For detailed information on how to design, test, and activate flows for your application, see Flows in Workflow Studio.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

To return to the comprehensive list of flows related to the selected table after editing a flow, select the Additional actions (⋮) menu to the right of the **Flows** tab, and then select **All flows** as shown in the following example.



Modify a record-based flow

Edit flows that are triggered when table records that you select are updated, created, or both updated and created by using the **Flows** tab in Table Builder.

Before you begin

- Launch Table Builder. For more information, see Accessing Table Builder.
- (Optional) Choose a domain to work within (if not global). For more information, see Domain separation and Table Builder.
- (Optional) Choose an application scope to work within (if not global). For more information, see Using an application scope with Table Builder.

Role required: personalize_forms, personalize_dictionary, and flow_designer permissions (or relevant AES user role and delegated developer permissions). For more information, see Delegate developers using AES.

About this task

The **Flows** tab in Table Builder provides a list of flows that are triggered by record changes to a table and enables you to manage them from the record table that triggers the workflow.

Procedure

1. Launch Table Builder as described in Accessing Table Builder.
2. Select the **Flows** tab.
   The available flows that are triggered by the table are listed as flow cards.
3. Filter or search the list of flow cards as needed to find the desired flow.
   See Find a record-based flow.
4. Select a flow card to open the flow for editing.
5. Edit the record-based flow by using Workflow Studio.
   For detailed information on how to work with flows for your application, see Create a flow .

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: To return to the list of flow cards that are related to the table, select the Additional actions (⋮) menu to the right of the **Flows** tab, and then select **All flows**.

6. Select **Save**.

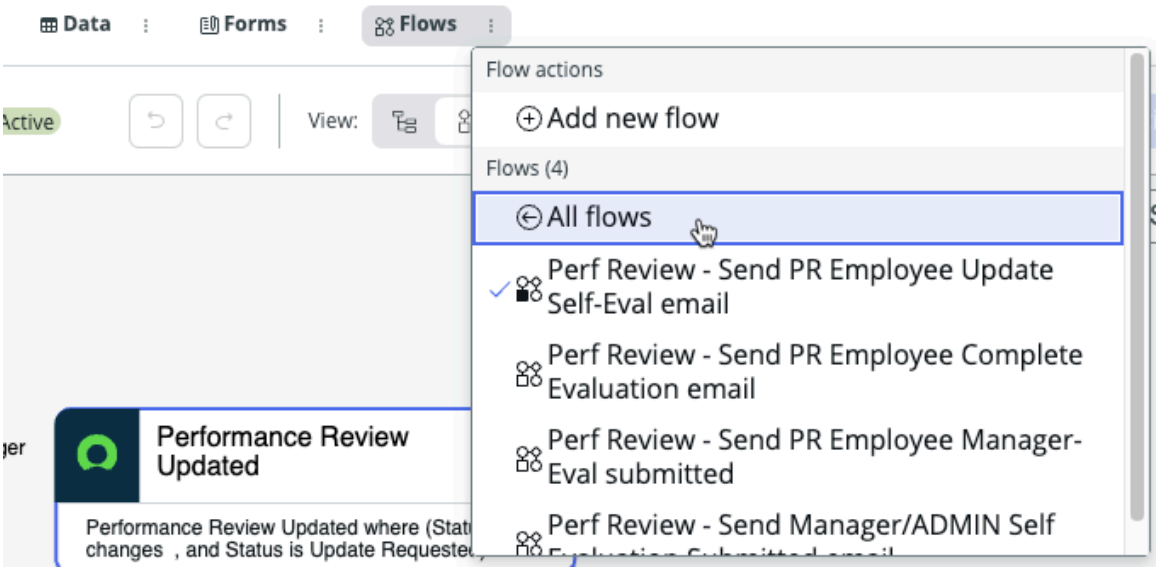
What to do next

Test your flow until you're ready to activate it. For more information on testing and editing flows, see Activate a flow.

Note: Your application can trigger only activated flows. For detailed information on how to design, test, and activate flows for your application, see Flows in Workflow Studio.

To return to the list of flows related to the table that you selected after you edited a flow, select the Additional actions (⋮) menu to the right of the **Flows** tab, and then select **All flows** as shown in the following example.

All flows



Delete a record-based flow

Delete a record-based flow by using the **Flows** tab in Table Builder.

Before you begin

- Launch Table Builder. For more information, see Accessing Table Builder.
- (Optional) Choose a domain to work within (if not global). For more information, see Domain separation and Table Builder.
- (Optional) Choose an application scope to work within (if not global). For more information, see Using an application scope with Table Builder.

Role required: personalize_forms, personalize_dictionary, and flow_designer permissions (or relevant AES user role and delegated developer permissions). For more information, see Delegate developers using AES.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Launch Table Builder as described in Accessing Table Builder.
2. Select the **Flows** tab.
   The available flows that are triggered by the table are listed as flow cards.
3. Filter or search the list of flow cards as needed to find the desired flow.
   See Find a record-based flow.
4. Select the Additional actions (⋮) menu in the top-right corner of the flow that you want to delete, and then select **Delete** from the list of options.
5. In the delete confirmation window, enter DELETE, and then select **Delete**.
6. Select **Save**.

Copy a record-based flow

Create a new record-based flow from an existing flow by using the **Flows** tab in Table Builder.

Before you begin

- Launch Table Builder. For more information, see Accessing Table Builder.
- (Optional) Choose a domain to work within (if not global). For more information, see Domain separation and Table Builder.
- (Optional) Choose an application scope to work within (if not global). For more information, see Using an application scope with Table Builder.

Role required: personalize_forms, personalize_dictionary, and flow_designer permissions (or relevant AES user role and delegated developer permissions). For more information, see Delegate developers using AES.

Procedure

1. Launch Table Builder as described in Accessing Table Builder.
2. Select the **Flows** tab.
   The available flows that are triggered by the table are listed as flow cards.
3. Filter or search the list of flow cards as needed to find the desired flow.
   See Find a record-based flow.
4. Select the Additional actions (⋮) menu in the top-right corner of the flow that you want to delete, and then select **Copy** from the list of options.
5. In the Create a copy of the flow window that appears, enter a name for the new flow that you want to create from the existing flow, and then select **Copy**.
6. Edit your new flow by using Workflow Studio.
   For detailed information on how to work with flows for your application, see Create a flow .
7. Select **Save**.

> **Note:** To return to the list of flows after editing a flow, select the Additional actions (⋮) menu to the right of the **Flows** tab, and then select **All flows**.

Table Builder reference

Reference topics provide additional information about the form and table elements you can configure using Table Builder.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Table properties in Table Builder

Various table and table column properties can be modified by using Table Builder.

Table column properties

The following table shows descriptions of the properties that you can modify for the columns in a data table.

Table column properties

Field	Description
Column label	Unique label for the column.
Column name	Database name for the column.
Type	Type of information that the column contains. For example, to contain plain text in the column, select **String**. Depending on the type that you select, fill in the additional fields to further define the table column. For example, if you select **String**, define the character limit of the string input. Or, if you select **Choice**, define the choices that users can choose from. See Field types.
Reference	Table that is associated with the column. This field applies only if the column type is **Reference**.
Max length	Maximum number of characters that users can enter in the field.
Default value	Value that populates the field automatically after a new record is created.
Display	Option to set the column as the display value for the table. A reference field shows the display value of the table to which it is referring. For example, the **Opened by** column of the task table refers to the user table. Because the display value of the user table is the user name, the **Opened by** field shows something like **Beth Anglin** or **Joe Employee**. When you select a display value, choose the table column that would act as an appropriate title for individual records. Only one column can act as the display value for a table.

Table properties

The following table shows the descriptions of the properties that you can modify for a selected data table.

Table properties

Option	Procedure
Change the table label	In the General information tab, update the **Table label** field.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Table properties (continued)

Option	Procedure
Make the table extensible	1. In the General information tab, select **Advanced**. 2. Select the **Make extensible** check box.
Add record numbers	1. In the General information tab, select **Advanced**. 2. Select the **Add record number** check box. 3. Define the record numbers by updating the **Prefix**, **Starting number**, and **Number of digits** fields.
Accessible from	1. Open the **Access** tab. 2. In the **Accessible From** field, select **All Application Scopes** or **This Application Scope Only**.
Application access controls	1. Open the **Access** tab. 2. In the **Application Access Controls** field, select Read, Create, Update, and/or Delete selection boxes to specify the desired level of access for the table.

Field configuration in Table Builder

You can configure the basic field properties for any field within Table Builder.

Field details

The Field details section lets you configure the basic field elements for the selected field in the form editor.

To see field type descriptions and learn more about what field types you can use, see Field types.

To understand the basic field properties and learn more about adding and customizing a field in a table, see Add and customize a field in a table.

The following table shows field descriptions for the Field details section.

Field details

Field	Description
Label	Name of the field that is displayed for form users.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field details (continued)

Field	Description
Active	Option that you can select to make a field active so that form users can view it. Deselecting **Active** inactivates a field. When **Active** is not selected, the field will not display in the form editor. To reactivate a field, hover to the left of the field column in the **Data** tab and select the Open side panel icon [⚙]. Select **Active** and save changes.
(**Deprecated**). Read only	Option that you can select so that the field is not editable by the form users.
Read only option	Read-only behavior is controlled by the new **read_only_option** choice field, which lets you choose between **Display Read Only** or **Strict Read Only**.
Mandatory	Option that you can select so that the data that your users enter in the field is required before they can save their changes to the form.
Configure Label details menu (...)	Properties for a field label: • **Tooltip** property that is displayed when a user points to the field. • **Plural** property that indicates if multiple instances of a field exist, your user should enter the plural form of the label (for example, the plural of cat is cats) • **URL** property that indicates if the field label links to some other location, your user should enter the URL here.

Default value

The default value automatically fills in the field on the empty form for a new record, or fills in the field later (if the field is empty) when your user submits the new record. You can specify the default values as either a constant or use a script to generate them.

To learn how to define a default value for the fields, see Specify a default field value.

The following table shows field descriptions for the Default value section.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Default value

Field	Description
Use dynamic default	Property that displays the **Dynamic default value** field where you can set the default value that is displayed for the field.
Default value	String or number that represents the default value of the field.
Dynamic default value	Keywords that can be entered in the search field. Your user can then select a field that contains the desired default value for the field.

Function definition

A function field generates a value that is based on the simple computations of other fields and constants.

This section is displayed in the editor when you are creating a field and you select **Function field** in the Advanced properties section.

For information on defining function fields, see Function field.

The following table shows field descriptions for the Function definition section.

Function definition

Field	Description
Function Definition	Function expression that generates the desired value for the field.

Reference

A reference qualifier restricts the records that are available for reference fields.

For information on defining reference fields, see Reference field type.

The following table shows field descriptions for the Reference section.

Reference fields

Field	Description
Reference	Data table that the selected field references.
Use reference qualifier	Qualifier options: • **Simple** option that enables your users to build a simple conditional expression. Your user can select the **Edit Conditions** button below the **Reference qualifier** field. • **Advanced** option that enables your user to build an advanced conditional expression by using the **Reference qualifier** field. The advanced reference qualifiers enable

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Reference fields (continued)

Field	Description
	your user to define an inline encrypted query string or JavaScript (actual code or the name of an existing script include or business rule) filter in the **Reference qualifier** field. • **Dynamic** option that enables your users to build a conditional expression for the selected reference field by using the **Reference qualifier** field. Dynamic reference qualifiers enable your users to use a dynamic filter to run a query against a reference field to filter the returned data set.
Reference qualifier	Referenced qualifier that is based on your user's selection for the **Use reference qualifier** field.
Edit conditions	Hyperlink to launch a pop-up window where your user can enter a simple conditional expression for the **Reference qualifier** field.

Dependent field

A choice or reference field can be declared dependent on another field on the same table. Dependent fields limit their available values based on the value in the dependent field.

Using dependent fields turns on choices. The available choices for your field rely on the value that a user chooses for the dependent field.

For information on defining field dependencies, see Make a field dependent.

The following table shows field descriptions for the Dependent fields section.

Dependent fields

Field	Description
Use dependent field	Field dependency that you can activate for the selected field.
Select dependent field	List where you select a different field on the form that can then be used to control which choices that are displayed to users in the form field you are configuring.

Choices

A choice list is a type of field that lets the user select from a pre-defined set of choices. You can define the available choices and customize the behavior and appearance of your choice lists. For information about the choice list configuration, see Choice list field type.

In the Choices section of the Config panel, select **Use choices**, and then select **Edit** or the **+Add** icon to customize the available choices for the selected field.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The following table shows field descriptions in the Choices section.

Choice properties

Field	Description
Label	Text that is displayed to your users for the choice in the list.
Value	Value for the choice field.
Dependent value	Value that you map in the selected Dependent field to the entered choice.
Domain	Domain that the choice resides in.
Active	Option to display the choice as a selection. Inactive choices do not display for users.
Options	Option that enables you to select **Use the choices from another field** to duplicate the displayed choices for this field from the choices in another selected field.
Show choices as	Options that control how a list of choices displays for users of your form. • **Dropdown** option. You must set a default value for the field. **Note:** If no default is set, then **None** automatically becomes the default value and is shown in the choice list to users. • **Dropdown with --None-- choice** option. **None** becomes the default value. • **Suggestions (Core UI only)** option. Choices are displayed as suggestions.
Filter	Option to filter the choice list. Select **Show inactive choices** to view all choices (including inactive ones).

Attributes

A dictionary attribute alters the behavior of the table or element that the dictionary record describes. You can add or modify the dictionary attributes. For information about dictionary attributes, see Dictionary attributes.

In the Attributes section of the Config panel, select **Edit** or **+Add** to customize the available attributes for the selected field.

The following table shows field descriptions for the Attributes section.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Attribute properties

Field	Description
Attribute	Option list of available attributes. Select an option from the list of available attributes. To review the list of available attributes, see Dictionary attributes.
Value	Attribute value that you can set to alter the field behavior that is described by the dictionary record.

Formula

A formula allows you to calculate the value of a column without writing a script. You can use one of the predefined formulas or combine two are more formulas to calculate the column value. For information on the predefined formulas and examples, see Formulas.

In the Formula section, select **Edit** or **+ Add** to edit or add a formula.

Formula properties

Field	Description
Formula editor	A text-editor like interface that allows you to enter and edit formulas.

Formulas for column values in Table Builder

You can use a predefined function and create a formula to calculate the value of a column without writing a script. Use a predefined function or create a nested formula by using the existing predefined functions to calculate the column value type.

Supported operators

The following comparison operators are supported only for number type values.

- = (Equal to)
- <> (Not equal to)
- > (Greater than)
- < (Less than)
- >= (Greater than or equal to)
- <= (Less than or equal to)

Simple math functions

Use simple math functions to perform basic mathematical calculations on numeric value columns.

AVERAGE

Returns the average value of the arguments.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Syntax	Input	Output
AVERAGE(argument 1, argument 2, … argument n)	Numeric value, function call, or variable	Numeric value

Examples:

- Function: AVERAGE(1,2,3)

  The result is 2.

- Formula: AVERAGE(LENGTH(first_name), LENGTH(last_name))

  The result is the average value of the number of characters in the first_name column and last_name column.

DIVIDE

Returns the final quotient value after consecutively dividing the first argument with the next argument until the function reaches the last argument.

Syntax	Input	Output
DIVIDE(argument 1, argument 2 … argument n)	Numeric value, function call, or variable	Numeric value

Examples:

- Function: DIVIDE(10,20, 0.25, 10)

  The result is 0.2.

- Formula: DIVIDE(LENGTH(full_name),2)

  The result is the number of characters in the full_name column divided by 2.

INDEXMATCH

Retrieves the first not null value from the specified set of arguments.

Syntax	Input	Output
INDEXMATCH(argument 1, argument 2 , … argument n)	String, numeric value, function call, or variable.	Numeric value

Example:

Function: INDEXMATCH("","" ",2,"string")

The result is 2.

MAX

Returns the highest value in the specified arguments.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Syntax	Input	Output
MAX(argument 1, argument 2, ... argument n)	Numeric value, function call, or variable	Numeric value

Examples:

- Function: MAX(1, -5, 20, 6)

  The result is 20.

- Formula: MAX(LENGTH(first_name), LENGTH(last_name))

  The result is the number of characters in the first_name column or last_name column whichever is the highest.

MIN

Returns the lowest value in the specified arguments.

Syntax	Input	Output
MIN(argument 1, argument 2, ... argument n)	Numeric value, function call, or variable	Numeric value

Examples:

- Function: MIN(1, -5, 20, 6)

  The result is -5.

- Formula: MIN(LENGTH(first_name), LENGTH(last_name))

  The result is the number of characters in the first_name column or last_name column whichever is the lowest.

MULTIPLY

Returns the total multiplied value of the arguments.

Syntax	Input	Output
MULTIPLY (argument 1, argument 2, ... argument n)	Numeric value, function call, or variable	Numeric value

Examples:

- Function: MULTIPLY(12, 4)

  The result is 48.

- Formula: MULTIPLY(order, 2)

  The result is the order column value multiplied by 2.

POWER

Returns the result of the base value raised to the power of the exponent value.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Syntax	Input	Output
POWER(argument 1, argument 2)	argument 1 is base and argument 2 is exponent. • base: number or variable • exponent: number or variable	Number

Examples:

- Function: POWER(3,2)

  The result is 9.

- Formula: POWER(LENGTH(full_name),2)

  The result is the number of characters in the full_name column to the power of 2.

SUBTRACT

Returns the result value after consecutively subtracting the next available argument from the earlier argument until the function reaches last argument.

Syntax	Input	Output
SUBTRACT(argument 1, argument 2 ... argument n)	Numeric value, function call, or variable	Numeric value

Examples:

- Function: SUBTRACT(1.15, 0.02, 0.45, -0.85)

  The result is 1.53.

- Formula: SUBTRACT(LENGTH(full_name), LENGTH(first_name))

  The result is the number of characters from the full_name column minus the number of characters from the first_name column.

SUM

Returns the sum of all the arguments.

Syntax	Input	Output
SUM(argument 1,argument 2, ... argument n)	Numeric value, function call, or variable	Numeric value

Examples:

- Function: SUM(0.03, -0.02, 1)

  The result is 1.01.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Formula: SUM(LENGTH(first_name), LENGTH(last_name))

  The result is the total number of characters in the first_name column plus the total number of characters in the last_name column.

COUNTIF

Returns the number of arguments that match the specified criteria within the specified set of arguments.

Syntax	Input	Output
COUNTIF(argument 1, argument 2, argument n-1, criteria)	• argument 1 … argument n: String, numeric value, function call, or variable. • criteria: Criteria that evaluates the specified set of arguments. String, numeric value, function call, or variable.	Numeric value

Example:

Function: COUNTIF(2,3,2,"string",2)

The result is 2.

MODE

Returns the most frequently repeating value in the specified set of arguments.

Syntax	Input	Output
MODE(argument 1,argument 2, ... argument n)	Numeric value, function call, or variable	Numeric value

Example:

Function: MODE(1, 2, 2, 3, 3, 3)

The result is 3.

String functions

Use string functions to reformat or perform calculations on string column values.

CONCATENATE

Joins one or more input strings into a single string.

Syntax	Input	Output
CONCATENATE(string 1, string 2, … string n)	String, function call, or variable	String

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Examples:

- Function: CONCATENATE(first_name, ".", last_name, "@", LOWERCASE(example), ".com")

  The result is the concatenated value <first_name_value>.<last_name_value>@example.com. In this example, <first_name_value> and <last_name_value> are placeholders.

- Function: CONCATENATE(first_name, " ", last_name)

  The result is the concatenated string of first_name column value and last_name column value separated by a white space.

ISBLANK

Finds white spaces or blank values in the string and returns true if there are any.

Syntax	Input	Output
ISBLANK(argument)	String or value	True or false

Examples:

- Function: ISBLANK(“example_string”)

  The result is false.

- Function: ISBLANK(full_name)

  The result is true only when there are empty spaces in the full_name column. Otherwise, the result is false.

LENGTH

Returns the total number of characters in the input string.

Syntax	Input	Output
LENGTH(argument)	String value, function call, or variable	Numeric value

Examples:

- Function: LENGTH("example_string")

  The result is 14.

- Function: LENGTH(full_name)

  The result is the total number of characters in the full_name column value.

LOWERCASE

Converts the input string to all lowercase characters.

Syntax	Input	Output
LOWERCASE(argument)	String, function call, or variable	String in lowercase

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Examples:

- Function: LOWERCASE(“ExamPle inpuT stRing”)

  The result is example input string.

- Function: LOWERCASE(sys_created_by)

  The result is the lower case string of the sys_created_by column value.

REPLACE

Replaces the characters in the source string with the characters in the target string.

Syntax	Input	Output
REPLACE(source_string, target_string, replacement_string)	• source_string: String, function call, or variable • target_string: String, function call, or variable • replacement_string: String, function call, or variable	String

Examples:

- Function: REPLACE(“Pepperoni Pizza”, “Pepperoni”, “Cheese”)

  The result string is Cheese Pizza.

- Function: REPLACE("abe.tuter@example.com", "example", company_name)

  The result string is abe.tuter@<company_name>.com. In this example, <company_name>is a place holder.

TITLECASE

Converts the input string to all title case characters.

Syntax	Input	Output
TITLECASE(argument)	String, function call, or variable	String in title case

Examples:

- Function: TITLECASE("example string")

  The result is Example String.

- Function: TITLECASE(full_name)

  The result is the full name column value in the title case.

UPPERCASE

Converts the input string to all uppercase characters.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Syntax	Input	Output
UPPERCASE(argument)	String value, function call, or variable	String in uppercase

Examples:

- Function: UPPERCASE("eXamPle sTring")

  The result is EXAMPLE STRING.

- Function: UPPERCASE(state)

  The result is the State column value in upper case.

FIND

Searches for the first occurrence of a substring within a string and returns the position of the first occurrence.

Note: This function is case sensitive.

Syntax	Input	Output
FIND(search_string, source_string, from_index)	• search_string: Substring, function call, or variable. • source_string: Main string, function call, or variable. • from_index: Index position in the main string from where the search should start. Numeric value, function call, or variable.	Numeric value (integer). When the substring does not exist in the main string, -1 is returned.

Example:

Function: FIND("morning", "Good morning")

The result is 5.

SEARCH

Searches for a substring within a string and returns the position of the first occurrence of the substring.

Note: This function is case insensitive.

Syntax	Input	Output
SEARCH(search_string, source_string, from_index)	• search_string: Substring, function call, or variable. • source_string: Main string, function call, or variable.	Numeric value (integer). When the substring does not exist in the main string, -1 is returned.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Syntax	Input	Output
	• from_index: Index position in the main string from where the search should start. Numeric value, function call, or variable.	

Examples:

- SEARCH("Morning", "Good morning")

  The result is 5.

- SEARCH("World","Hello world!")

  The result is -1.

SUBSTRING

Retrieves a substring from a string at the specified index position and for the specified length.

Syntax	Input	Output
SUBSTRING(source_string, start_index, length)	• source_string: String, function call, or variable. • start_index: Position in the string from where the substring is extracted. Numeric value, function call, or variable. • length: Length of the substring that should be extracted.	String

Example:

SUBSTRING("Hello, Good Morning", 7, 4)

The result substring is 'Good'.

Date and time functions

Use date and time functions to calculate or reformat the date and time column values.

NOW

Returns the current date and time of the instance in ISO format (YYYY-MM-DD hh:mm:ss).

Syntax	Input	Output
NOW()	No arguments are required for this function	ISO format of current date and time

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example:

Function: NOW()

The result is the current date and time in ISO format.

TODAY

Returns the current date with time offset to start of the day in ISO format in UTC time zone.

Syntax	Input	Output
TODAY()	No arguments are required for this function	Current date with time offset to start of the day in ISO format.

Example:

Function: TODAY()

The result is the current date and start time of the day in ISO format.

TIMEDIFF

Evaluates the time duration difference between two dates.

Syntax	Input	Output
TIMEDIFF(argument1, argument2)	Date in ISO format (YYYY-MM-DD hh:mm:ss) as string or variable	Duration

Examples:

- Function: TIMEDIFF("2021-05-02 9:10:12", "2021-04-07 6:2:23")

  The result is 25 03:07:49.

- Formula: TIMEDIFF(sys_created_on, NOW())

  The result is the time duration difference between the sys_created_on date and the current date of the system.

DATEDIF

Evaluates the difference between the two dates in days, months, or years.

Syntax	Input	Output
DATEDIF(start_date, end_date, date_difference_unit)	• start_date: Date in ISO format (YYYY-MM-DD or YYYY-MM-DD hh:mm:ss) as string or variable. • end_date: Date in ISO format (YYYY-MM-DD or	Numeric duration value based on the specified date difference unit.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Syntax	Input	Output
	YYYY-MM-DD hh:mm:ss) as string or variable. • date_difference_unit: Character String and either "Y","M", or "D" in lowercase or uppercase. Default is "D".	

Example:

Function: DATEDIF("2021-05-02 9:10:12", "2021-05-05 6:2:23 ","d")

The result is 3.

DATE

Creates a date from the specified individual year, month, and day values. The created date is in Coordinated Universal Time (UTC) time zone.

Syntax	Input	Output
DATE(year,month,day)	• year: Numeric value, variable or function. • month: Numeric value, variable or function. • day: Numeric value, variable or function.	Date in ISO format (YYYY-MM-DD hh:mm:ss)

Example:

Function: DATE(2021,5,2)

The result is 2021-05-02 00:00:00.

DAY

Retrieves the numerical day component from the specified date.

Syntax	Input	Output
DAY(date)	Date in ISO format (YYYY-MM-DD or YYYY-MM-DD hh:mm:ss) as string, variable, or function.	Numeric value (integer). The values range from 1 through 31.

Examples:

- Function: DAY("2021-05-029:10:12")

  The result is 2.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Function: DAY(NOW())

  The result will be the day component of the current date and time.

MONTH

Retrieves the numerical month component from the specified date.

Syntax	Input	Output
MONTH(date)	date: Date in ISO format (YYYY-MM-DD or YYYY-MM-DD hh:mm:ss) as string or variable.	Numeric value (integer). The values range from 1(January) through 12(December).

Examples:

- Function: MONTH("2021-05-02 9:10:12")

  The result is 5.

- Function: DAY(NOW())

  The result will be the month component of the current date and time.

YEAR

Retrieves the year component from the specified date.

Syntax	Input	Output
YEAR(date)	Date in ISO format (YYYY-MM-DD or YYYY-MM-DD hh:mm:ss) as string, variable, or function.	Numeric value (integer)

Examples:

- Function: YEAR("2021-05-02 9:10:12")

  The result is 2021.

- Function: YEAR(NOW())

  The result will be the year component of the current date and time.

WEEKDAY

Returns the numerical day of the week for the specified date. The day range is 1 (Sunday) through 7 (Saturday).

Syntax	Input	Output
WEEKDAY(date)	date: Date in ISO format (YYYY-MM-DD or YYYY-MM-	Numeric value (integer)

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Syntax	Input	Output
	DD hh:mm:ss) as string or variable.	

Example:

Function: WEEKDAY("2021-05-02 9:10:12")

The result is 1.

TEXT

Retrieves the specific date components in a date in string format.

Syntax	Input	Output
TEXT(date, format_text)	• date: Date in ISO format (YYYY-MM-DD or YYYY-MM-DD hh:mm:ss) as string, variable. • format_text: Date components as string or variable that are to be extracted.	String

Example:

TEXT("2022-08-17 9:10:12","yyyy-MM")

The result is 2022-08.

DATEVALUE

Converts a date in text format into a date in ISO format.

Syntax	Input	Output
DATEVALUE(date_text)	date_text: Date stored as text must be in YYYY-MM-DD format.	Date in ISO format (YYYY-MM-DD hh:mm:ss) as string.

Example:

Function: DATEVALUE("2021-05-02")

The result is 2021-05-02 00:00:00.

WORKDAY

Returns the nearest working day for the specified input date by excluding the specified holidays and weekends before or after the specified n number of days.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Syntax	Input	Output
WORKDAY(start_date, days, holiday 1,holiday 2, ..., holiday n)	• start_date: Date in ISO format (YYYY-MM-DD or YYYY-MM-DD hh:mm:ss) as string or variable. • days: Number of days as a numeric value, string, or function. • holiday 1...holiday n (Optional): Date in ISO format (YYYY-MM-DD or YYYY-MM-DD hh:mm:ss) as string or variable.	Date in ISO format (YYYY-MM-DD hh:mm:ss) as string.

Example:

Function: WORKDAY("2022-08-17 9:10:12",2)

The result is 2022-08-19 00:00:00.

NETWORKDAYS

Calculates the number of working days between two dates by excluding weekends and specified holiday dates. Number of working days includes the start date and end date.

Syntax	Input	Output
NETWORKDAYS(start_date,end_date,holiday 1,holiday 2, ... holiday n)	• start_date: Date in ISO format (YYYY-MM-DD or YYYY-MM-DD hh:mm:ss) as string or variable. • end_date: Date in ISO format (YYYY-MM-DD or YYYY-MM-DD hh:mm:ss) as string or variable. • holiday 1,holiday 2, ... holiday n (optional) : List of holidays that should be excluded while calculating working days.	Numeric value (integer)

Example:

Function: NETWORKDAYS("2022-08-17 20:10:12","2022-08-19 9: 10:12")

The result is 3.

Logical functions

Use logical functions to perform logical operations on column values.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

AND

Performs a logical AND operation on the arguments.

Syntax	Input	Output
AND(argument 1, argument 2)	String, function call, or variable	True or false

Examples:

- Function: AND(2>3, 4<5)

  The result is false.

- Formula: AND(LENGTH(sys_created_by)>25, LENGTH(sys_updated_by)>25)

  The result is true only when the number of characters in both the sys_created_by and sys_updated_by columns are greater than 25. Otherwise, the result is false.

IF

Executes the specified statements based on the Boolean output of the conditional expression.

Syntax	Input	Output
IF(<conditional_expression>, <do_this_when_true>, <do this_when_false>)	• conditional_expression: Logical conditional expression, function call, or variable **Note:** Logical comparison of strings is not supported in the conditional expression. • do_this_when_true: String, numeric value, function call, or variable that is returned when the condition evaluates to true • do_this_when_false: String, numeric value, function call, or variable that is returned when the condition evaluates to false	String, numeric value, function call, or variable based on the Boolean output of the conditional expression.

Examples:

- Function: IF(number_of_incidents >= 5, "High", "Medium")

  If the number of incidents is greater than 5, the string ‘High’ is returned. In other cases, the string ‘Medium’ is returned.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Function: IF(LENGTH(full_name) > 100, "Number of characters exceed the limit", "Number of characters within the limit")

  If the number of characters for the full_name column is above 100, the string 'Number of characters exceed the limit' is returned. Otherwise, the string 'Number of characters within the limit' is returned.

OR

Performs logical OR operation on the arguments.

Syntax	Input	Output
OR(argument 1, argument 2)	Conditional expression, function call, or variable	True or false

Examples:

- Function: OR(2>3,4<5)

  The result is true.

- Formula: OR(LENGTH(first_name)>25, LENGTH(last_name)<25)

  The result is true when the number of characters in the first_name column is greater than 25 or the number characters in the last_name column is less than 25. Otherwise, the result is false.

IFERROR

Evaluates expression 1 and returns the expression 1 value when there are no errors in the expression 1. When an error occurs while evaluating expression 1, expression 2 is evaluated and expression 2 value is returned.

Syntax	Input	Output
IFERROR(expression 1, expression 2)	• expression 1: Arithmetic, Logical expression, function call, String, numeric value or variable. • expression 2: Arithmetic, Logical expression, function call, String, numeric value or variable.	Result of expression 1 when there are no errors in expression 1. Otherwise, result of expression 2.

Example:

Function: IFERROR(MULTIPLY(snr_factor, signal), MULTIPLY(default_factor, signal))

If the snr_factor value is a valid number, the multiplied value of snr_factor with signal is returned. If the snr_factor value is not a valid number, the multiplied value of default_factor value with signal is returned.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add a formula to a column in Table Builder

Use a predefined function or create a formula to calculate a value for a column without writing a script in Table Builder. You can use two or more functions to create a formula according to your requirements.

Before you begin

- The scope of the table should be within the scope of the application. Otherwise, you cannot add a formula to a column in the table that is outside the scope of the app.
- Role required: personalize_dictionary or AES user role and delegated developer permissions. For more information, see Delegate developers using AES.

Procedure

1. From the **Data** or **Forms** tabs, click the Open side panel icon ( ) for a column.
2. In the **Config** tab of **Formula** section, click **+ Add**.
3. Enter a predefined function or a formula in the **Formula editor** modal.



For example, to concatenate the First name column value with the Last name column value and populate the Full name column value, enter CONCATENATE(`first_name,last_name`). For more information about available predefined functions and example formulas, see Formulas for column values in Table Builder

For syntax guidance and examples, click the function in the auto-suggestion that appears as you begin typing.

Note: You cannot add a formula for unsupported columns, columns that are function fields, and the columns that already have script as the calculated value type.

4. Click **Submit**.

Result

The formula is validated and added to the column if it doesn't have any errors.

Policies and rules properties in Table Builder

You can configure the basic field policies and rules for any field that you work with in Table Builder.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

UI Policies

The UI Policies section enables you to configure how forms appear based on roles and user input.

To understand the basic field properties, see Create a UI policy in Table Builder.

The following table shows the field descriptions for the Policy details section when adding or editing a UI policy.

Policy details

Field	Description
Short description	Short summary of the UI policy.
Table	Data table that the policy applies to.
Domain	Domain that the choice resides in.
Order	Sequence for processing, from the lowest number to the highest number. If two policies conflict, the UI policy with the higher number runs. For inherited UI policies, the extended (child) table's UI policies run first. Then, the base table UI policies run, both from the lowest to the highest specified value.
Active	Status of the UI policy. Only active UI policies are applied.
Apply to all views/Apply to view [Advanced settings]	Option for specifying whether the UI policy applies to all form views or specific views. For more information on form views, see View management.
On load [Advanced settings]	Option for specifying that the UI policy behavior should be performed OnLoad as well as when the form changes. Check or clear the UI policy **On load** check box to control whether it runs every time a form is loaded and conditions are satisfied. In the following example, an administrator does not want an incident to enter the **Awaiting user info** state unless the user provides an explanation to the customer. The administrator creates a UI policy with the following settings: • In the When these conditions are met section, adds the condition **[State] [is] [Awaiting user info]** and clears the **On load**

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Policy details (continued)

Field	Description
	check box. The UI policy applies only when the state is changed to **Awaiting user info**. • In the UI Policy Actions list, creates a record that makes the **Additional comments** field required when the condition is met.
Reverse if false [Advanced settings]	Option for specifying that the UI policy action should be reversed when the conditions of its UI policy evaluate to false. When the conditions are true, actions are taken and when they change back to false, the actions are reversed (undone).
Inherit [Advanced settings]	Option for specifying whether extended tables inherit this UI policy. When a child table has an inherited UI policy from its parent table, the UI policy on the child table always runs first, regardless of the **Order** of the UI policies. In the following example: • A child table has a UI policy with the**Order** value **500** that shows the **Urgency** field when its conditions are met. • Its parent table has a UI policy with the same conditions that hides the **Urgency** field. The parent table UI policy has the **Order** value **100**. • Although the parent table **Order** field has a lower value, the child UI policy runs first and then the parent UI policy runs. When the conditions are met, the **Urgency** field is hidden.

The following table shows field descriptions for the When these conditions are met section when adding or editing a UI policy.

When these conditions are met

Field	Description
Condition set	Conditions that, if fulfilled, cause the UI policy to be applied. Conditions are built with the condition builder by using logic statements. To set conditions using a script, use a client script instead. Conditions are only rechecked if a user manually changes a field on a form. If the change is made by a UI action, context menu

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

When these conditions are met (continued)

Field	Description
	action, or through the list editor, it is not evaluated.

The following table shows field descriptions for the Do the following section when adding or editing a UI policy.

Do the following

Field	Description
Field name	Field on the selected table to which the UI policy performs an action if conditions are met. **Note:** If the specified field is not on the form, the UI policy performs the action on the variable with the same name.
Mandatory	List for specifying how the UI policy affects the required state of the field. The choices are the following: • **Leave alone** • **True** • **False**
Visible	List for specifying how the UI policy affects the visible state of the field, that is, whether it appears. The choices are the following: • **Leave alone** • **True** • **False**
Read only	List for specifying how the UI policy affects the read-only state of the field. The choices are the following: • **Leave alone** • **True** • **False**
Clear the field value	Option to clear the specified field if the conditions are met.

Access control rules

The Access control rules section enables you to configure access to the specified resource based on role, condition, and script criteria being met.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The following table shows field descriptions for the Access control rules section.

Field	Description
Name	Names each user who has permission to access the form or data. • Enter the names as firstname.lastname in lower case letters, separated by a period (for example, `john.smith`). Each name must have a corresponding user record in *`hi.servicenow.com`*. • Enter multiple names and separate them by commas if multiple users have permission to access the form or data. • Enter an asterisk (*) as the name to enable all users to access the form or data.
Description	Overview of what the access control rule restricts or enables.
Type	Type of object the access controls, including the following: • Client-callable script includes • Processor • Record • UI page • UX route For more information on object types, see ACL matching requirements for objects.
Operation	Type of action the system can take on the specified object, such as delete or execute. Some objects, such as records, support multiple operations, while other objects, such as a REST_Endpoint, only support one operation. For more information on action types, see ACL rule types.
Active	Active state of the rule: **true** The rule is active and available for use. **false** The rule is inactive and unavailable for use.
Condition	Any conditions that apply to the rule.
Roles	Any roles the rule applies to.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Admin overrides	Option to force the rule's evaluation for admin overrides at the access level. For more information on admin overrides, see Evaluate the admin override at the access level.
Advanced	Option to associate a script with the rule.

Client Scripts

The Client Scripts rules section enables you to create scripts that control how form fields appear based on defined criteria.

For a complete list of values for client script fields, see Client script form

The following table shows field descriptions for the Client Scripts rules section.

Field	Description
Name	Name for the script.
Description	Overview the script's functionality and purpose.
Table	Table or form that the script is available for.
Active	Active state of the script: **true** The script is active and available for use. **false** The rule is inactive and unavailable for use.
UI type	Type of application the script is available for, such as desktop or mobile.
Type	When the script runs on the table or form, including the following: • onCellEdit • onChange • onLoad • onSubmit
Field name	Field the script is run on.
View	Views on which the client script runs (not applicable to global scripts).
Inherited	Indicates whether the client script applies to extended tables.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Business Rules

The Business Rules section enables you to create business rules to accomplish tasks like automatically changing values in form fields when certain conditions are met.

For a complete list of values for business rules, see Create a business rule.

The following table shows field descriptions for the Business Rules rules section.

Field	Description
Name	Name for the business rule.
Table	Table or form that the business rule runs on.
Active	Active state of the business rule: **true** The business rule is active and available for use. **false** The business rule is inactive and unavailable for use.
When	When the business rule executes: **display**, **before**, **async**, or **after** the database operation is complete.
Filter conditions	Conditions created using the condition builder to determine when the business rule should run based on the field values in the selected table.
Role conditions	Roles that users who are modifying records in the table or form must have for the business rule to run.
Actions	Actions taken by the business rule on the specified fields, such as **insert**, **update**, and **delete**.
Set field values	Table or form fields that the action executes on.
Add message	Whether a message appears when this business rule is run: **true** A message appears when the business rule is run. **false** No message appears when the business rule is run.
Abort action	Whether the business rule aborts the current database transaction: **true**

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
	The business rule aborts the current database transaction when run. **false** The business rule doesn't abort the current database transaction when run. For example, on a before insert business rule, if the conditions are met, do not insert the record into the database.
Order	Indicates the sequence in which this business rule should run. If there are multiple rules on a particular activity, the rules run in the specified order specified, from lowest to highest.

Workspace view rules

The Workspace view section enables you to define rules to control how users view workspaces based on criteria.

The following table shows field descriptions for the Workspace view rules section.

Field	Description
Name	Name for the workspace view rule.
Table	Table or form that the workspace view rule is available for.
View	View that is used to render the form. The default view is used if this field is left empty or contains an invalid value.
Roles	Any roles the rule applies to.
Conditions	Any conditions that apply to the rule.
Hide section navigation	Option to enable or disable section navigation.
Disable section collapsing	Option to enable or disable section collapsing.
Default tab order	Order in which form tabs appear by default.

Workspace Builder

Workspace Builder enables you to quickly create and tailor custom workspaces where agents work on open records.

This video shows you how to create a workspace in Workspace Builder.

You can use Workspace Builder to quickly create workspaces for workspaces built starting with Brazil. Workspace Builder enables you to edit the workspace home page, list pages, and record page contextual side panel directly in the Workspace Builder preview canvas. For workspaces created before Brazil, or if you need more robust functionalities and configurations, edit the workspace in UI Builder from a tab within AES.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Get started

Explore  Explor using Workspace Builder to create workspaces.	Configure  Configure Workspace Builder.
Build  Build and customize a workspace.	Reference  Get details about Workspace Builder components.

Exploring Workspace Builder

Workspace Builder for App Engine is a streamlined, no-code environment that enables you to create a custom workspace from within App Engine Studio (AES) quickly and efficiently.

What to explore next

To learn more about configuring and using Workspace Builder, see:

- Configuring Workspace Builder
- Using Workspace Builder
- Workspace Builder reference

Accessing Workspace Builder

You can access Workspace Builder in both ServiceNow Studio or App Engine Studio (AES).

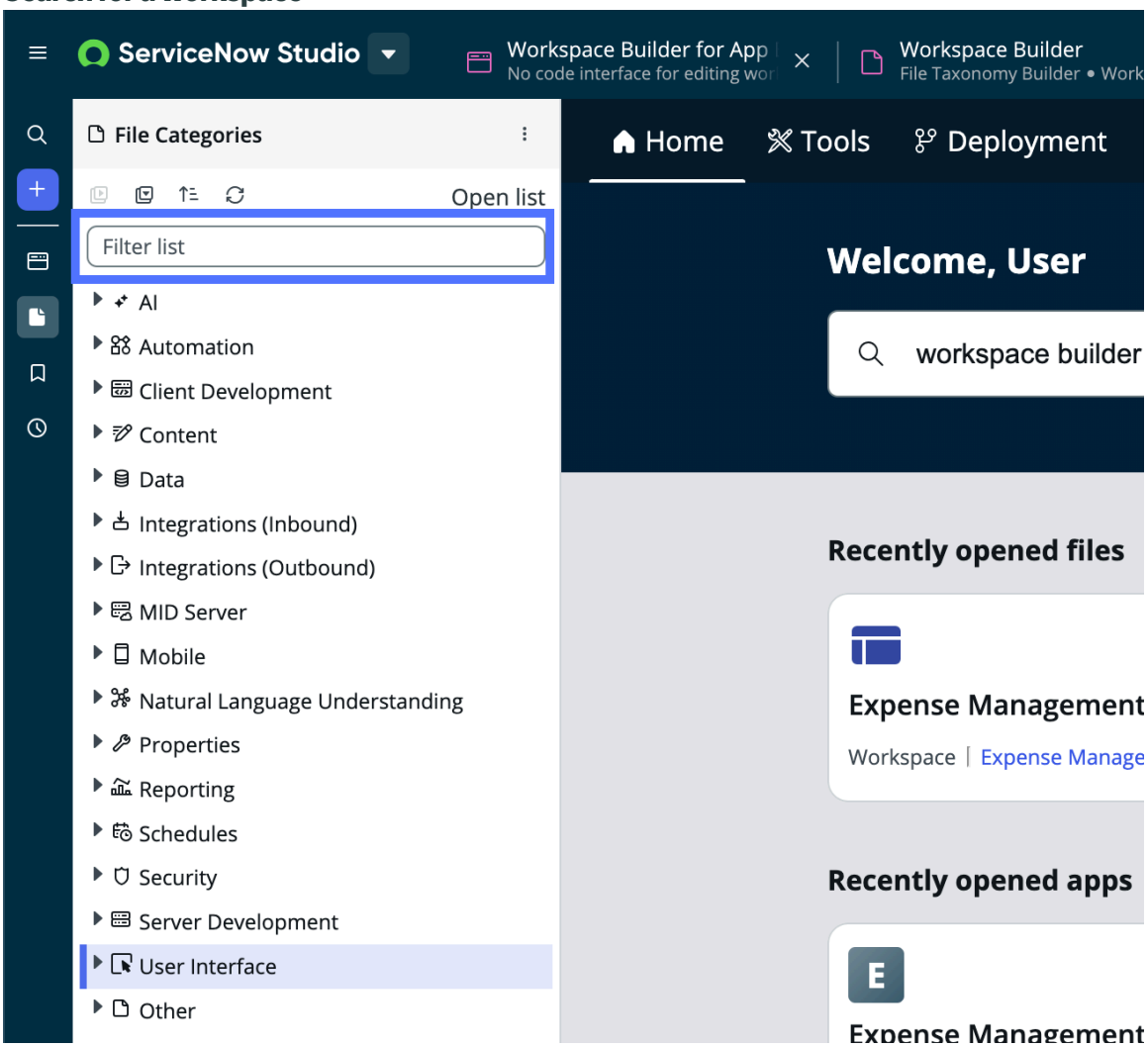
Workspace Builder in ServiceNow Studio

To open an existing workspace in Workspace Builder using ServiceNow Studio, complete the following steps:

1. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
2. Open the workspace in one of the following ways:
   - Search for the workspace by name or code snippet and select it. For details, see Find an app or app file using code search.
   - Search for it in the Filter list bar at the top of the Navigator panel.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Search for a workspace



- Find it in the Navigator panel using the following steps:

  a. Select the file categories ( ) next to the Navigator panel.

  b. Expand the **User Interface** section.

  c. Select **Workspace**.

  d. Scroll to and then select the workspace you want to open.

Access Workspace Builder in AES

To open an existing workspace in Workspace Builder using AES, complete the following steps:

1. Navigate to **All** > **App Engine** > **App Engine Studio**.
2. From the My Apps page, open the application that contains the workspace.
3. Next to the workspace you created, select the additional actions icon (•••) and then select **Edit**.
4. Select **Workspace**, and then select **Begin**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Sample workspaces you can build

Workspace Builder provides a basic workspace that includes a home page, record pages, lists, an Analytics Overview, and other functions.

The workspace that you create in Workspace Builder includes several default pages. You can use these pages as-is or edit them to suit your business needs.

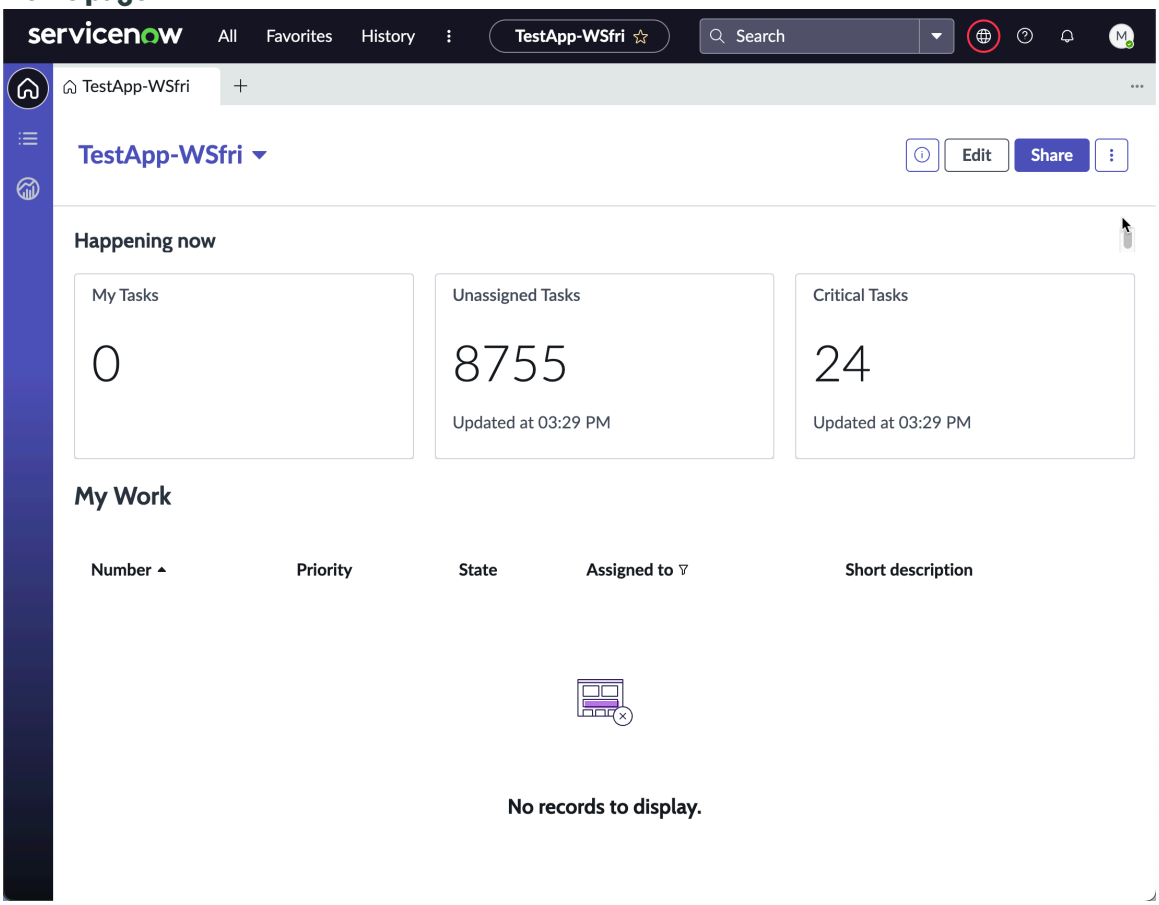
The following are some examples of the page customizations you can apply to the workspace after you add it:

- Home page
- List page
- Record page
- Search page
- Analytics Overview

Workspace home page

Note: You can make changes to the home page, but editing all other pages requires creating a variant or copy of the page. If you edit the pages directly, then they won't get the latest code in a system upgrade.

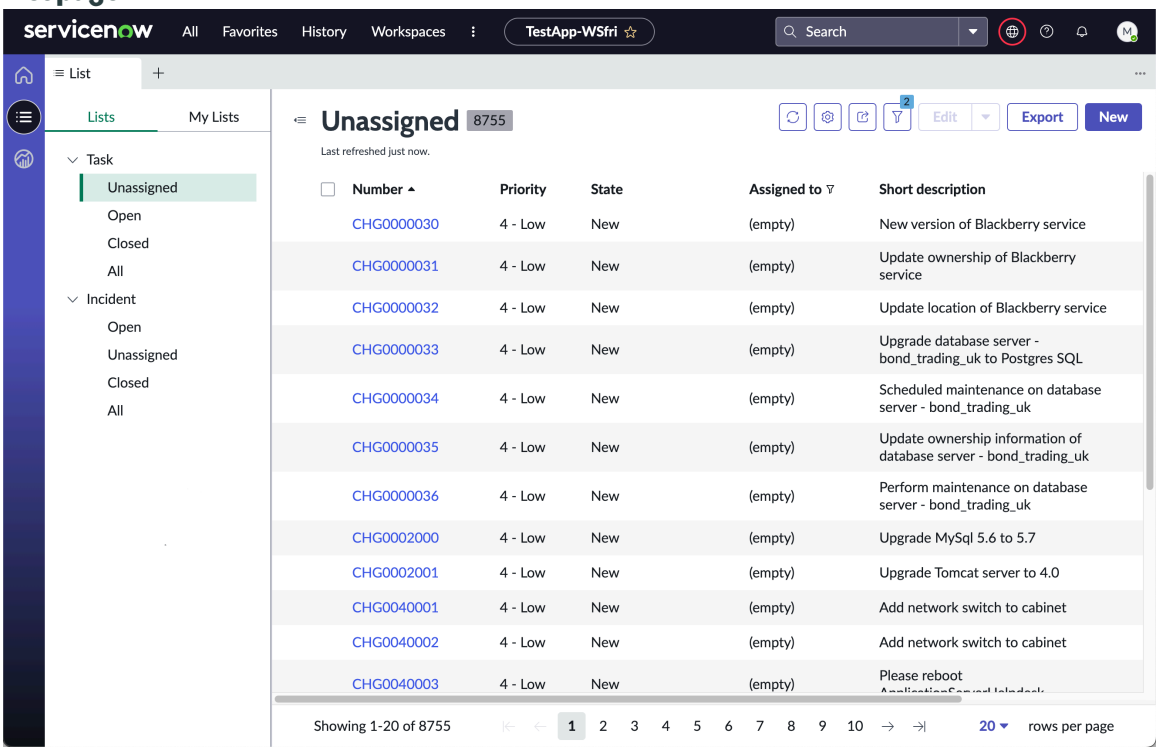
Home page



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

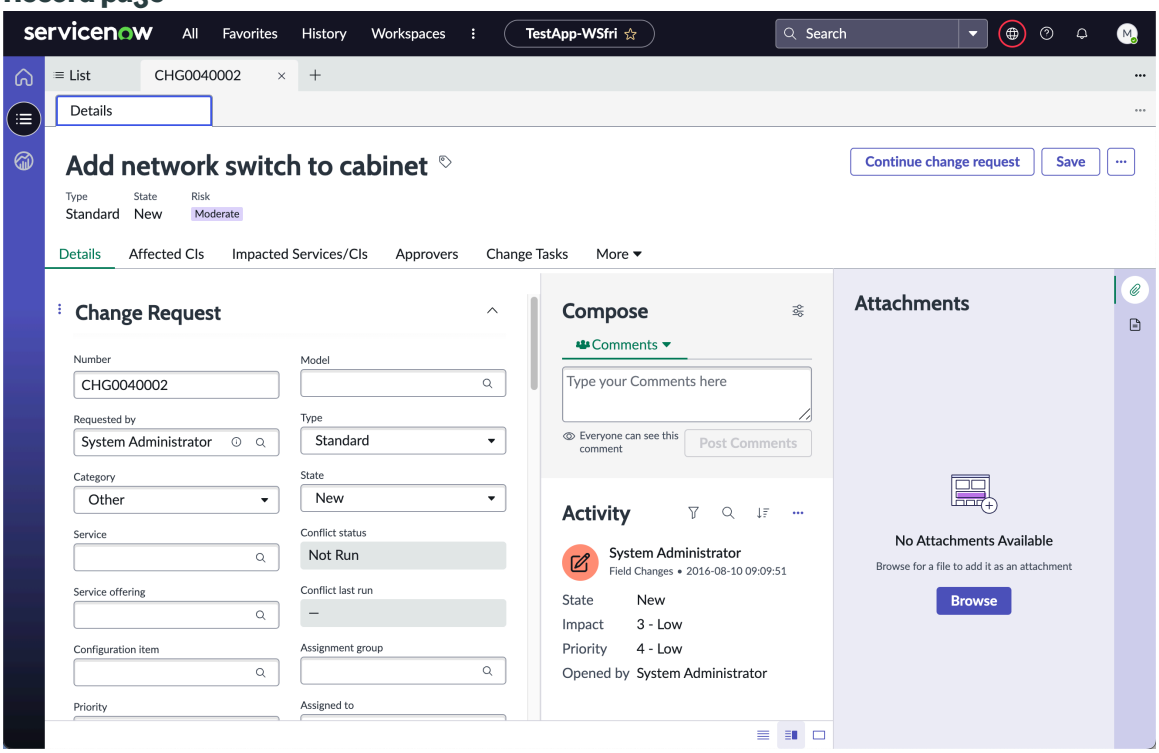
Workspace list page

List page



Workspace record page

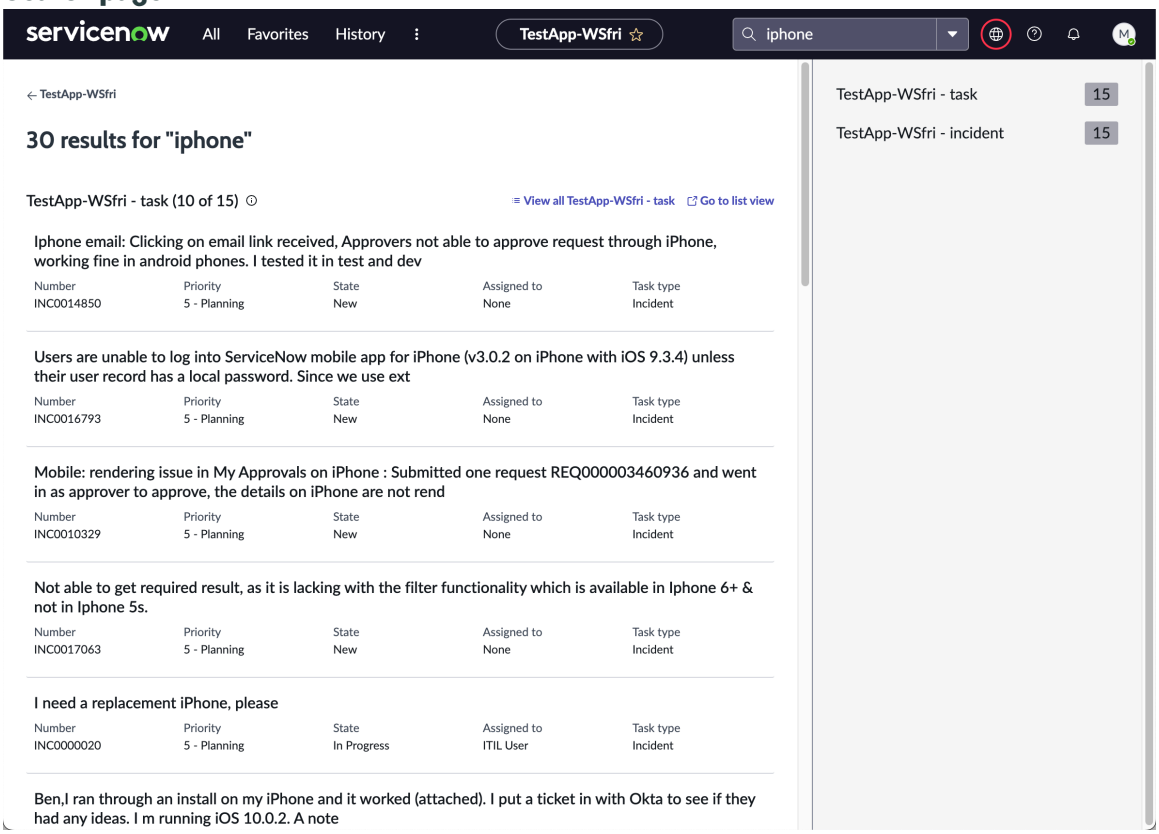
Record page



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Workspace search page

Search page



Workspace Analytics Overview

The Analytics Overview includes the following components:

Analytics Q&A

A field where you can ask natural language questions about the trends and health of your processes. (The Natural Language Query feature is required.)

Indicator Scorecard

A list of the Performance Analytics indicators that you are allowed to access based on your roles. Depending on the configuration, selecting an indicator can open it in KPI Details.

Dashboards Overview

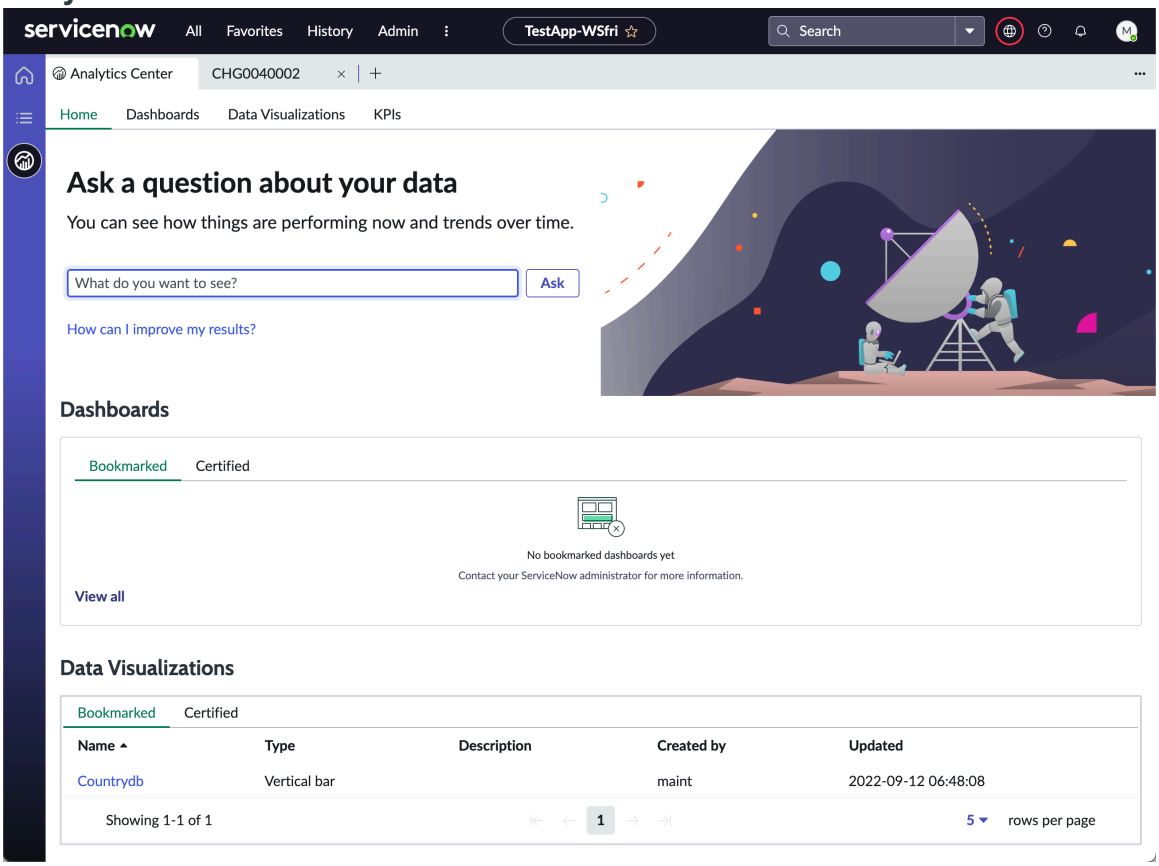
A set of tiles for each dashboard on the instance that you are allowed to access. Selecting a tile opens that dashboard.

> **Note:** You can create and edit dashboards in-line. For more information, see Configure Platform Analytics dashboard details.

For more information on Analytics Overview, see Platform Analytics overview.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Analytics Overview



Parts of a workspace in Workspace Builder

Every workspace has parts that enable workspace users to complete their tasks.

Build your workspace by including the following pages and elements.

Workspace pages and elements

Component	Definition
Home	Landing page for the workspace, which contains a dashboard and appears to users when first accessing the workspace. **Note:** The home page is an essential part of the workspace, and thus can't be hidden or removed. For more information on configuring the home page, see Customize a workspace home page in Workspace Builder. For information on configuring the basic settings for a workspace, see Configure workspace settings in Workspace Builder.
List	List pages for records from the specified tables, which help users to find individual records in a workspace.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Workspace pages and elements (continued)

Component	Definition
	Note: If you remove the list page/ route, you must use UI Builder to edit the workspace, not Workspace Builder. You can create a list category and then add filtered lists for each list category. Filtered lists apply unique conditions, ensuring that only the filtered subset of records appears. For more information on adding and editing lists in Workspace Builder, see Create lists for a workspace in Workspace Builder.
Analytics	The customizable Analytics Overview enables users to track and analyze records and usage with dashboards, data visualizations, and insights on your instance. **Note:** If Analytics Overview page/route doesn't exist for a workspace, Workspace Builder is still available, but the Analytics Overview tab doesn't appear in the workspace. For more information on enabling the Analytics Overview for a workspace, see Enable analytics for a workspace in Workspace Builder.
Record pages	Pages in a table that give users useful information to work on a task, issue, or incident. Useful information can include priority, state, activity, and so on. Create and customize record pages by using containers and components in different sections of the layout. You can also add a Playbook Experience to a record page. For more information, see Add a record page for a workspace in Workspace Builder.

Note: Your workspace must have a home page and lists for you to edit it in Workspace Builder. If they aren't present, you must edit the workspace in UI Builder.

Layout for Workspace Builder

Workspace Builder has a navigation panel, header, preview pane, and configuration panel to enable bulding custom workspaces.

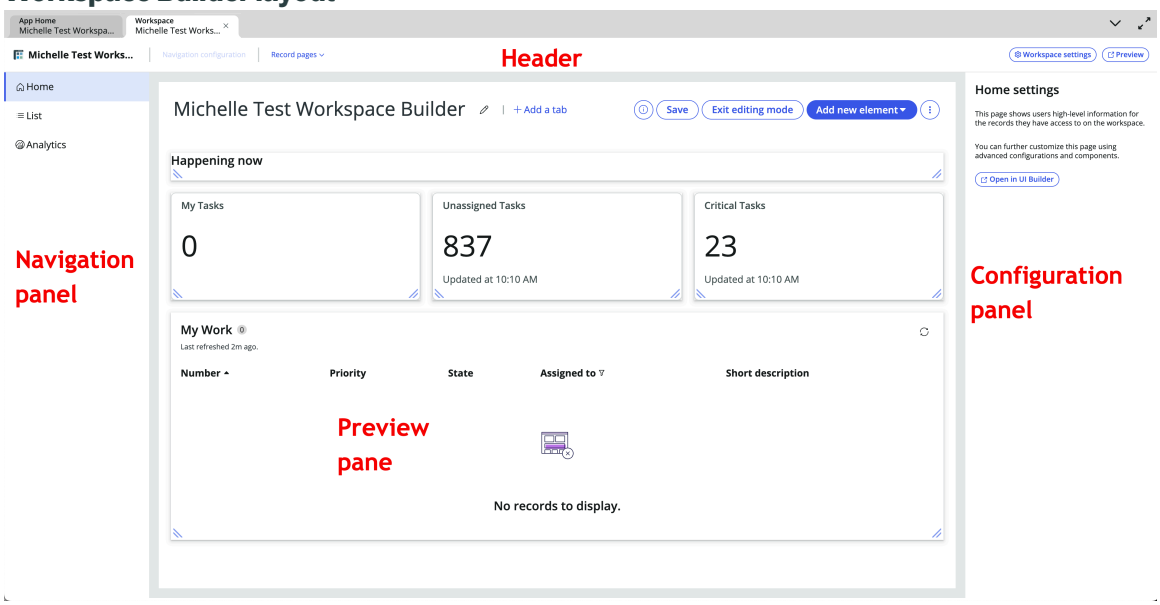
The first time you access Workspace Builder, an onboarding modal displays a brief tour of what you can do in Workspace Builder.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Workspace Builder consists of the following sections:

- A left navigation panel to configure the pages that users can navigate to within the workspace.
- A preview of the workspace in the middle canvas.
- A right configuration panel for configuring the selected workspace component.
- An option to preview the workspace in a new browser tab.

Workspace Builder layout



The Workspace Builder header has the following navigation options just below the tab, where you can work with the following:

- Navigation configuration
- Record pages

Configuring Workspace Builder

Install and configure Workspace Builder to enable users to build custom workspaces.

Configuration overview

1. Install App Engine Studio from the ServiceNow Store.
2. Install the Workspace Builder for App Engine app, if it's not already installed on your instance.

Installing Workspace Builder

Workspace Builder is available on the ServiceNow Store.

Check your entitlements to determine if you have access to Workspace Builder.

When you install Workspace Builder, you also install the following related items:

- Roles and groups
- Tables
- Plugins

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Installing Workspace Builder for App Engine from the ServiceNow Store

1. Navigate to the ServiceNow Store.
2. In the search bar, enter `Workspace Builder`.
3. Select the Workspace Builder for App Engine application.
4. Select **Buy**.
5. Log in to your ServiceNow instance.
6. Navigate to **All** > **Application Manager**.
7. Enter `Workspace Builder` in the search bar.
8. Select **Install**.

For more information about installing applications using the Application Manager, see Installing applications, plugins, and products.

Using Workspace Builder

Create a workspace in Workspace Builder before customizing its settings, home page, record pages, lists, and analytics.

Overview of creating a workspace

First, add a workspace and configure its settings. Then, you customize its parts, such a s the home page and record page.

Add and configure the settings of a workspace:

1. Add a workspace

   Create a workspace before customizing it.

2. Edit a workspace in Workspace Builder

   Define workspace settings to control the basic functionality of the workspace, such as name and record page navigation.

After you've created the workspace, customize how it will appear to users:

- Customize a workspace home page in Workspace Builder

  Customize what users see when first accessing the workspace, which is a dashboard.

- Add a record page for a workspace in Workspace Builder

  Use containers and components to customize the record for a workspace.

- Create lists for a workspace in Workspace Builder

  Create and customize list categories to add workspace pages that list table records.

- Enable analytics for a workspace in Workspace Builder

  Add analytics that enable users to track and analyze records and usage with dashboards, data visualizations, and insights on the instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add a workspace

Create a workspace using Workspace Builder in ServiceNow Studio or App Engine Studio (AES). Workspaces provide agents, case managers, help desk professionals, and managers with tools to help answer customer questions and resolve customer problems.

Before you begin

Role required:

- admin or Guided Application Creator for ServiceNow Studio.
- sn_app_eng_studio.user or delegated_developer for AES. For more information, see Delegate developers using AES.

About this task

Workspaces are primarily used for request and fulfillment processes, such as a service desk to manage tickets.

When you add a workspace you're creating a basic version of it, which you must then continue editing in Workspace Builder before it's ready to deploy. For more complex customizations, edit the workspace in UI Builder.

If you create a workspace from an App Engine Studio workspace experience template, you can edit the home page, lists, and record pages in Workspace Builder. However, if you create the workspace using an application template, you must edit some of its pages and objects in UI Builder.

Procedure

1. Navigate to ServiceNow Studio or AES.
   - To add a workspace using ServiceNow Studio, complete the following steps:
     a. Navigate to **All** > **App Engine** > **ServiceNow Studio**.
     b. Select the create icon ( ) next to the Navigator panel and select **File**.
     c. Select the **User interface** file type, and then select **Workspace**.
     d. Select **Continue**.
     e. Select **Begin**.
   - To add a workspace using AES, complete the following steps:
     a. Navigate to **All** > **App Engine** > **App Engine Studio**.
     b. From the My Apps page, open the application you want to build a workspace for.
     c. In your application, next to Experience, select **Add**.
     d. Select **Workspace**, and then select **Begin**.
2. On the form, fill in the fields.

Field	Description
Name	Name of the workspace that appears to users. By default, the workspace shares the same name as your application. **Tip:** Use a name that uniquely identifies the workspace from other experiences. For example, `Office art requests workspace`.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Description	Description of the workspace.
URL	Unique web address of the workspace. By default, the workspace URL is based on the application name, but you can edit it.
Roles	User roles to limit who can access the workspace. ○ You must specify at least one role to grant access to the workspace. ○ To use a custom role for your workspace, you must create one in **Security** first. For more information, see Add application security.

3. Select **Continue** to define data for your workspace.
4. On the form, fill in the fields.

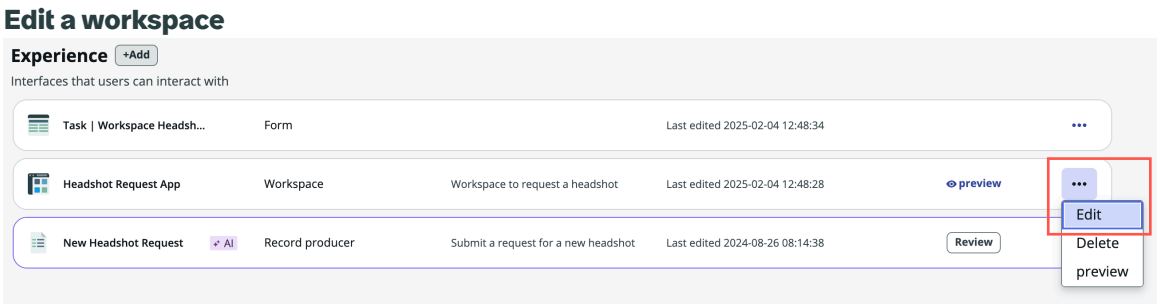
Field	Description
Primary Table	Data table that workspace users view or update.
Secondary Tables	Additional data tables that workspace users view and update.

Note: You can select only tables with default views. You can select from two sections: Tables that are already available in your app, and all tables outside the app scope.

Workspace Builder automatically generates a record page for the table you select.

5. Select **Continue**, and then select *Done*.
After you create the workspace in AES, you must edit it in Workspace Builder by adding or configuring components on each page.
6. Next to the workspace you created, select the additional actions icon (•••) and then select **Edit**.

You can also select the bar that lists the workspace in the Experience section of your app.



7. **Optional:** Edit your workspace home page in the In-line editor by selecting **Preview**.

a. On your workspace home page, select **Edit**.
If you don't see **Edit**, you may not have the permissions needed to edit the workspace home page.

b. If the workspace is missing an Access Control List (ACL), enter the roles that should have access in the Add user roles to continue modal and select the **Add roles** button.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For more information, see Configure workspace settings in Workspace Builder and Access Control Lists (ACLs).

c. Adjust the home page elements and widgets by resizing, reorganizing, or reconfiguring them.

Editing your workspace home page works similarly to editing a dashboard on your workspace. For more information, see Edit Platform Analytics dashboards.

Note: If the workspace was created before Tokyo, you must edit it in UI Builder. See UI Builder for more information.

If you created the workspace that contains a technical dashboard, Workspace Builder prompts you to **Open in UI Builder** when you try to edit the workspace.

d. Select **Add new element** to add items to your workspace home page.

e. Select **Exit Editing Mode** to stop editing in the In-line editor.

8. Edit the workspace in Workspace Builder. For more information, see Edit a workspace in Workspace Builder.

You can also add more record pages to the workspace, if needed.

Note: The default workspace record pages are read-only and can't be edited. To change the content of each record page, create a page variant and edit the variant.

What to do next

For more information on using Workspace Builder to customize your workspace, see Using Workspace Builder.

Configure workspace settings in Workspace Builder

Define workspace settings in Workspace Builder to control the basic functionality of the workspace, such as name and record page navigation.

This video shows you how to perform the following procedure.

Before you begin

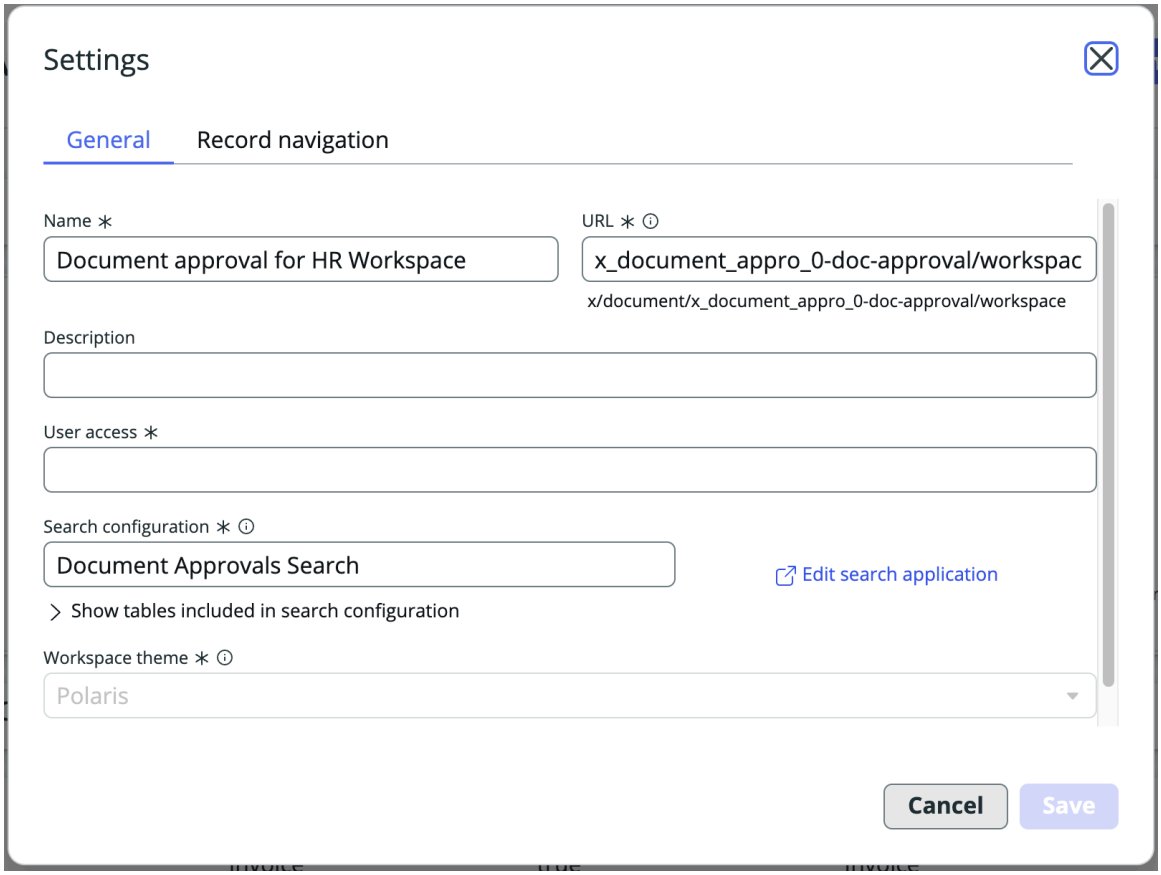
Role required:

- admin or Guided Application Creator for ServiceNow Studio.
- sn_app_eng_studio.user or delegated_developer for AES. For more information, see Delegate developers using AES.

Procedure

1. Navigate to Workspace Builder.
   For details on how to open Workspace Builder in ServiceNow Studio or AES, see Accessing Workspace Builder.
2. Display the Settings form by selecting **Workspace settings** in the Workspace Builder header.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



3. On the **General** tab of the form, specify or change the basic settings for the workspace by filling in the editable fields.

Note: Some settings are specified when adding a workspace, and can't be changed.

General workspace settings

Field	Description
Name	Name for the workspace, which is visible to users.
URL	Relative URL for the workspace. The path is automatically created based on the name of your workspace, and is preceded by your instance URL. However, you can customize the path in this field.
Description	Brief description of the purpose of the workspace.
User access	People and groups that can access the workspace. To give unrestricted access to the workspace, leave this field empty.
Search configuration	Source of information when users perform a search in the workspace. You can reuse an existing search configuration by selecting the profile from the menu. Select the **Edit search application** link to make additional, advanced configuration changes.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
	If you select an AI Search profile, the following features are available: ○ Selection of multiple facet values simultaneously, which you can apply using the condition builder. ○ Ability to view parent documents with linked attachments, which you can download. ○ Dark mode. **Note:** AI Search must be enabled for your instance to select an AI Search profile. For more information, see AI Search.
Show tables included in search configuration	Option that displays one or more tables that the workspace searches as sources.
Workspace theme	Theme to define the visual style of the workspace, which you can select only if Polaris isn't enabled for the instance. Themes are retrieved from the UX Themes [**sys_ux_theme**] table..

Trouble?
Add only one **User access** role at a time. For more information, see this Knowledge article.

4. Select **Save**.
5. Specify what type of records users can create by selecting the **Record navigation** tab of the form and then filling in the fields.

Record navigation workspace settings

Field	Definition
Navigation type	How users interact with or navigate the workspace. **Tabbed navigation** creates separate tabs for each page, enabling users to work on multiple records at a time. Choose tabs if you anticipate that users must change between records quickly. Tabs also enable users to create records directly from the navigation.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Definition
	Note: The default navigation type for workspaces built in AES is tabbed, and can't be changed after the workspace is created. You must create the workspace in UI Builder to specify breadcrumb navigation. You can't change the navigation type after the workspace is created.
Table	Data table that is added to when a user creates a record. The table you select automatically has associated lists. **Note:** The tables specified when the workspace was created appear automatically. Select **Add a table** to incorporate additional tables into the workspace. You can add each table only once. To remove a table, select its delete icon ( ). No confirmation message appears, but you can select **Cancel** to revert the deletion.
Label	Brief label for the option that appears when users select the new record menu. Specify a label that tells users what the record is used for.

6. Select **Save**.

Customize a workspace home page in Workspace Builder

Edit the home page for a workspace in Workspace Builder. Adjust what users see when first accessing the workspace, which is a dashboard.

This video shows you how to perform the following procedure.

Before you begin

Role required:

- admin or Guided Application Creator for ServiceNow Studio.
- sn_app_eng_studio.user or delegated_developer for AES. For more information, see Delegate developers using AES.

About this task

You can edit workspace home pages in Workspace Builder only for workspaces created in Tokyo and going forward. You can edit older workspaces in UI Builder.

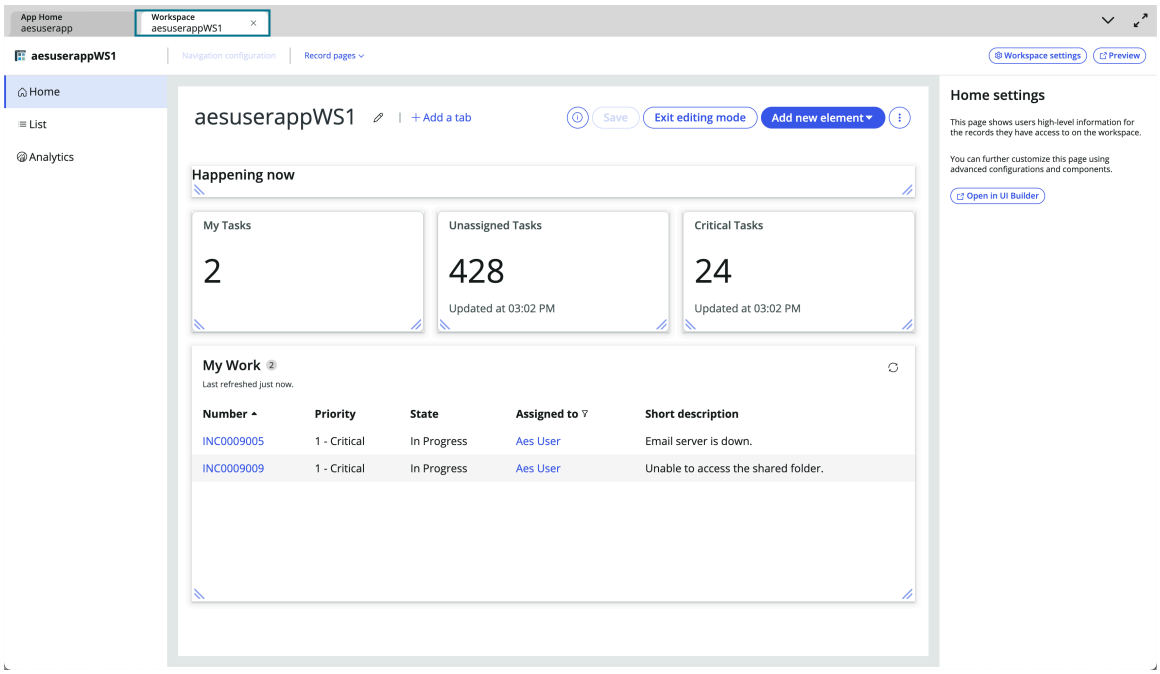
Note: The home page is an essential part of the workspace, and thus can't be hidden or removed.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to Workspace Builder.
   For details on how to open Workspace Builder in ServiceNow Studio or AES, see Accessing Workspace Builder.
2. Add an element to the page.
   You must be in the **Navigation configuration** tab. If you're already in the **Navigation configuration** tab, you can't select it.

   a. In the navigation panel, select **Home**.



   b. In the home page canvas, select **Add new element**.

   c. From the list, select the element or component that you want to add.

   For example, to add a graphic of data from the specified table, select **Data visualization** and choose a **New data visualization** or a **Saved data visualization**.

   You can add the following elements to the home page in Workspace Builder.

Home page elements

Element	Description	More information
Data visualization	Graphic, visual representation of current instance data. You can create a new data visualization or repurpose an existing data visualization by choosing it in the Select visualization from library modal and selecting **Add to dashboard**.	▪ Data visualizations in Platform Analytics ▪ Create a single score data visualization ▪ Developer site data visualization component listing

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Element	Description	More information
	In addition to data visualizations, you can select the **Library** element to add an already-saved data visualization to the home page.	
Filter	Refine the visualizations on a dashboard based on specified criteria. The filters can apply to both tables and indicators.	▪ Filters in Platform Analytics ▪ Developer site filter component listing
Heading	Title or text for the top of the home page or section.	Developer site heading component listing
Image	Static or animated images or pictures for the workspace.	Developer site image component listing
List - Simple	Table data that you can customize for the dashboard audience.	▪ Simple List widget or List - Simple Usage - Developer site ▪ Developer site List - simple component listing

The Workspace Builder components for home pages has details on home page elements. For more information, see Exploring Platform Analytics dashboards.

If you want to include additional elements that aren't available in the **Add new element** menu, you must edit the workspace in UI Builder. For a complete list of elements you can add in UI Builder, see the components documentation on the ServiceNow.

The element appears on the dashboard, where you can make edits.

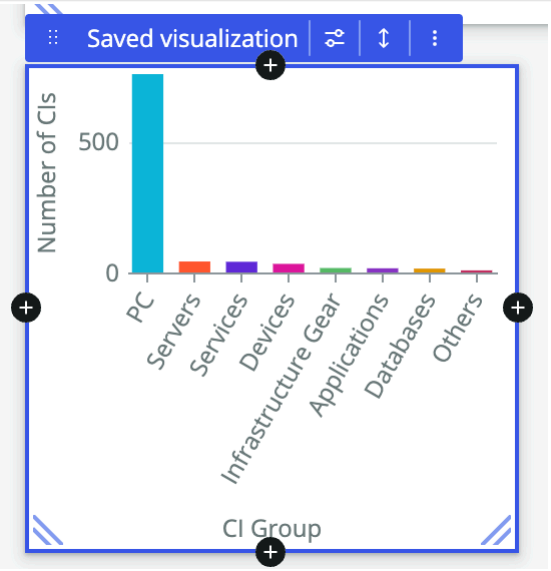
d. Drag the element to move or resize it.

e. In the canvas header, select **Save**.

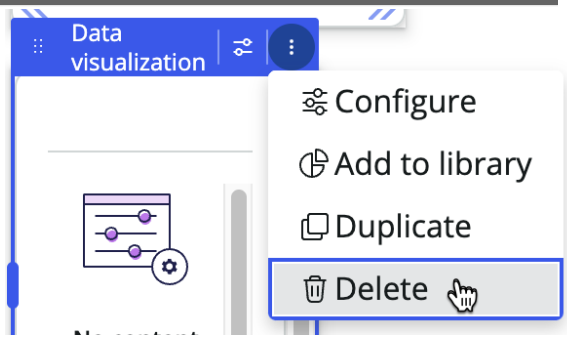
3. **Optional:** If needed, complete any additional changes to an element.

Option	Description
Edit an element	**a.** Select the element that you want to edit, such as a data visualization. (Optional) A header appears with the name of the element type.

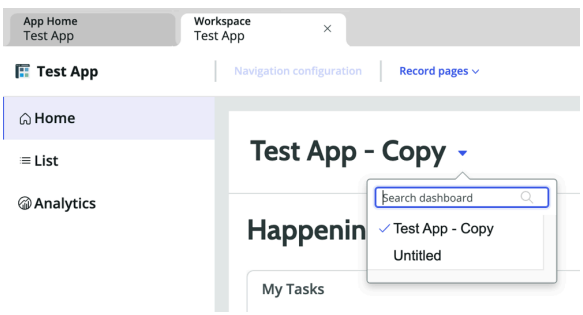
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Description
	 **b.** Select the settings icon ( ) in the element header to open the configuration panel within the canvas. **Note:** You can also select the con text menu icon ( ) and select **Con figure** to edit the element. **c.** Edit the configuration of the element in the configuration panel as needed. (Optional) For example, if you're editing a data visualization, you must specify the source table. For more information on edit ing components, see Add and configure components. **d.** In the canvas header, select **Save**.
Copy an element to reuse on the home page	Select the context menu icon ( ) and select **Duplicate** to copy the element. You can then make additional changes to the copied ele ment.
Add an element to the library for reuse	Select the context menu icon ( ) and select **Add to library** to include the element in your library to reuse on other dashboards.
Delete an element from the workspace dash board	**a.** Select the element that you want to delete. (Optional) A header appears with the name of the element type. **b.** In the header, select the context menu icon ( ) and select **Delete**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Description
	 **Note:** There's no confirmation mes sage. The element disappears from the dashboard.

4. **Optional:** If needed, complete any additional configuration changes to the dashboard.

Option	Description
Configure the dashboard layout	Select and drag an entire element to move it around the dashboard layout. (Optional) Resize an element by selecting and dragging the handlebars ( ) in the low er corners of the element.
Duplicate the dashboard	To make a copy of the dashboard: ○ Select the more options icon for the dash board ( ). ○ Select **Duplicate**. ○ Enter the **New name** and **Description** for the copied dashboard on the modal. ○ Select **Duplicate**. Access the duplicated dashboard by exiting out of editing mode and selecting the new dashboard from the dashboard name drop-down menu. 

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Description
Add tabs to the workspace and edit them as needed	**a.** Add a tab to the home page by selecting **Add a tab**. When you add your first tab to a home page dashboard, a new part of the layout ap pears below the title. Elements you add to the tab are visible no matter which tab the user is on. A tab labeled **untitled** is added next to any existing tabs. **b.** Rename a tab by selecting the tab name, selecting the edit icon ( ), typing the new name, and pressing **Enter**. **c.** Reorder a tab by selecting the tab and drag ging it to the new position. **d.** Delete a tab by selecting its delete icon ( ) then selecting **Delete**. **Note:** No confirmation message ap pears. The tab disappears from the dashboard.
Preview a workspace and make in-line edits	Make basic edits to a workspace when pre viewing it using the In-line editor. **a.** In the Workspace Builder header, select **Preview**. **b.** In the preview that appears in a new brows er tab, select **Edit**. **c.** Adjust the home page elements and wid gets by resizing, reorganizing, or reconfigur ing them. For more information, see Edit Platform Ana **d.** Select **Add new element** to add items to your workspace home page. **e.** Select **Exit Editing Mode** to stop editing in the In-line editor.
Open a printer-friendly version of the dash board	Select the more options icon for the dash board ( ) and select **Printer friendly** to dis play the workspace home page in a format suitable for printing.
Add a bookmark to Analytics Overview	Select the more options icon for the dash board ( ) and select **Add to bookmarks** to include a bookmark for the workspace in the Analytics Overview Bookmarks section.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Description
Make additional or more advanced configurations to the home page	In the configuration panel, select **Open in UI Builder**.
Delete the dashboard	Select the more options icon for the dashboard ( ) and select **Delete**. You must select **Delete** again to conform. **Note:** Workspaces should have at least one dashboard home page.

5. Share the dashboard.

   a. Select the more options icon for the dashboard ( ) and select **Share**.

   b. On the modal, fill in the fields.

   Share dashboard options

Option	Description
Grant access to	Users and groups that can access the dashboard.
Allow recipients to add, edit, or delete sharing permissions associated with this dashboard	Option that enables the users and groups you share the dashboard with to add, edit, and delete the sharing rights for any user, group, or role.
Add as viewer / Add as editor	Option to specify read-only or write access for the dashboard.
Manage access	Roles that can access the dashboard. Search for and select a role in the search box. For each role with access, select the access level: ▪ **Can edit** ▪ **Can view** ▪ **Can share**
Copy link with filter	Button that copies the dashboard URL for the specified users and roles with the currently applied filters.
Copy link	Button that copies the dashboard URL for the specified users and roles without any filters.

   c. Select **Confirm**.

6. In the Workspace Builder header, select **Save**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add a record page for a workspace in Workspace Builder

Configure a record page for a workspace in Workspace Builder. Use containers and components to guide a user through an experience. Each record page is linked to only one data table.

Before you begin

Role required:

- admin or Guided Application Creator for ServiceNow Studio.
- sn_app_eng_studio.user or delegated_developer for AES. For more information, see Delegate developers using AES.

About this task

Components are the building blocks of your page. Add components to your page to build or customize your workspace experience. For example, add a button component to your page that lets users submit requests.

A container is a type of component that can contain other components. Containers have layout properties applied to them. Change the page layout and visual styling to make the record page a unique experience.

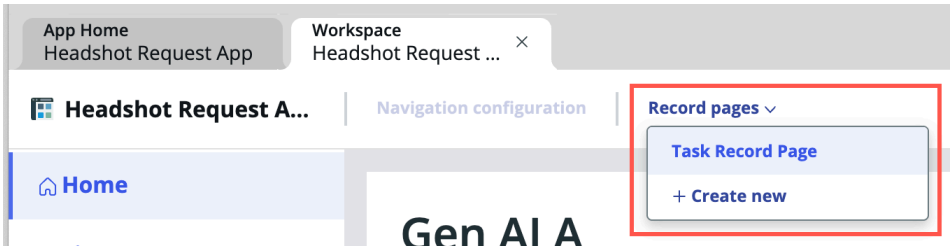
Note: Starting with workspaces built after the AES Workspace UI Template v22.0.3 release, you can work with record pages in Workspace Builder. For older workspaces, you must edit the record pages in UI Builder.

If you don't see the record page for a table you configured for the workspace, see this Knowledge article for more information.

Procedure

1. Navigate to Workspace Builder.
   For details on how to open Workspace Builder in ServiceNow Studio or AES, see Accessing Workspace Builder.
2. Create a record page.
   AES automatically generates record pages for the tables that you selected when you created the workspace.

   a. In the Workspace Builder header, from the **Record pages** menu, select **+ Create new**.



   b. On the form, fill in the fields.

Record page settings

Field	Description
Name	Brief name that identifies the purpose of the record page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

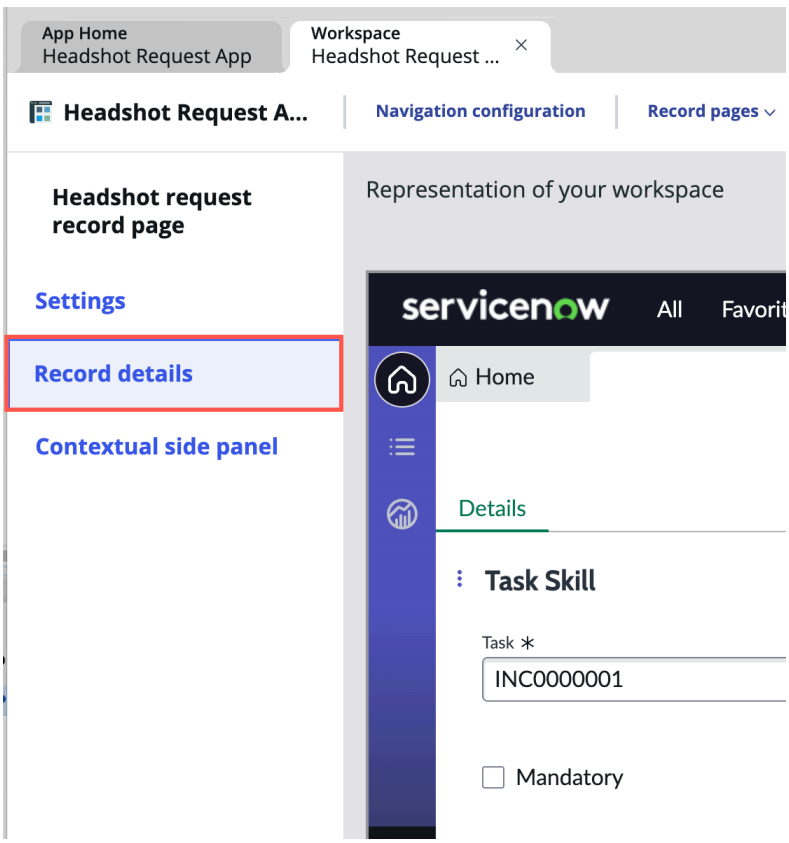
Field	Description
Table	Table that the record page is linked to and updates. The following rules apply: ▪ If you're using a table that you or someone at your organization created, the table must contain at least one new field or column. Otherwise, the form view and the record for the table can't be created. ▪ You can link each table to only one record page. You can select from two sections: Tables that are already available in your app, and all tables outside the app scope.

c. Select **Create**.

The page appears in the preview panel. You can select the page from the **Record pages** menu.

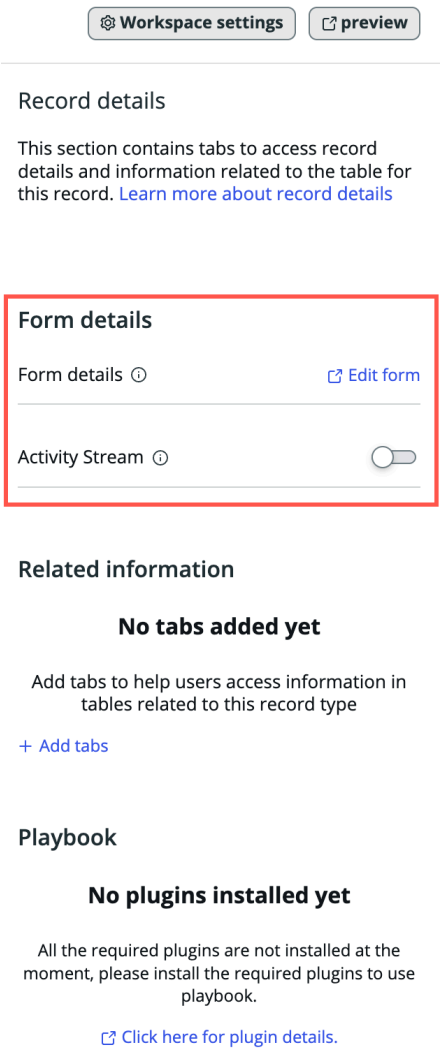
3. Edit the record details for the record page.

a. In the navigation panel, select **Record details**.



b. In the **Form details** section of the configuration panel, select **Edit form** and fill in the fields.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Form details for record page

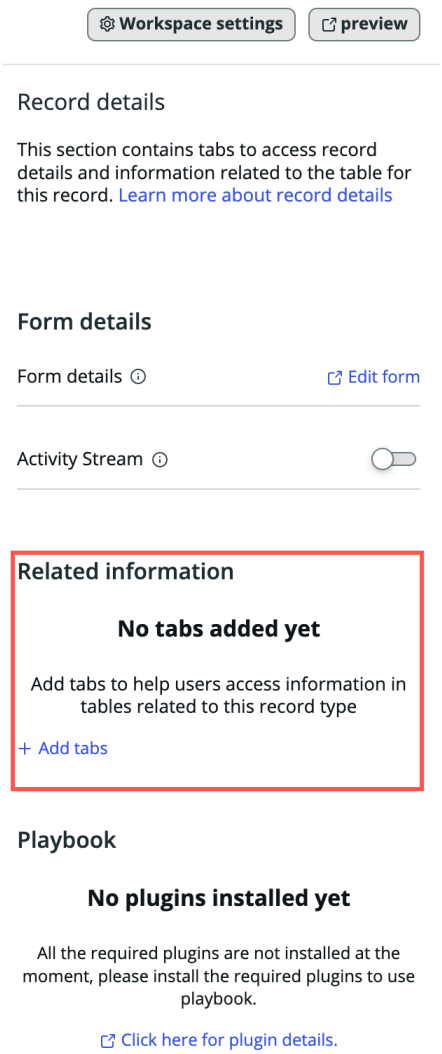
Field	Description
Form details	Select the **Edit form** link to edit the form in a Table Builder tab inside AES. **Note:** After you leave the record page in the Table Builder tab, a dialog prompts you to **Reload this page**. For more information on editing forms for a record page, see Forms in Table Builder.
Activity stream	Option to enable the Activity stream for the record page. The Activity stream enables your workspace users to communicate with requesters and make internal notes about the work done on a record. For more information on the Activity stream, see Activity streams.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

4. Configure the related information in the Related information section of the configuration panel for the record page.
   Related information includes tables that are related to the record, which appear as tabs, such as Child Incidents. For more information on configuring related information, see Related lists.

   Note: You can't create a table for the related information in a record page. The table must exist in ServiceNow AI Platform.

   a. If no related lists exist yet, add a new related list by selecting **+ Add tabs** in the **Releated information** section.



   If related lists exist for the record page, select the **Manage related information** link in the configuration panel.

   b. In the form that appears in a new tab within AES, move the related list that you want to appear as a tab from the **Available** panel to the **Selected** panel.

   c. Use the arrow buttons to move the selected lists into the order you want them to appear.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

d. Select **Save** and return to the Workspace Builder tab within AES.

e. View the changes that you made by selecting **Reload this page** in the dialog box that appears.

Each related information component has an information icon (ⓘ) that, when selected, displays its related table.

5. Provide fulfillers with visibility into cross-business workflows and the actionable tasks used to complete these workflows by specifying the playbook.

> **Note:** You must have the Playbook Experience plugin installed. If you used a template to create the app, you can't add a playbook to the record page. For details on how to add a playbook for workspaces created from templates, see this knowledgebase article.

a. In the Playbook section of the configuration panel, select **Add a playbook**.

b. On the form, fill in the fields.

Playbook details

Field	Description
Tab name	Label for the record tab that contains the playbook.
Playbook experience	Existing playbook to add to the record. **Note:** You can choose only from the playbooks that are available for the table.

c. Select **Add**.

d. Update the playbook by selecting the settings icon (⚙), making the change, and then selecting **Update**.

> **Note:** If you want to make additional changes to the playbook, you must open and edit the playbook in Playbooks.

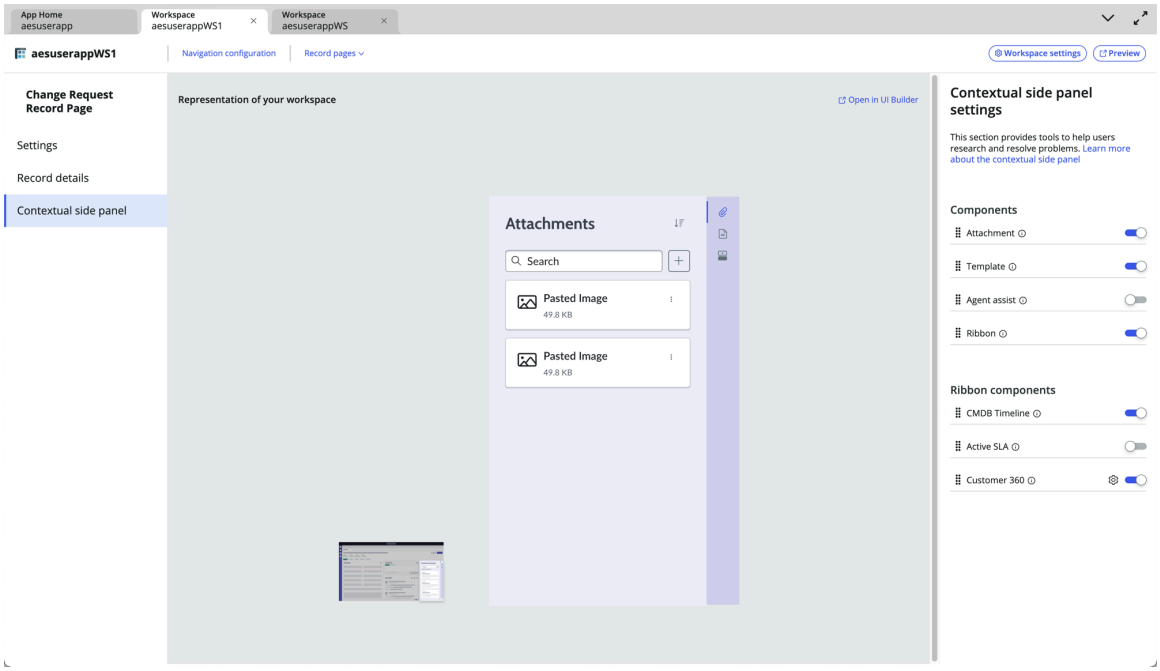
e. Delete a playbook from the workspace by selecting its delete icon (🗑) and selecting **Delete** on the confirmation dialog.

6. Configure the contextual side panel for the record page, which contains the icons for the tools that appear on the right side of the page.
The contextual side panel contains icons that enable agents to research and resolve issues using various ServiceNow AI Platform tools. If your organization created any custom right tab components in UI Builder, you can add them as well..

> **Note:** If you used a template to create the app, you can't add a contextual side panel to the record page.

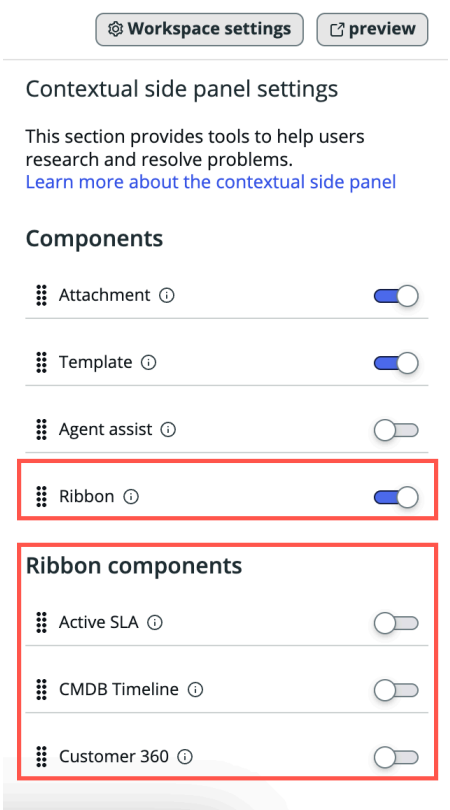
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. In the navigation panel for the record page, select **Contextual side panel**.



b. In the **Context side panel settings**, select the toggle switch for each component and ribbon component that you want to appear on the record page.
For example, you can include Agent assist and attachments for the record. Attachments and templates are enabled by default, though you can disable them by deselecting their toggle options.

To see the ribbon components, set the **Ribbon** component toggle to on.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

To add pre-existing custom components, edit the workspace in UI Builder.

Contextual side panel components

Tool	Type	Description	More information
Active SLA	Ribbon component	Display active Service License Agreements (SLAs) for the case, including time remaining, the SLA state, and any breaches.	SLA ribbon
Attachment	Component	Add attachments to the record, such as images and PDFs.	Attachments
Configuration Management Database (CMDB) Timeline	Ribbon component	Display a chronological summary of case activities, including state changes and interactions between the agent and the requester (usually the customer). The ribbon also displays how much time that the fulfiller and requester spent on the case.	Timeline ribbon component
Customer 360	Ribbon component	Display customer information, such as the contact or consumer name, email address, and phone numbers.	Customer 360 ribbon component
Ribbon	Component	Display the contextual side panel ribbon, which contains tools to help agents resolve issues. Ribbon components can have additional components added to them.	Viewing ribbon information in the contextual side panel

The selected tools appear in the record page preview canvas.

c. In the configuration panel, drag the components to rearrange them.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

d. **Optional:** Complete any additional configuration for components by selecting the settings icon (⚙).
For example, Customer 360 requires that you specify additional settings before the component works on the record page.

Workspace Builder automatically saves the changes to the contextual side panel.

7. Complete any additional changes and configurations.

Option	Description
Update the name of the record page	**Note:** You can't change the table that a record page is linked to after you cre ate the page. To change the table for the record page, you should create a new, separate record page. **a.** From the Record pages menu, select the page that you want to edit. A preview of the record page appears. **b.** In the navigation panel, select **Settings**. **c.** In the Record page settings configuration panel that appears, enter the new page name. **d.** Select **Save**.
Preview the record page in a new browser tab	Select **Preview** in the Workspace Builder header. The record page appears in the workspace. If no data exists for the table, AES displays a page where you can add data to the table.
Delete a record page	**a.** From the Record pages list, select the page. **b.** In the navigation panel, select **Settings**. **c.** At the bottom of the configuration panel, se lect **Delete record page**. **d.** On the confirmation dialog, select **Delete**.

Create lists for a workspace in Workspace Builder

Create list categories in Workspace Builder to add pages that list table records. You can add filter conditions and change columns to create variations on the list.

Before you begin

Role required:

- admin or Guided Application Creator for ServiceNow Studio.
- sn_app_eng_studio.user or delegated_developer for AES. For more information, see Delegate developers using AES.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

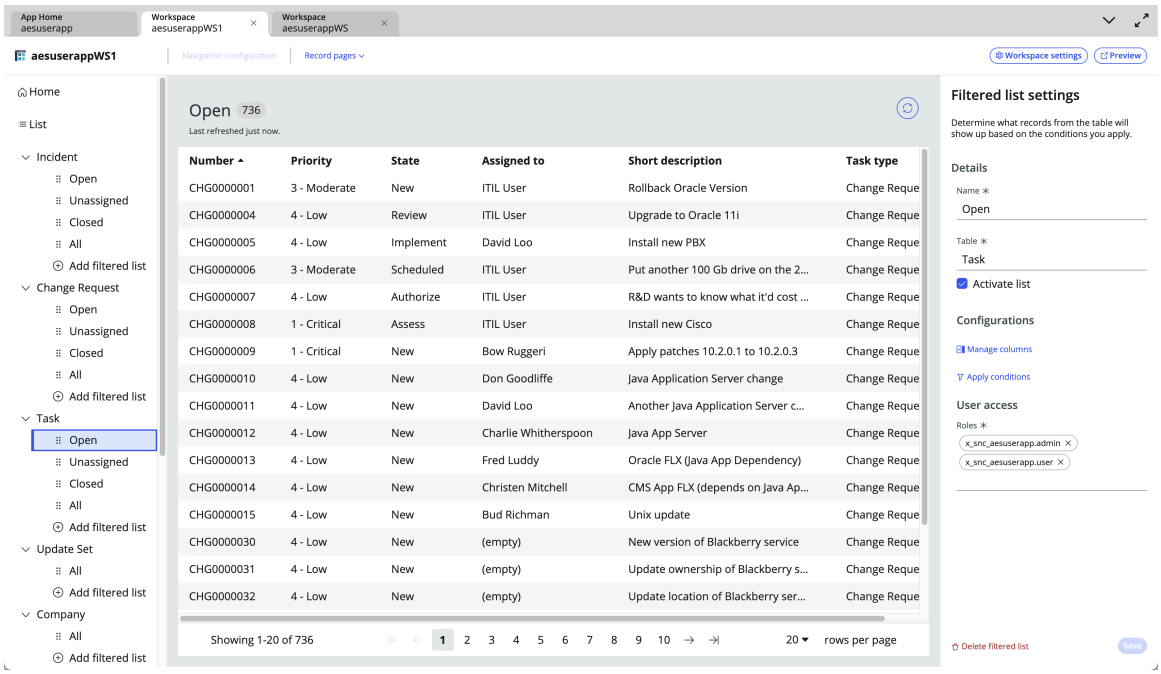
About this task

Note:

- You can launch and view workspaces in Workspace Builder as well. However, you can only edit workspaces created through ServiceNow Studio and App Engine Studio within those specific platforms only.
- If you remove the list page/route, you must use UI Builder to edit the workspace, not Workspace Builder.

Procedure

1. Navigate to Workspace Builder.
   For details on how to open Workspace Builder in ServiceNow Studio or AES, see Accessing Workspace Builder.
2. If they aren't already activated, activate lists to enable them for the workspace.
   You must be in the **Navigation configuration** tab. If you're already in the **Navigation configuration** tab, you can't select it.
   a. In the navigation panel, select **List**.
   b. In the list, select **Task** > **Open**.
   c. In the configuration panel, select the **Activate list** option.



3. Add a list category.
   a. In the navigation panel, select **Add list category**.
      If the table associated with the workspace isn't a core table, you must add new list categories.
   b. On the form, fill in the fields.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

List category settings

Field	Definition
Name	Name to identify the list category.
Description	Brief description of the contents of the list category to help developers understand the purpose. **Note:** Workspace users don't see the description.

c. Select **Add**.

4. **Optional:** To rearrange the order in which list categories appear, drag them to a new position.
5. Add a filtered list to a list category.
   For example, in the list category for the Task table, you could create separate filtered lists for Opened, Unassigned, and Closed tasks.

   a. In the navigation panel, expand the list category you want to create a filtered list for.

   b. Select **Add filtered list**.

   c. On the form, fill in the fields.

Filtered list settings

Field	Definition
Name	Name to identify the filtered list.
Table	Table that contains the subset of records that appear in the filtered list.You can select from two sections: Tables that are already available in your app, and all tables outside the app scope.

   d. Select **Add**.

   The preview canvas displays the contents of the filtered list.
6. Refine how the filtered list displays its content by adding filter conditions to narrow what it displays.
   You can make additional configurations to the filtered list.

   a. In the navigation panel, select the filtered list to edit.

   b. In the configuration panel, specify the additional settings.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

More filtered list settings

Field	Definition
Activate list	Option to enable users to work with the filtered list.
Manage columns	Option to select and arrange the columns that appear. In the Manage columns dialog box, move columns from **Available columns** to **Selected columns**, drag to rearrange the columns, and select **Apply**. You can select a field on a related table by dot-walking to it. For more information, see Dot-walking to data in related tables.
Apply conditions	Option to add a filter condition to the list. In the dialog box that appears, you can do the following: ▪ To apply a predefined filter, select **Use existing filter** and search for the filter. ▪ To view existing conditions, expand the **Filter overview** section. ▪ To create a filter condition, use the condition builder. For more information on building conditions, see Condition builder. ▪ To change the sort order of the filtered results, expand the **Sort by** section and make changes. ▪ To save the filter condition that you created for reuse, select **Save filter**. When you're done creating the filter condition, select **Apply filter**. For more information on working with filters for lists, see Filters.

c. Select **Apply filter**.

d. In the configuration panel, select **Save**.

7. Complete any additional actions on the list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Description
Edit a list category	**a.** In the navigation panel, select the list category. **b.** In the configuration panel, specify its **List category settings**. **c.** Select **Save**.
Change how the filtered list appears by selecting the list in the navigation panel	**a.** In the configuration panel, select **Manage columns**. **b.** In the form that appears, move columns from **Available columns** to **Selected columns**, drag to rearrange the columns, and select **OK**. **c.** Refresh the canvas preview by selecting the refresh icon ( ) in the canvas header.
Delete a list category	**a.** In the navigation panel, select the list category. **b.** In the configuration panel, select **Delete list category**. **c.** In the confirmation dialog, select **Delete**.
Delete a filtered list	**a.** In the navigation panel, select the filtered list. **b.** In the configuration panel, select **Delete filtered list**. **c.** In the confirmation dialog, select **Delete**.

Enable analytics for a workspace in Workspace Builder

Enable the Analytics Overview for a workspace in Workspace Builder. The Analytics Overview enables users to track and analyze records and usage with dashboards, data visualizations, and insights on the instance.

Before you begin

Role required:

- admin or Guided Application Creator for ServiceNow Studio.
- sn_app_eng_studio.user or delegated_developer for AES. For more information, see Delegate developers using AES.

About this task

If Analytics Overview page/route doesn't exist for a workspace, Workspace Builder is still available, but the Analytics Overview tab doesn't appear in the workspace.

The Analytics Overview always displays current, refreshed content.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to Workspace Builder.
   For details on how to open Workspace Builder in ServiceNow Studio or AES, see Accessing Workspace Builder.
2. In the navigation panel, select **Analytics**.
   You must be in the **Navigation configuration** tab. If you're already in the **Navigation configuration** tab, you can't select it.
   A mock-up of the Analytics Overview appears in the preview panel.
3. In the configuration panel, select the **Activate Analytics Center** toggle option.
4. Make additional configuration updates to the Analytics Overview by selecting the **Open in UI Builder** link in the canvas header.

Result

For more information on working with the Analytics Overview, see Dashboards in Platform Analytics.

Edit a workspace in Workspace Builder

Edit a workspace in Workspace Builder to make customizations.

This video shows you how to perform the following procedure.

Before you begin

Before you can configure a workspace in Workspace Builder, you must first create the workspace. See Add a workspace.

Role required:

- admin or Guided Application Creator for ServiceNow Studio.
- sn_app_eng_studio.user or delegated_developer for AES. For more information, see Delegate developers using AES.

About this task

You can customize essential elements and components in Workspace Builder. For more complex configurations, or if you don't have the full entitlement for AES, you must edit the workspace in UI Builder.

Procedure

1. Navigate to Workspace Builder.
   For details on how to open Workspace Builder in ServiceNow Studio or AES, see Accessing Workspace Builder.

   The Workspace Builder tab displays your workspace within AES:

   - The left navigation panel is a list of pages or components, or building blocks of a page, that you can include in your workspace.

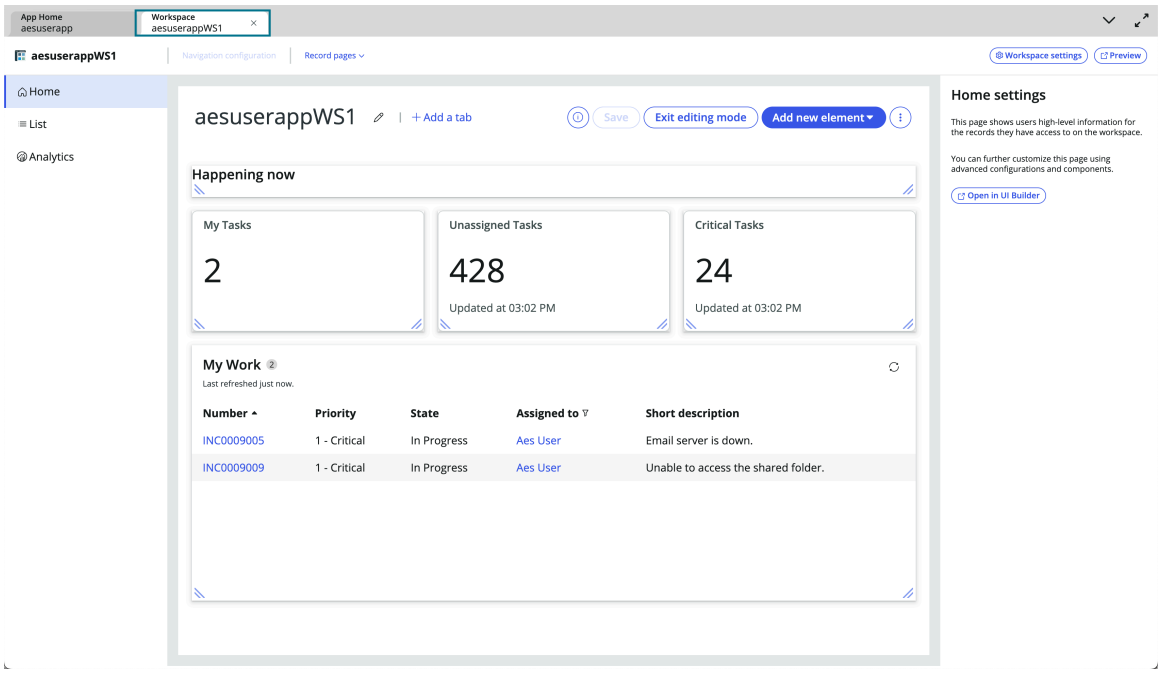
     Note: The home page is an essential part of the workspace, and thus can't be hidden or removed.

   - The middle canvas is an in-line editing space for the home page for workspaces created in AES after Tokyo. For lists and other pages, the middle canvas is a preview of what you

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

configure in the right configuration panel. You can preview workspace home pages built before Tokyo in Workspace Builder, but you must edit them in UI Builder.

- The right configuration panel is the configuration panel for working with selected components.



If you created the workspace that contains a technical dashboard, Workspace Builder prompts you to **Open in UI Builder** when you try to edit the workspace.

2. Edit the basic settings for the workspace by selecting **Workspace settings**.
   For more information, see Configure workspace settings in Workspace Builder.
3. Adjust the home page elements and widgets by resizing, reorganizing, or reconfiguring them.

   For more information on editing a clickable home page, see Customize a workspace home page in Workspace Builder.

   Note: If the workspace was created before Tokyo, you must edit it in UI Builder. See UI Builder for more information.

4. Add or edit a list category and any subsequent filtered lists.
   For more information, see Create lists for a workspace in Workspace Builder.
5. **Optional:** Enable an Analytics Overview for the workspace.
   For more information, see Enable analytics for a workspace in Workspace Builder.
6. Add and edit a record page to configure the settings, tables, and related links that the workspace should support.

   When you create a record page, you're creating the metadata view, or shell, using a record page template for a type of record. You then edit the record page in Table Builder. Within Workspace Builder, you can only change the contextual side panel and the related items.

   For more information, see Add a record page for a workspace in Workspace Builder.
7. Preview the workspace in a new browser tab by clicking **Preview**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Result

If you made additional, more complex configurations to the workspace in UI Builder, such as custom additions to the contextual side panel or custom components, those changes may not appear in Workspace Builder.

Note: Your workspace must have a home page and lists for you to edit it in Workspace Builder. If they aren't present, you must edit the workspace in UI Builder.

Workspace Builder reference

Reference topics provide additional information about the lists and forms that you use to configure and administer Workspace Builder.

Workspace Builder components for home pages

Several UI Builder components are available as elements to build home pages and contextual side panels in Workspace Builder. Workspace Builder helps you quickly create workspaces.

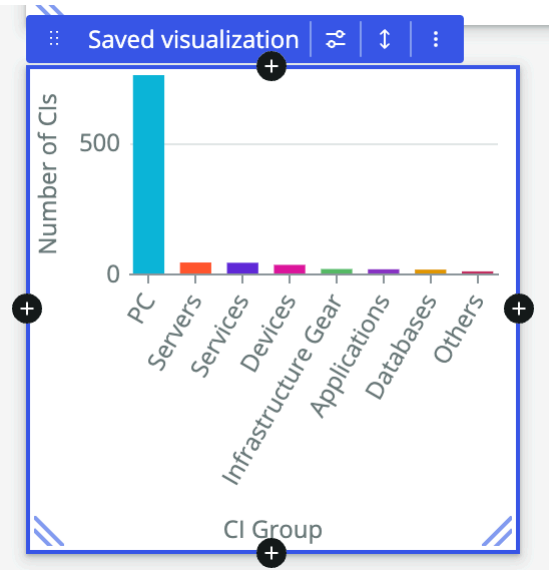
For complete details on all elements and components, see the ServiceNow® Developer site.

Data visualization components

The data visualization component is a container for available chart types for use in Workspace Builder home pages. Select the chart type to display data in different ways depending on your application. You can create a new data visualization or repurpose an existing data visualization by choosing it in the Select visualization from library modal and selecting **Add to dashboard**.

For more information on data visualization components, see the following topics and sources:

- Data visualizations in Platform Analytics
- Create a single score data visualization
- Developer site data visualization component listing



Filter components

Filters enable you to filter data visualizations without modifying the visualizations. Add a filter element when working with the home page in the canvas. Then select a table, field, and other properties to configure the filter. When a user selects one or more values of the field, the filter applies to those data visualizations that are based on the same data source.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

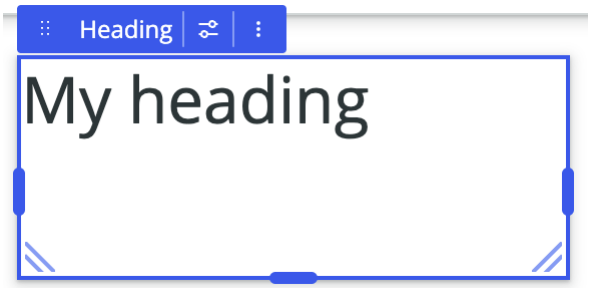
For more information on using filters components to build a workspace, see the following topics and sources:

- Next Experience filters
- Developer site filter component listing

Heading components

Heading components are available in the home pages for workspaces. Add a title to your landing page using a heading component.

The following image shows a "My Heading" heading.

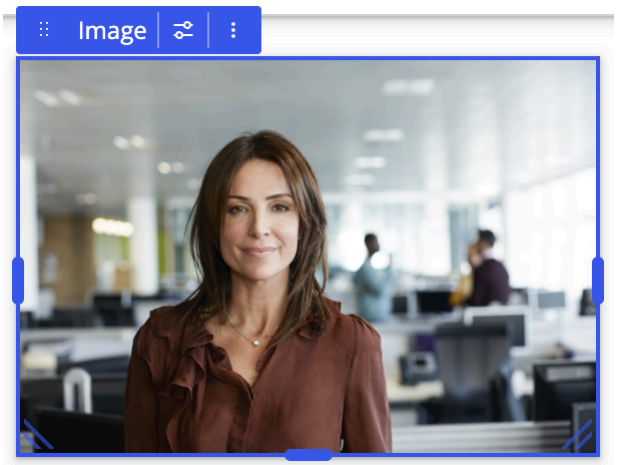


For more information on heading components, see the Heading Overview on the Developer site.

Image components

Add images to your landing page with an image component.

When you drag the image component icon (Image) into the preview pane, a stock image appears that you can change and resize. You can change the stock image by supplying your image's URL.



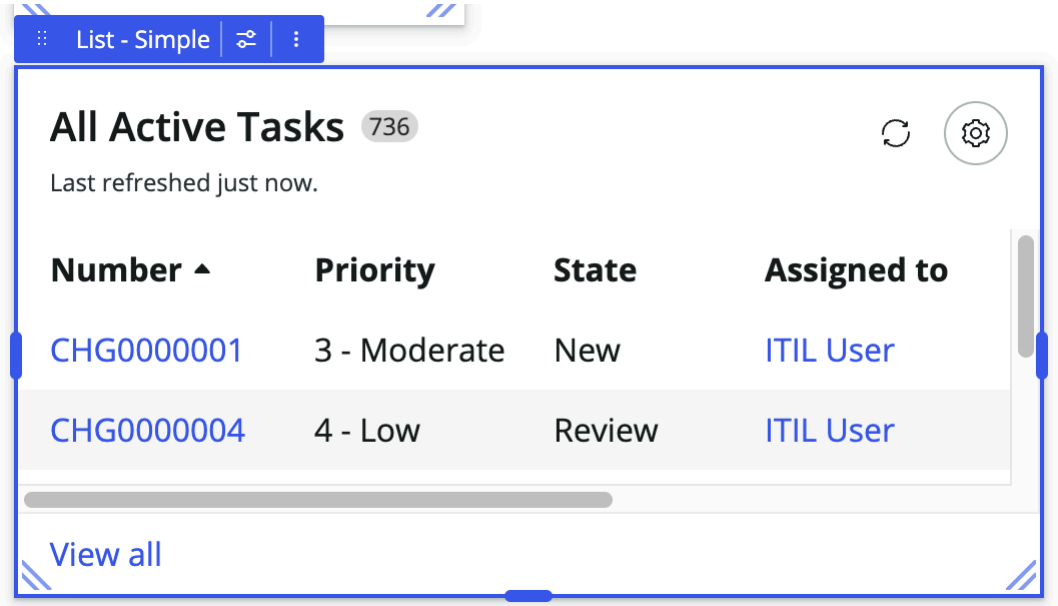
For more information on image components, see the Developer site image component listing.

List components

Use a list component to add lists to your landing page.

When you drag the list component icon (List) into the preview pane, a sample list appears.

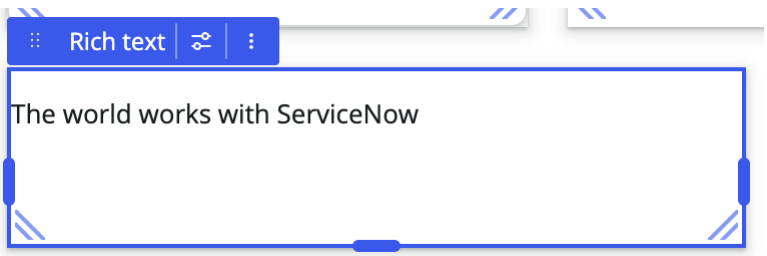
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



For more information on list components, see the Developer site List component reference.

Rich text components

Add text to your landing page with a rich text component to explain parts of your landing page, for example, "Tackle these priority 1 incidents first."



For more information on rich text components, see the Developer site Rich text component reference.

UI Builder

UI Builder is a web user interface builder. Use UI Builder to build pages for App Engine Studio generated workspaces or custom web experiences using Next Experience Components and custom web components.

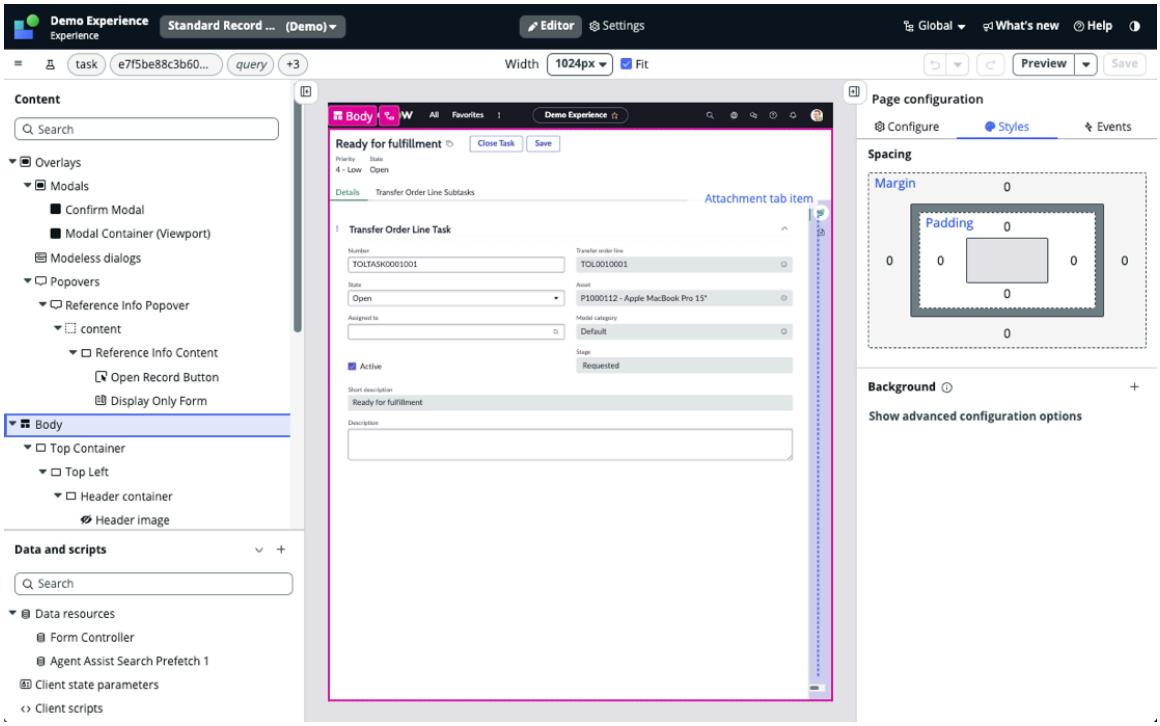
UI Builder overview

Create or customize pages for workspace experiences. A page is a collection of components that make up a workspace user interface.

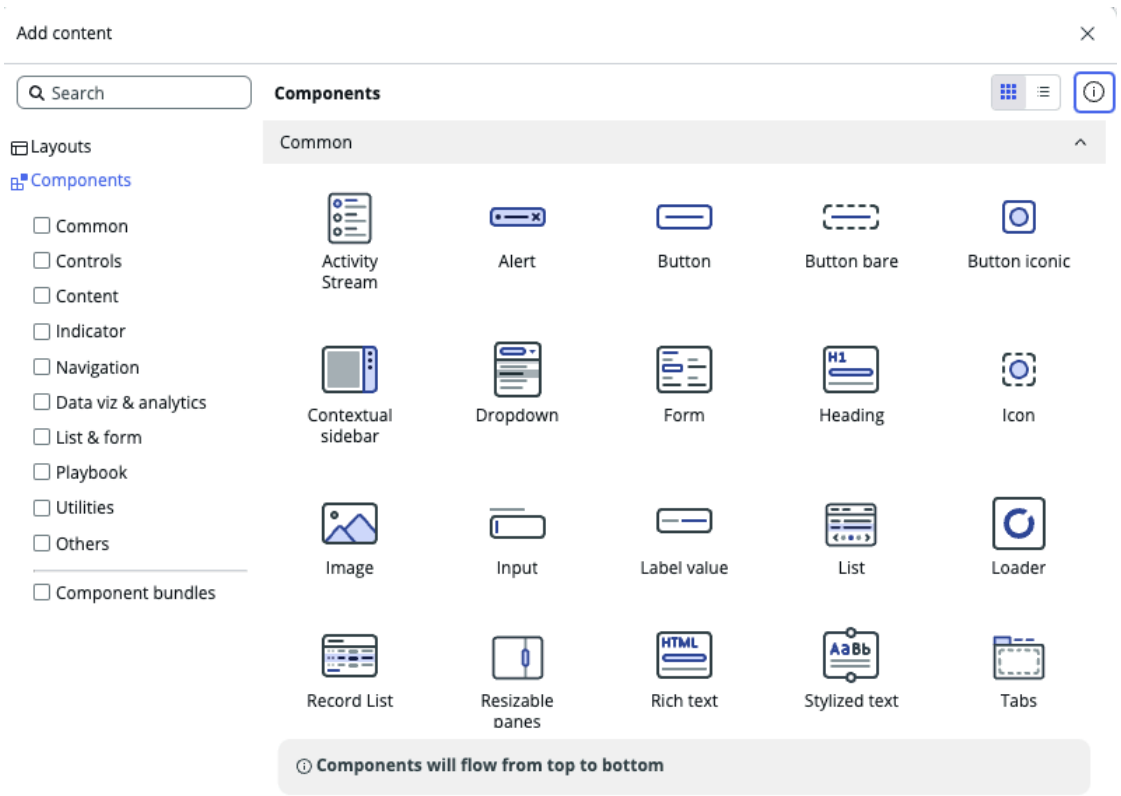
Create variants of pages to target experiences for different audiences. For example, you can create a home page for agents, and a variant for managers at the same URL.

Note: The ui_builder_admin role is required to complete tasks in UI Builder.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

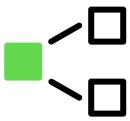


Use the UI Builder library of components to build your pages. Configure them any way that you need and then connect your organization's data to the components.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Get started with UI Builder

Explore	Learn
	
Explore UI Builder concepts and features.	Learn the basics of using UI Builder.
Work	**Advanced**
	
Dive deeper into configuring UI Builder experiences and pages.	Learn how to add data and make your pages dynamic.

Note: UI Builder isn't yet capable of building or configuring ServiceNow base system service portals, such as the Employee Center. For service portals, continue to use the Service Portal Designer.

Troubleshoot and get help with UI Builder

- Ask or answer questions in the UI Builder community
- Search the Known Error Portal for known error articles
- Learn more about how to create your own apps on the developer site.
- Contact Customer Service and Support

Exploring UI Builder

Explore facets of UI Builder to set you up for creating pages quickly.

UI Builder is a web user interface builder. Use UI Builder to build pages for CRM Workspace, App Engine Studio generated workspaces and portals, or custom web experiences using Next Experience Components and custom web components.

Check out the UI Builder quick start topic to understand the basics of UI Builder. If you want a deeper understanding of UI Builder, go through the tutorial to learn how to create your first page.

Find information relating to UI Builder audiences, security and roles settings like application scope and domain separation, as well as experience settings.

See how you can use UI Builder with CSM Workspace and portal experiences.

UI Builder quick start

This quick start guides you through the process of creating your first page in UI Builder. Creating your first page is the first step in understanding how to build user interface pages for your workspace or custom portal experiences.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

Role required: ui_builder_admin

In this UI Builder quick start, you perform the following tasks to build your first page in UI Builder:

- Start UI Builder.
- Create a page for your workspace or custom portal experience. For more information about creating pages, see Create a page in UI Builder.
- Build your page by adding components. For more information about components, see Customize UI Builder pages using components.
- Save your page.
- Preview your page to see how it looks in a browser.

Procedure

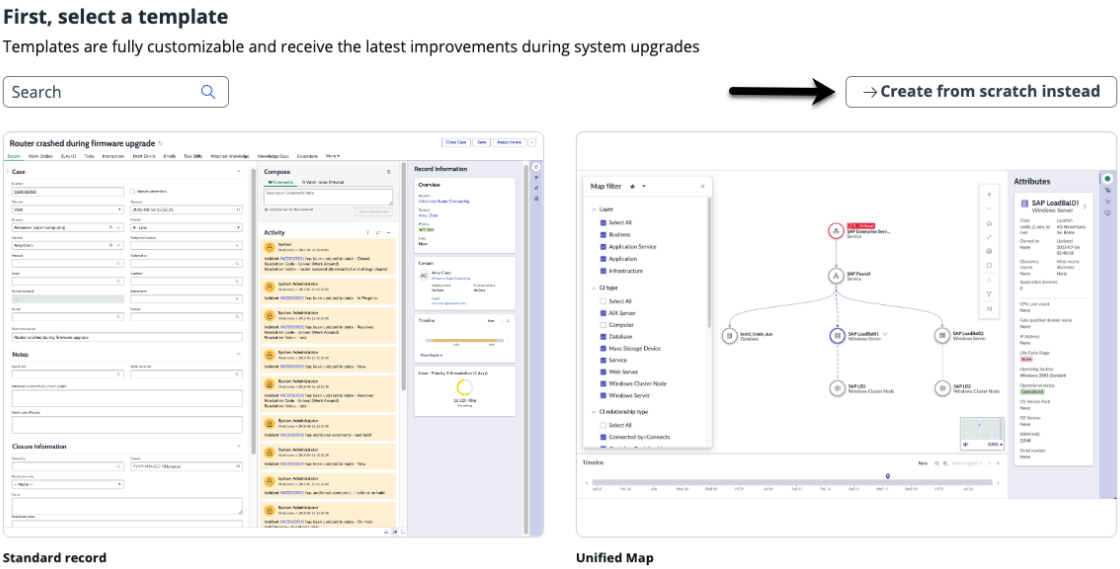
1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Select an experience that you want to work in from the UI Builder home page.
   If you don't see any experiences listed in which to work, contact your administrator to get access to an experience, or create an experience. For more information, see Configure how users interact with your applications in UI Builder.
3. Create a page.

   a. Select the **+** icon in the **Pages** section.

   b. Select **Create a new page**.

   c. Select **→ Create from scratch instead**.

   You can also create pages using page templates, see Create a page from a template for more information.

   d. Enter `Start Page` as the unique name for the page in the **Name** field.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

e. Specify a path for your page in the **URL Path** field. UI Builder generates a default path based on the name that you gave in the last step.

   A default path is added based on your page name. You can also create your own path. The path is required and must be unique. The path can include digits (0-9), letters (A-Z, a-z), and a few special characters (`" - "`, `" . "`, `" _ "`, `" ~ "`), with the words separated by a forward slash or hyphen. The **URL preview** shows the path of your page.

   Note: The application scope defaults to the scope that the user is currently in within the ServiceNow AI Platform®. For more information about the application scope, see Learn about security and roles.

f. Select a URL page type in the **Type** drop-down list.
   URL page types help you categorize and find important pages in the experience view. Select from **None**, **Dashboard**, **Knowledge base**, **List**, and **Record** page types. Pages assigned a type other than **None** are filtered more effectively in the experience view.

   Note: A landing page is assigned a **Landing** URL type by default.

g. Select **Continue**.

h. **Optional:** Add parameters to your page URL by selecting **+ Add**.
   For more information, see Manage UI Builder pages and page variants.

i. Select **Looks good**.

j. Enter `Manager Start Page` as the name for the page variant.

k. **Optional:** Add one or more audiences for this page.
   If an audience you need isn't listed, you can choose the **Open audiences in the platform** link to create one.

l. **Optional:** Add conditions for when to display the page or tab.

m. Select **Continue**.

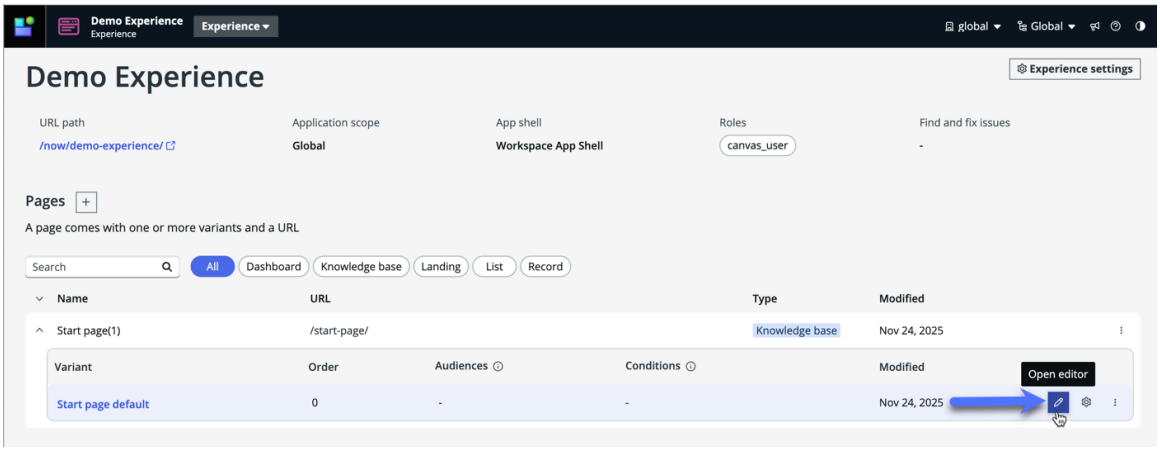
n. On the next screen, select **Build responsive**.

o. Select **Create** to create your blank page.

   Congratulations! You've created your first page! The page is empty of content. You add components to the page to build functionality to it. Components are the building blocks of a page. UI Builder comes with many components ready for you to add to your page. Components can be as simple as a **Heading**, or as complex as a **List**.

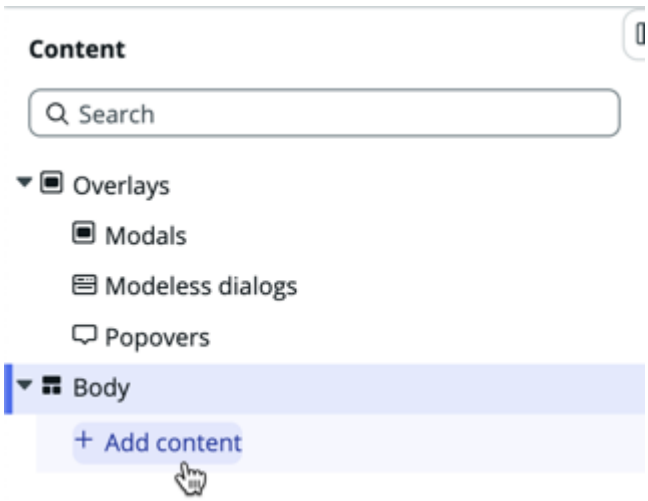
4. Select the pencil icon to start customizing your page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



5. Add a container component to your page.
   A container is what holds your components. Think of a container as an area of the page where you add information, images, or functionality (your components). You can have as many containers on a page as you want, with as many containers within containers, with as many components in the containers.

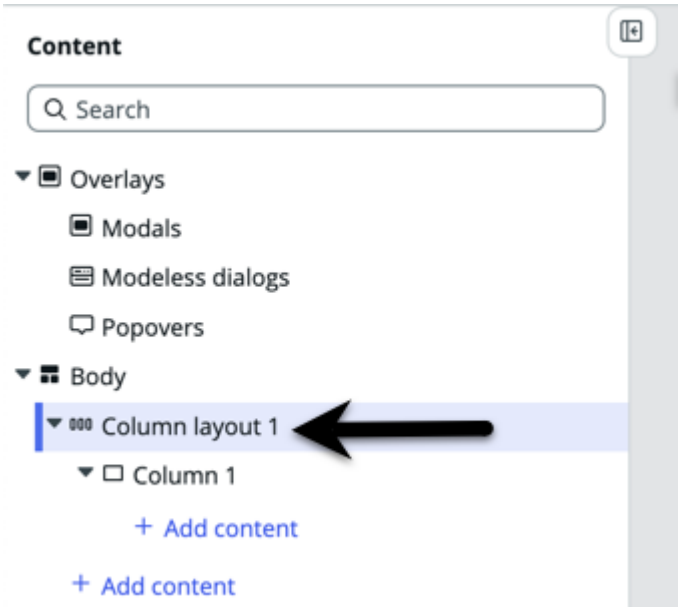
   a. Select **+ Add content** in the content tree.



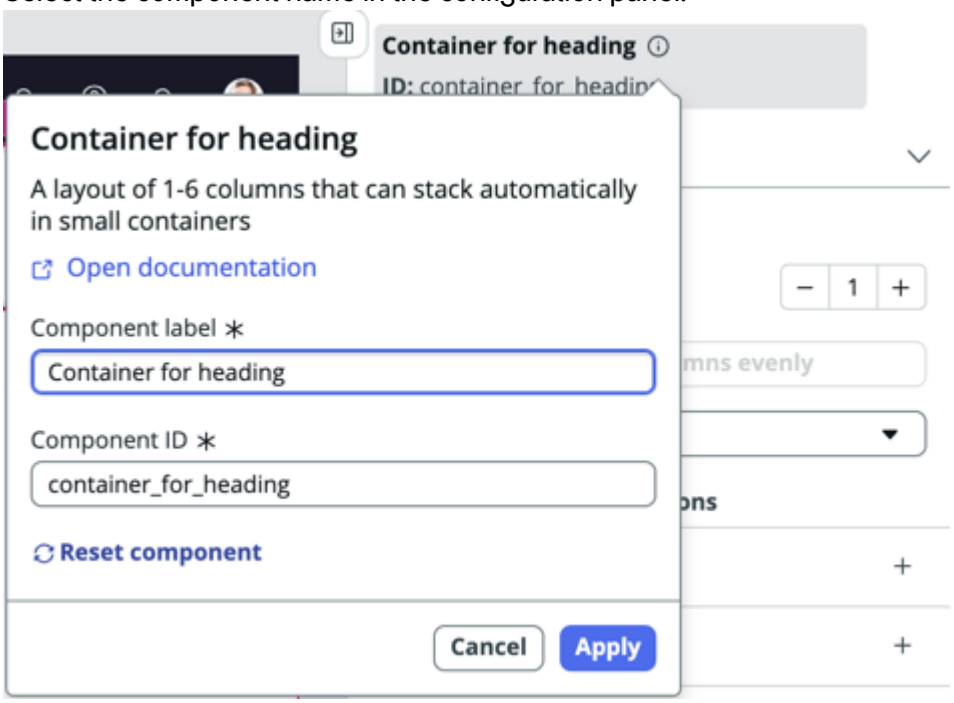
   b. Select the **Single column** layout in the toolbox.

   c. Select **Column layout 1** in the content tree to highlight the container.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



d. Select the component name in the configuration panel.



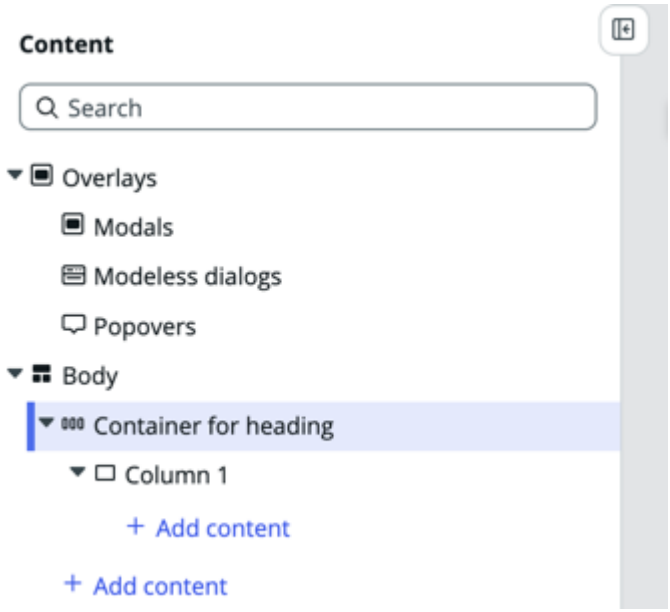
e. In the **Component label** field, type `Container for heading`.

f. In the Component ID field, type `container_for_heading`.

g. Select **Apply**.

See that the column layout name changes to **Container for heading** in the content tree. The content tree is an important concept. The content tree is an easy way to see the structured layout of your page. The content tree is especially important when you have multiple components on your page. You select the component in your content tree to highlight the component on the page, making it easier to build your page. You can do a text search for a component.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

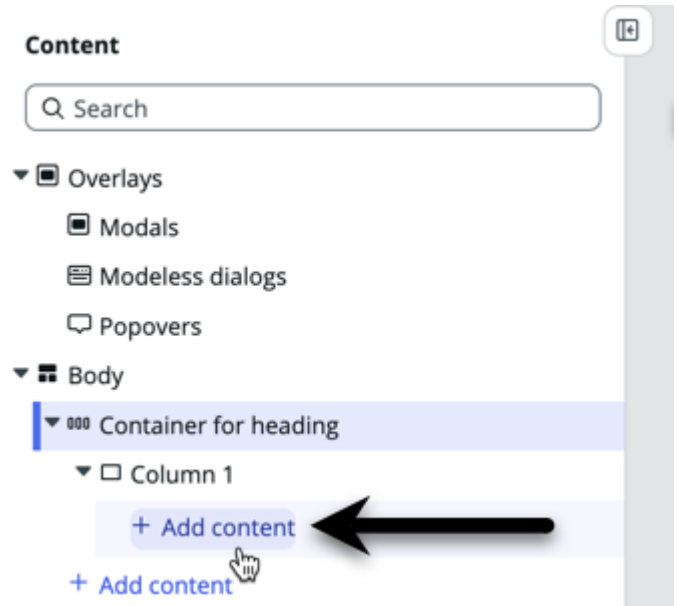


For more information on component IDs, see Component ID.

You've successfully added your first column layout to your page.

6. Add a **Heading** component to your column layout.
   You can add components to the page in different ways. For more information on the ways you can add components to your page, see Add and configure components.

   a. Select **+ Add content** in the content tree below the column layout created in step 5.



   b. Type `heading` into the search and select the **Heading** component.

   c. Select your new **Heading** component in the content tree to highlight it.

   d. Select **None** in the configuration panel.

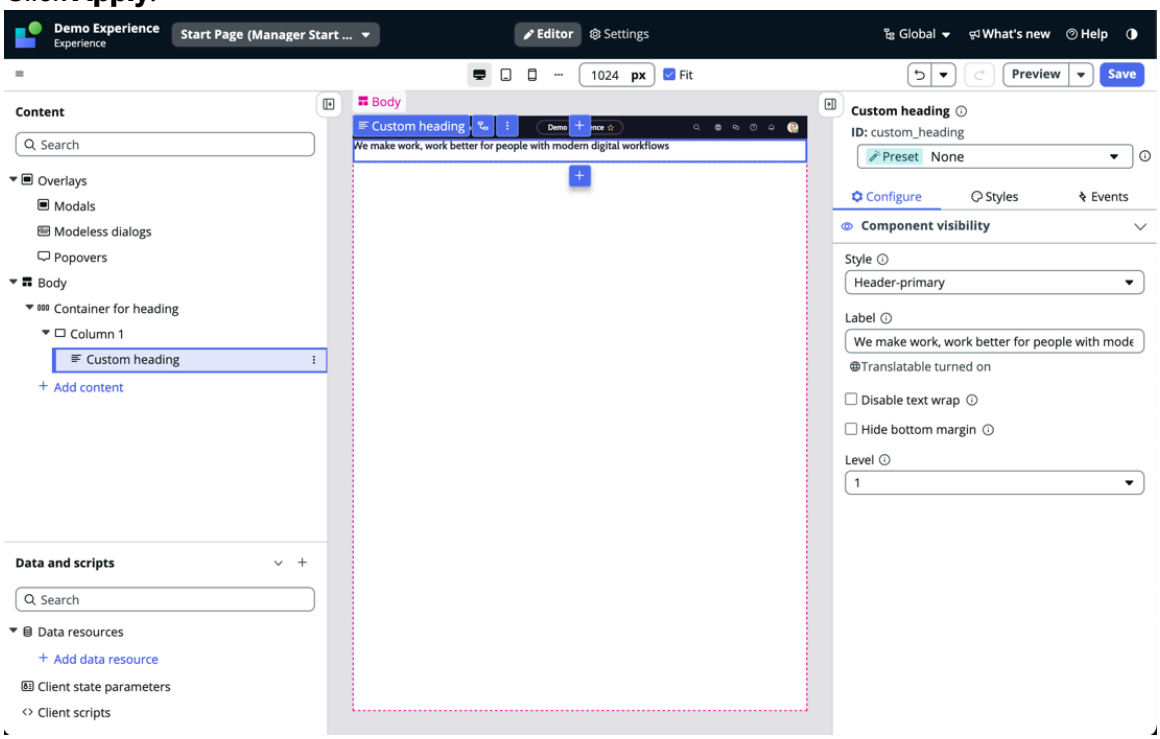
   e. Select the component name in the configuration panel.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

f. In the **Component label** field, enter `Custom heading` as the unique label for the heading component.

g. In the **Component ID** field, enter `custom_heading` as the unique ID for the heading component.

h. Click **Apply**.



7. Customize the heading component.

a. Select the heading component in the content tree.

b. Select the **Configure** tab in the configuration panel.

c. In the **Label** field, enter the text of your heading, such as `My new heading`.

d. The Style changes the size of the heading text.
For example, if you select **Header-secondary**, the text is smaller. Different headings sizes are useful if you have two headings and want the second heading smaller that the primary heading. For more information on configuring components, see Configure components in UI Builder.

e. Leave the **Level** as **1**.

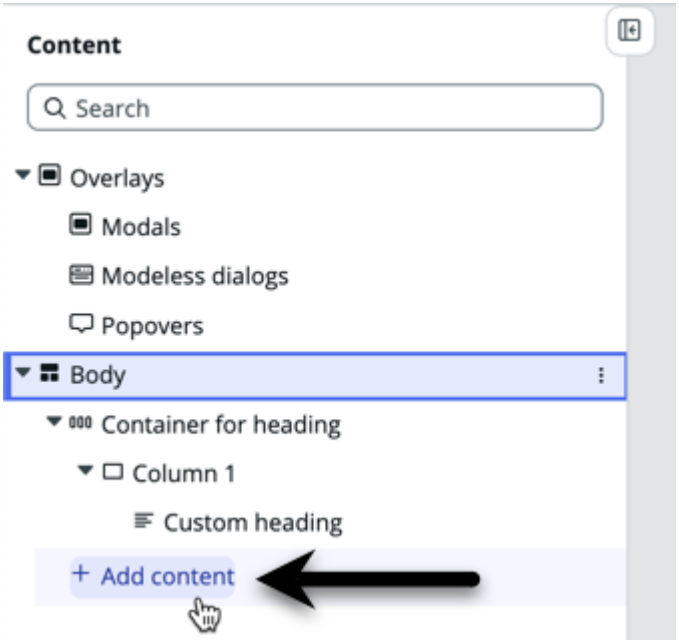
f. Select **Save**.

You've added and customized a heading component to your page.

8. Add a second column layout to your page.

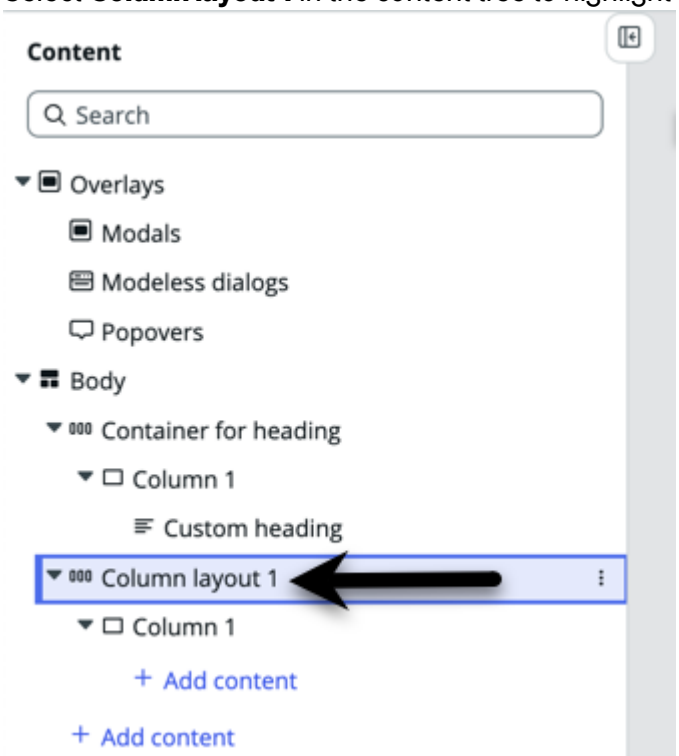
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. Similar to before, select **+ Add content** in the content tree.



b. Select the **Single column** layout in the toolbox.

c. Select **Column layout 1** in the content tree to highlight the container.



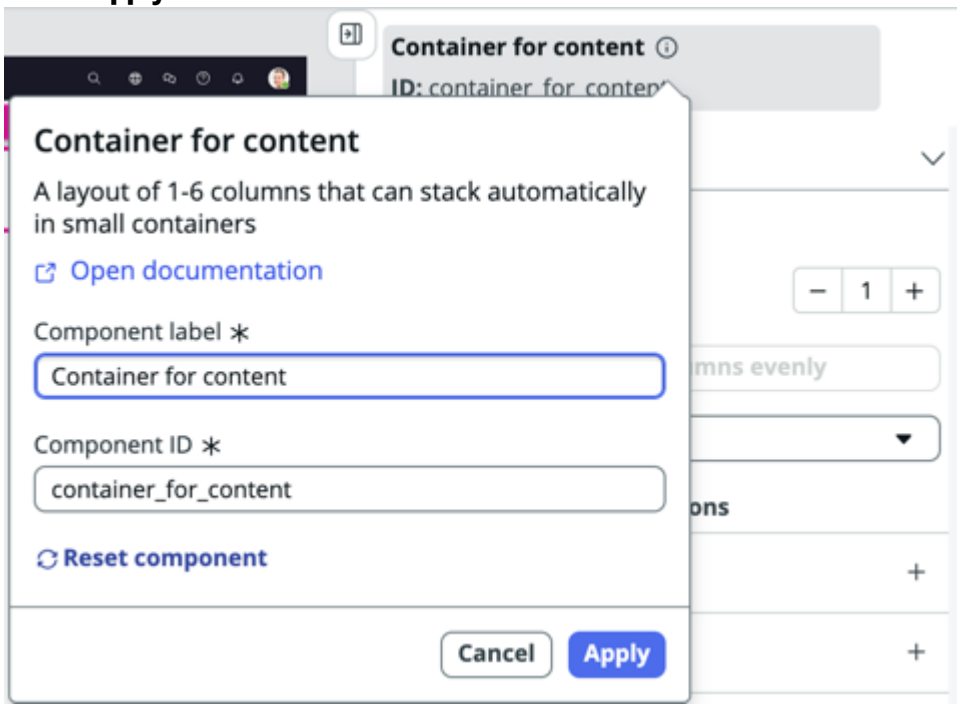
d. Select the component name in the configuration panel.

e. In the **Component label** field, type `Container for content`.

f. In the **Component ID** field, type `container_for_content`.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

g. Select **Apply**.



h. Select **Save**.

You've successfully added your second container component to your page.

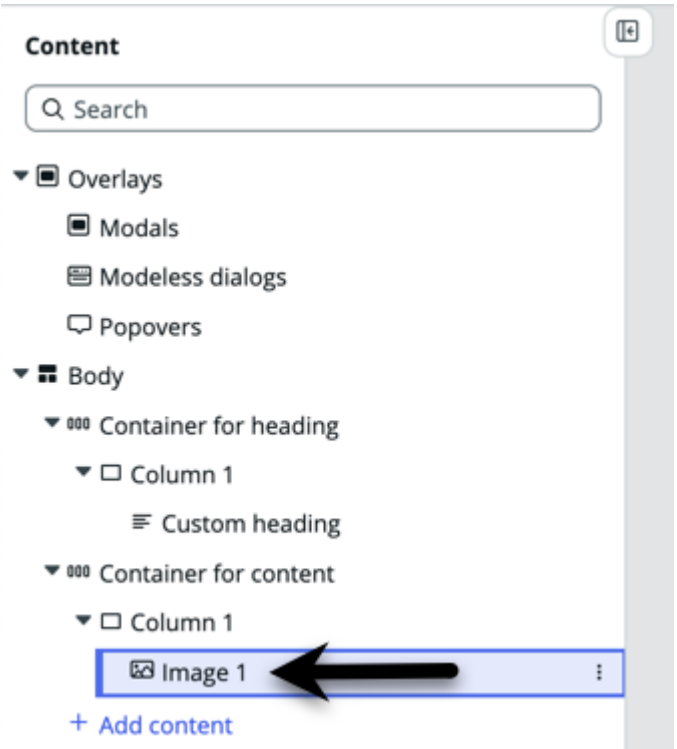
9. Add an image component to your page.
An image is a nice way to add a visual for your page. In this quick start, you add a stock photo that comes with UI Builder. But you can add any image that is available to you.

a. Select **+ Add content** in the content tree below the container created in step 8.

b. Type `image` into the search and select the **Image** component.
A default image is loaded in the image component. You can add your own image by adding a URL to the image. You can use images hosted on the internet or images in the ServiceNow AI Platform®. If you use an image hosted on an external site, you must use the `https` protocol for security.

c. Select the **Image 1** component in the **Content** tree to highlight the image.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



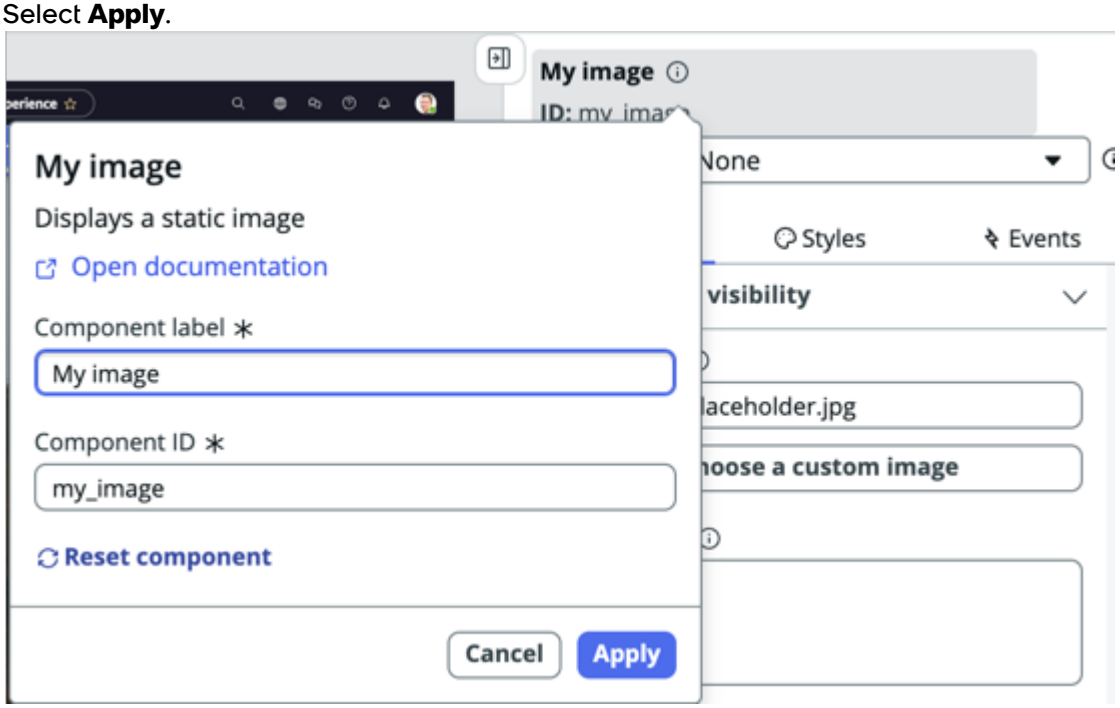
d. Select **None** in the configuration panel.

e. Select the component name in the configuration panel.

f. In the **Component label** field, type `My image`.

g. In the **Component ID** field, type `my_image`.

h. Select **Apply**.



i. Select **Save**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

You added an image component to your page.

10. Select **Preview** in the UI Builder header to preview your page.
    Congratulations! You completed the UI Builder quick start. Your page generates a preview of how it looks in your Workspace or portal experience.

Result

1. Created your first page in UI Builder.
2. Added your first container component to the page. You changed the label for the container.
3. Added a **Heading** component as a title to the container. You changed the text of the heading. You also changed the label.
4. Added a second container component to your page. You changed the label for the container.
5. Added an **Image** component. You changed the label of the image component.
6. Saved your new page.
7. Previewed your page in the browser.

You successfully completed the UI Builder quick start!

UI Builder tutorial

Learn how to use the basics of UI Builder to create a page called My Tutorial.

Before you begin

Role required: ui_builder_admin

In this UI Builder tutorial, you perform the following tasks to build a page in UI Builder:

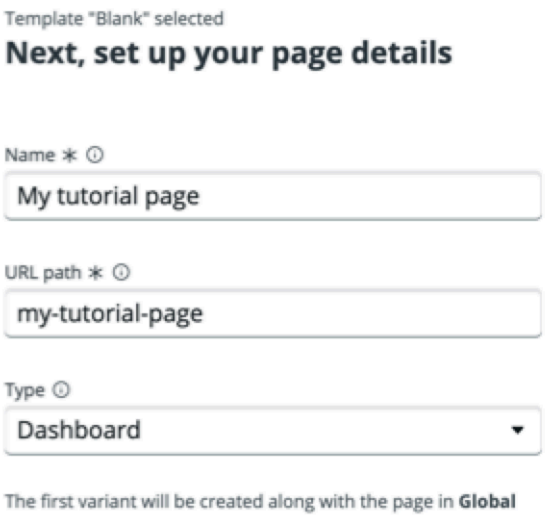
- Start UI Builder.
- Create a page for your workspace or custom portal experience. For more information about creating pages, see Create a page in UI Builder.
- Change the layout of the page to have two columns. For more information, see Organize components in UI Builder pages.
- Build your page by adding two container components.
- Rename your container components in the content tree.
- Add a **Heading** component and a **Button** component to the first container. Add a **data visualization component** to the second container. For more information about components, see Customize UI Builder pages using components.
- Configure your components as follows:
  - Link the button to the ServiceNow® website.
  - Connect the **Data visualization** component to a data source to display task data. For more information about data resources, see Dynamically expose data in UI Builder pages (advanced feature).
- Save your page often.
- Preview your page to see how it looks in a browser.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
   You can also type `UI Builder` directly in the **Filter navigator**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

2. From the UI Builder home page, choose an experience you want to work in.
   If you don't see any experiences listed in which to work, contact your administrator to get access to an experience or create an experience. For more information, see Configure how users interact with your applications in UI Builder.
3. Create a page.

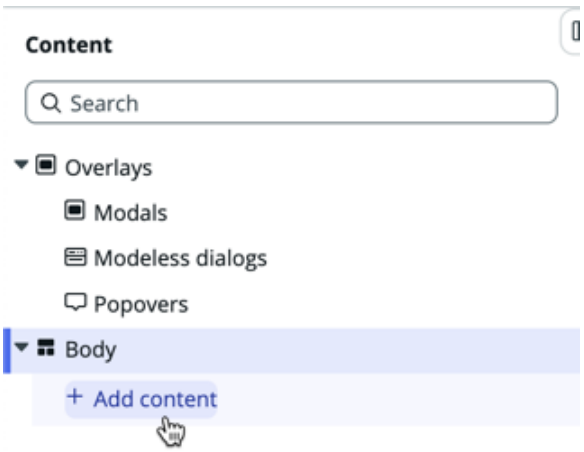
To do this task	Do the following
Create a page	**a.** Select the **+** icon in the **Pages** section.  **b.** Select **Create a new page**. **c.** Select **Create from scratch instead**.  **d.** Enter the name of your new page. The name can be anything you want. In this example, you can type `My tutorial page`. **e.** In the **Path** field, a path is automatically entered based on your page name. In this example, the Path is `my-tutorial-page`. The path is the URL of the page. You can change this path to anything you want, but the path must be unique. The path has to be lowercase and contain no spaces. The **URL preview** shows what the path of your page will be.  **f.** Select a URL page type in the **Type** drop-down list. URL page types help you categorize and find important pages in the experience view. **g.** Select **Continue**. **h.** (Optional) Add parameters to your page URL.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

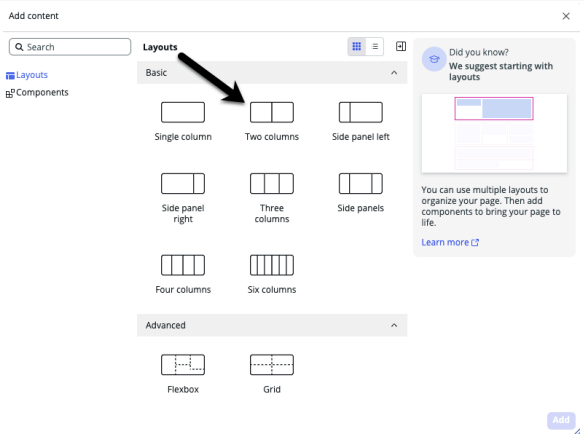
To do this task	Do the following
	i. Select **Looks good**. j. Enter the name of your page variant. The name can be anything you want. In this example, you can type `My first page variant`. k. Add an audience for the page variant by selecting **+ Add an audience**. In this example, select **Admin**. l. Select **Continue**. m. On the next screen, select **Build responsive**. n. Select **Create**. o. After your page is created, the page editor screen ap pears. You can change advanced settings such as the **URL settings** and **Variant** settings. For this tutorial, you can skip these advanced options. For more information on changing advanced settings, see Create a UI Builder page: Advanced settings.

Congratulations! You created your tutorial page! The page is empty of content. You add components to the page to build functionality to it. Components are the building blocks of a page. UI Builder comes with many components ready for you to add to your page. Components can be as simple as a **Heading**, or as complex as a **List**.

4. Change the layout of the page to a two-column layout.
For more information about layouts, see Organize components in UI Builder pages.

To do this task	Do the following
Change the layout of the page to two columns	a. Select the **Body** level of the **Content** tree to highlight the page. b. Select **+ Add content** in the content tree.  c. Select the **Two columns** layout.

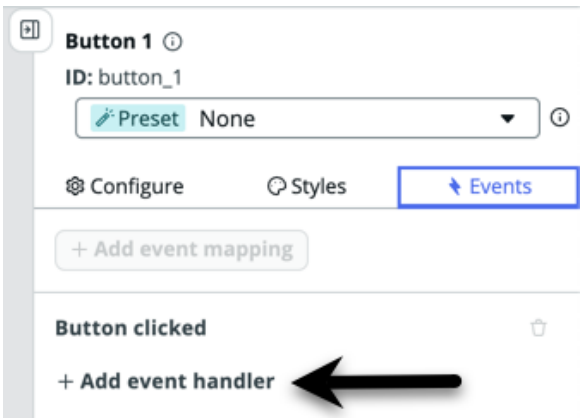
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

To do this task	Do the following
	 **d.** Select **Add**. **e.** The two column layout appears.

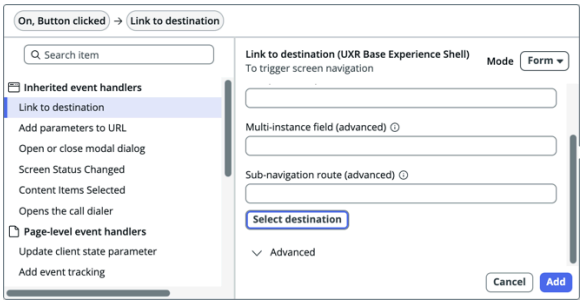
5. Add a **Heading** component, and a **Button** component to the left column.

To do this task	Do the following
Add a heading component to your left-column container	You can add components to the page in different ways. For more information on the ways you can add components to your page, see Table 1. **a.** Select the **+** button in the middle of the left column. **b.** Type `Heading` in the search field. **c.** Select the **Heading** component to add it to your container. A heading component is a way to add text or a title to your page. **d.** Select **Add**. **e.** Select the component name in the configuration panel to see the **Component label** and **Component ID**. You can change these to anything you want, as long as they're unique. **f.** Select **Apply**. **g.** Configure your **Heading** component. Component authors configure components to expose the appropriate proper ties required to set up their components. ▪ Set the style to **Header-primary**. The Style changes the class of the heading text, like font, weight, and col or. For example, if you select **Header-secondary**, the text is smaller. Different headings sizes are useful if you have two headings and want the second heading small er that the primary heading. ▪ Enter `My Tutorial` as the heading text in the **Label** field. ▪ Leave the Level as **1**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

To do this task	Do the following
Add and configure a button component to your left-column container	Add a **Button** component to your page. Configure the button by changing the text and the way that it looks. Then change what happens when you click the button by adding an event handler for the button. The event handler adds an action to the button to link to the ServiceNow® home page. So when you click the button, the ServiceNow® home page opens in a new tab of your browser. For more information about configuring components, see . **a.** Select the **+** on the bottom edge of the heading component that we added in the previous section. **b.** Type `Button` in the search field. **c.** Select the **Button** component in the list. For more information on adding components to your page, see Table 1. **d.** Select the component name in the configuration panel to see the **Component label** and **Component ID**. You can change these to anything you want, as long as they're unique. **e.** Configure the button component as follows. ▪ Enter `Home Page` in the Label field to change the name of your button. ▪ Change the Size to **Large**. ▪ Enter `home` in the **Icon** search field. ▪ Select **Home Outline**. **f.** Add an event handler to configure what action applies to the button. ▪ Select the **Events** tab in the configuration panel. ▪ Under Button clicked, select **+ Add a new event handler**.  ▪ Under Inherited event handlers, select **Link to destination**. ▪ Click **Select destination**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

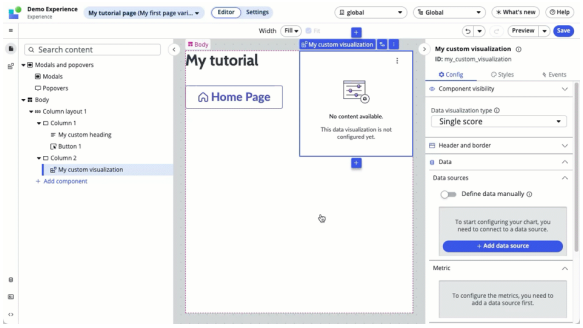
To do this task	Do the following
	▪ In the **Select destination** screen, select **External URL**. ▪ Type your URL. For this tutorial, type `https://www.servicenow.com` Select **OK**. ▪ Select **Add**.  For more information about event handlers, see Manage actions in UI Builder pages. **g.** Select **Save**. It is always a good idea to save your page often as you work. **h.** Select **Preview**. **i.** Test the button you created in the previous steps.

You added and configured the **Heading** and **Button** component for your page.

6. Add a **Data visualization** component to the container.
A data visualization component contains data that you display in a visual manner. Configure the data visualization component parameters. Then, add a data resource to it.

To do this task	Do the following
Add and configure a data visualization component to your right-column container	You can add components to the page in different ways. For more information on the ways you can add components to your page, see Table 1. **a.** Select the **+** button in the middle of the right column. **b.** In the **Search** field, start typing `Data visualization` until you see the **Data visualization** component, then select the component to add it to your page. **c.** Select the component name in the configuration panel to see the **Component label** and **Component ID**. You can change these to anything you want, as long as they're unique. **d.** Add a data resource to your data visualization component to bring in customer account data. ▪ Select the data visualization component, then select the **Configure** tab. ▪ Select **Add data source**. ▪ Type `Task [task]` in the search field and look for a task that is available on your UI Builder instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

To do this task	Do the following
	▪ Select **Add this source**.  **e.** Select **Data visualization type** drop-down and select **Di al**. You can choose a variety of visualization types to best suit your data. **f.** Select **Header and border** in the configuration panel. **g.** Enter `My Tasks` in the **Chart title** field.

You've added a **Data visualization** component to the right column and configured it. You also added a data resource to bring in customer account data.

7. Save the new page one last time.

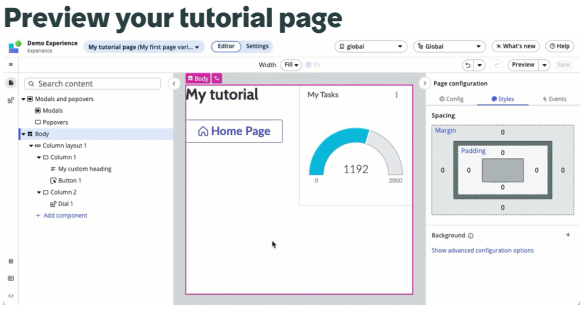
To do this task	Do the following
Save your page	**a.** In the far right corner of UI Builder, select **Save**. **b.** Now that you saved your page, you can access it in your experience at any time.

8. Now preview your page to see what it looks like in a browser.

To do this task	Do the following
Preview your page	**a.** Select **Preview** in the UI Builder header. **b.** See the heading in the left column of your page. **c.** Select the **Home Page** button. The ServiceNow® home page opens in another browser tab. Close the browser tab to go back to your preview page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

To do this task	Do the following
	d. See the customer account data visualization in the right column of your page.



Congratulations! You completed the UI Builder tutorial.

Result

1. Created a page in UI Builder.
2. Changed the layout of your page to have two columns.
3. Added a **Heading** component as a title to your left column.
   - Changed the text of the heading.
   - Changed the label of the heading.
4. Added a button component below your heading component.
   - Configured your button.
   - Added an event handler for your button.
   - Added an event handler to set up a link to a web page. When you click the button, it opens the web page.
5. Changed the label for the right column.
6. Added a **Data visualization** component.
   - Configured the data visualization component.
   - Added a data resource to bring in task data.
7. Saved your new page.
8. Previewed your page.

UI Builder and configurable workspaces

Use UI Builder to create pages for your configurable workspace experiences.

A workspace is a collection of tasks and workflows in a single focused working area that enables a user to efficiently complete an entire job. It includes tools that a user can employ to quickly and easily assist customers and resolve questions and issues. A workspace also includes features that enable a user to be more efficient, including a multi-tab interface for managing multiple cases and a contextual display that provides quick orientation to the current task.

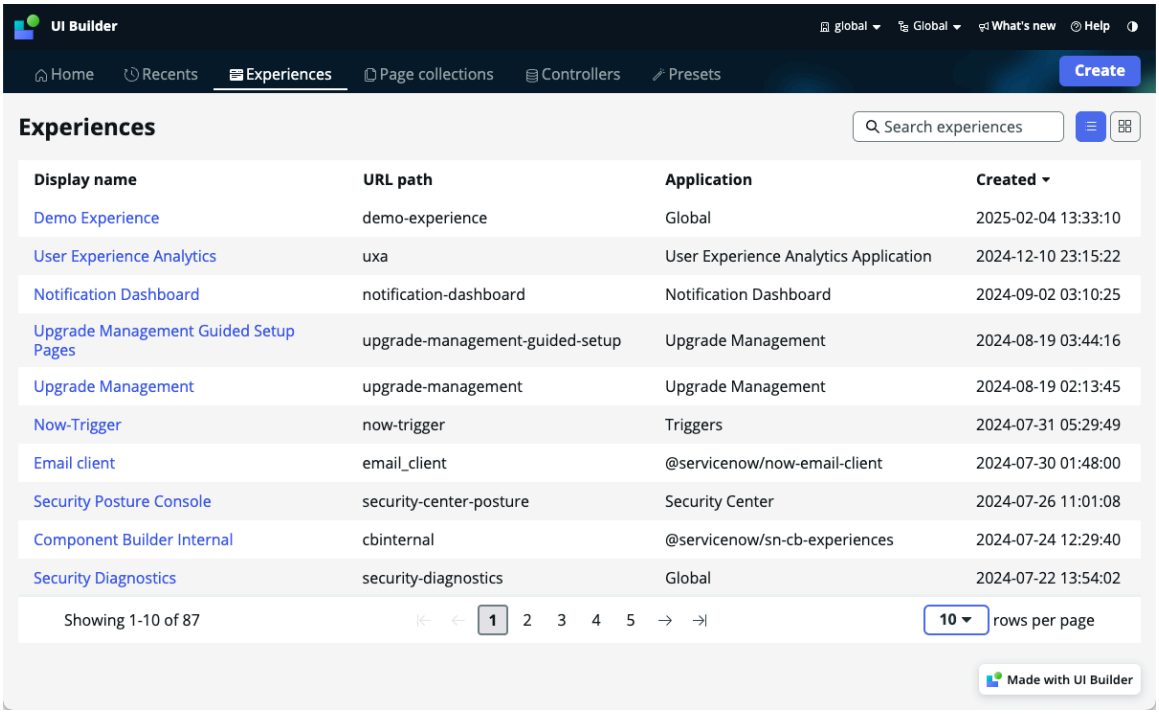
You can use UI Builder to create pages for your workspace experience. UI Builder provides base system components that you can use to build workspace pages.

To create a page in UI Builder for a Workspace experience, you navigate to **All** > **Now experience framework** > **UI Builder** to open UI Builder.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The UI Builder home displays your available experiences under the **Experiences** tab. If you are working in a Workspace experience, you should see the experience listed here.

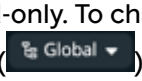
UI Builder home



Experience settings for UI Builder

Learn about the UI Builder experience settings to build your own workspace and custom portal experiences.

By changing the settings of the workspace or portal experience that you're working in, you can affect how your users interact with the experience, how the experience looks, and how users navigate to and from the experience.

Before you can edit the experience settings, you must be in the correct application scope. If you're in a different scope, the experience settings are read-only. To change your application scope, go to the main header, select the application picker (

), and then select the application scope that you want. For more information about the application scope, see Learn about security and roles.

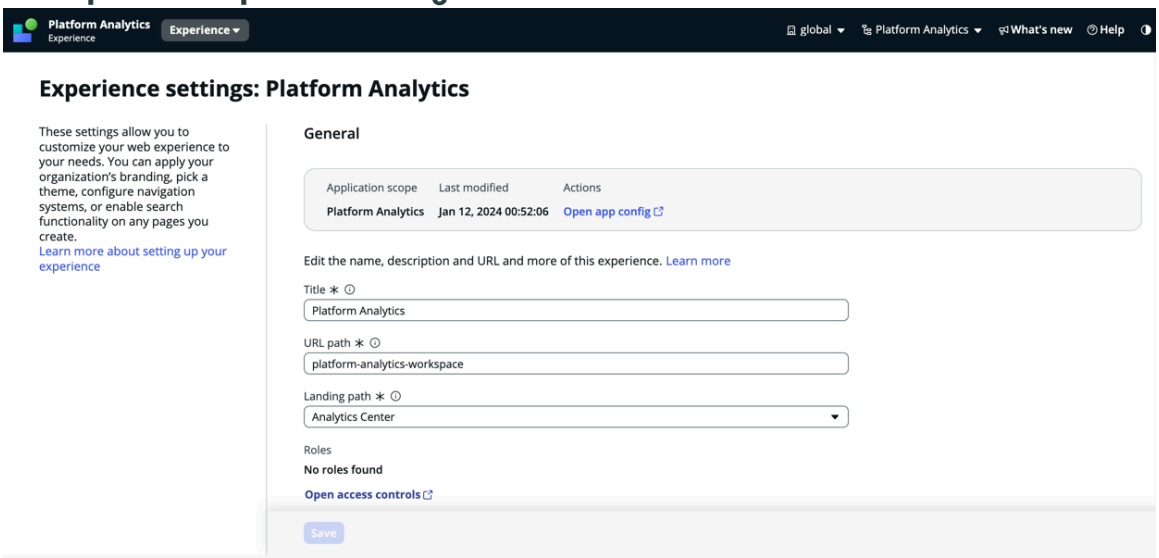
Workspace experience settings

You can change these settings for your workspace experience:

- Modify or add some general settings for your experience. For example, you can change the title, description, and the path of the workspace or portal.
- Apply your organization's branding to all pages in your experience or to a page within your experience.
- Configure your side navigation settings to add pages to the side navigation in your experience.
- Turn on or turn off the notifications for your experience.
- You can change the global search settings for your experience. For example, you can choose to show or hide the search bar in your experience or change the search source that determines where the search results come from.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Workspace edit experience settings screen



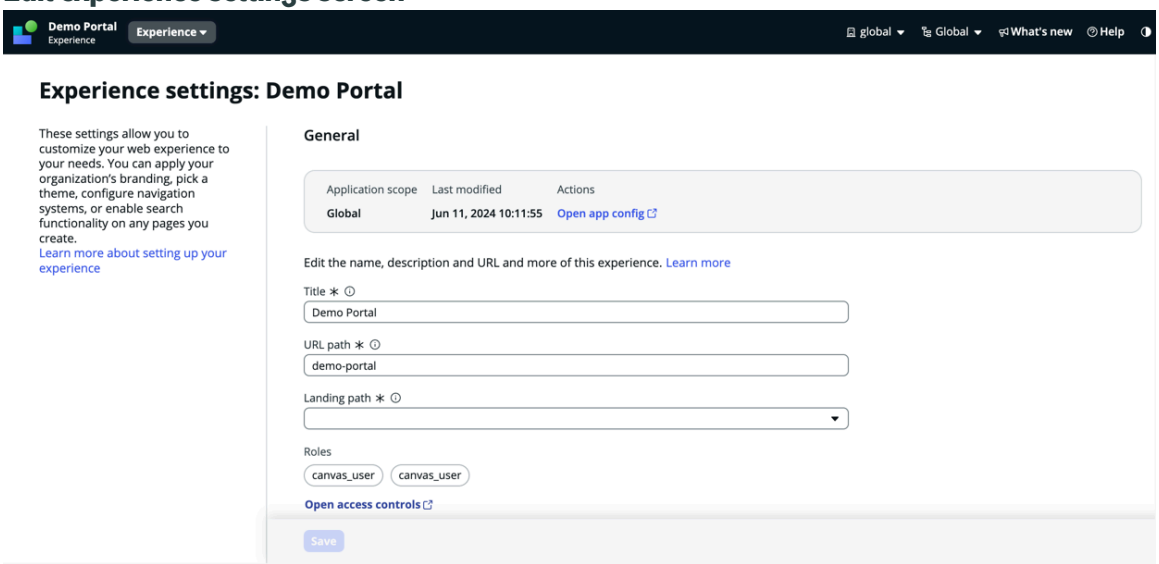
To learn more about the workspace experience settings, see Configure UI Builder workspace experiences.

Portal experience settings

You can change these settings for your portal experience:

- Modify or add general settings for your experience. For example, you can change the title, the description, and the path of the workspace or portal.
- Apply your organization's branding to all of the pages in your experience or to a page within your experience.
- Set up the navigation and menu settings in the app shell of your portal experience. The app shell, which is the wrapper of the portal contents, is similar to a Google Chrome web page.
- Control whether the global search functionality is visible to the users of your portal experience. For example, you can enable or disable the global search for either your public or private pages.

Edit experience settings screen



To learn more about the portal experience settings, see Configure UI Builder portal experiences.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Helpful resources for UI Builder

Some ServiceNow resources that can provide you with helpful information.

Some ServiceNow resources that can provide you with helpful information are:

Video

ServiceNow Dev Program YouTube channel .

Community YouTube channel

UI Builder 101 with Chris Johnson

Next Experience Academy UI Builder



Component Documentation

Components documentation

Component Design Documentation

ServiceNow Community

ServiceNow Community site

You and I Builder Bytes!



Development information

ServiceNow Developer site

ServiceNow University

ServiceNow University

Support

https://support.servicenow.com/now

Search the Known Error Portal for known error articles

UI Builder glossary

Learn about the terms and concepts used in UI Builder (UIB).

action

Actions in UI Builder are specifically activity on a page or within a page component. Events and event handlers are used to add actions. For example, add a button component to a page and then add an event handler to apply an action for the button, such as opening a web page.

app shell

App shells are the static elements of a web experience (for example, the header, footer, and menu navigation) that stays with the end user as they navigate throughout the

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

experience. App shells are primarily used and supported in Workspace and Portal experiences.

binding

See data binding.

client script

Client-side JavaScript that interact with components and client state parameters on a page. Client scripts are mapped to events as event handlers in UI Builder.

client state parameter

Page variables that are defined for a page to store a piece of data (a client state) only for that page. For example, create three client state parameters to store the input needed to create a record and specify when to refresh the list. Page variables can be updated using client scripts and events to make a page dynamic.

component (UI Builder)

Components are used in the UI Builder to build pages. Components have an interface that an end user can view and interact with. Components can talk to each other through events and properties. Commonly used components include Heading, Image, List, Form, and Button.

component id

Used to reference a component when adding a script or binding data to the component. A component ID is automatically created (based on the component label) when you add a component to a page, but the component ID can be edited.

component preset

Presets apply predefined configuration values and event mappings to components. Presets apply prebuilt configurations to component properties and events handlers. Presets are only available for certain components.

component properties

Available in the Configuration panel and used to configure a component. Each component has unique properties. Component properties are specified within each tab on the Configuration panel: Config, Style, and Event. Some components have presets available. Use the component presets to set component properties automatically.

controller

A type of data resource that includes data and event logic and enables component presets. Controllers are added automatically when using a page template. There are two types of controllers:

- Data controllers contain data resources and can be manually added to a page
- UI controllers are added to pages when using page templates and can't be added manually. Creating controllers isn't supported currently.

data binding

The process of associating data with a UI element that displays the information.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

data resource

A dynamic, reusable way of defining what data to fetch for a page's components.

entity view action mapper

Also known as EVAM. Standardizes the format for displaying data in cards and lists.

event (UI Builder)

Action a user takes (such as selecting a button) or an occurrence that happens on a page. Most UI Builder components, pages, and data resources have default-associated events. Use event handlers with the events to add additional actions to pages.

event handler

An action performed when an event occurs.

event mapping

The process of identifying an event handler to run when an event occurs.

macroponent

A core data structure that drives UI Builder pages. Its fields contain JSON data that builds the page.

modal

A user experience window that overlays another content window and takes control of the user experience.

now code editor

A rich-text editor that supports CSS, HTML, JavaScript, XML, and JSON. Use Now Code Editor to change UI configuration, data resource configuration, styles, events, client-side scripts, and server-side scripts in Next Experience UI Builder components.

page

Collection of column layouts, columns, and components. Create or customize multiple UI Builder pages for workspace and portal experiences.

page collection

Group of pages that can be reused in experiences within tabs or modals.

popover

A page overlay that enables users to continue using the rest of the page. Popovers can be configured just like UI Builder pages with text, components, images, fields, and menu items.

repeater

In UI Builder, a repeater is a component that acts as a basic loop that repeats the data you provide in multiple components. Repeaters use an array or an array of objects. Repeaters bind values to a data array property. For example, [{"task": "A"},{"task": "B"}], repeats the content inside it two times.

UI Builder (UIB)

A WYSIWYG web user interface builder. UI Builder enables developers to build new pages or customize existing pages for Agent Workspace and portals using Now Experience UI Framework components.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

variant

The version of a UI Builder page with access controlled by role or condition. Create variants of pages to target experiences for different audiences. For example, create a home page for agents and a variant for managers at the same URL. Alternatively, create a page variant that users see under different conditions.

Learning UI Builder

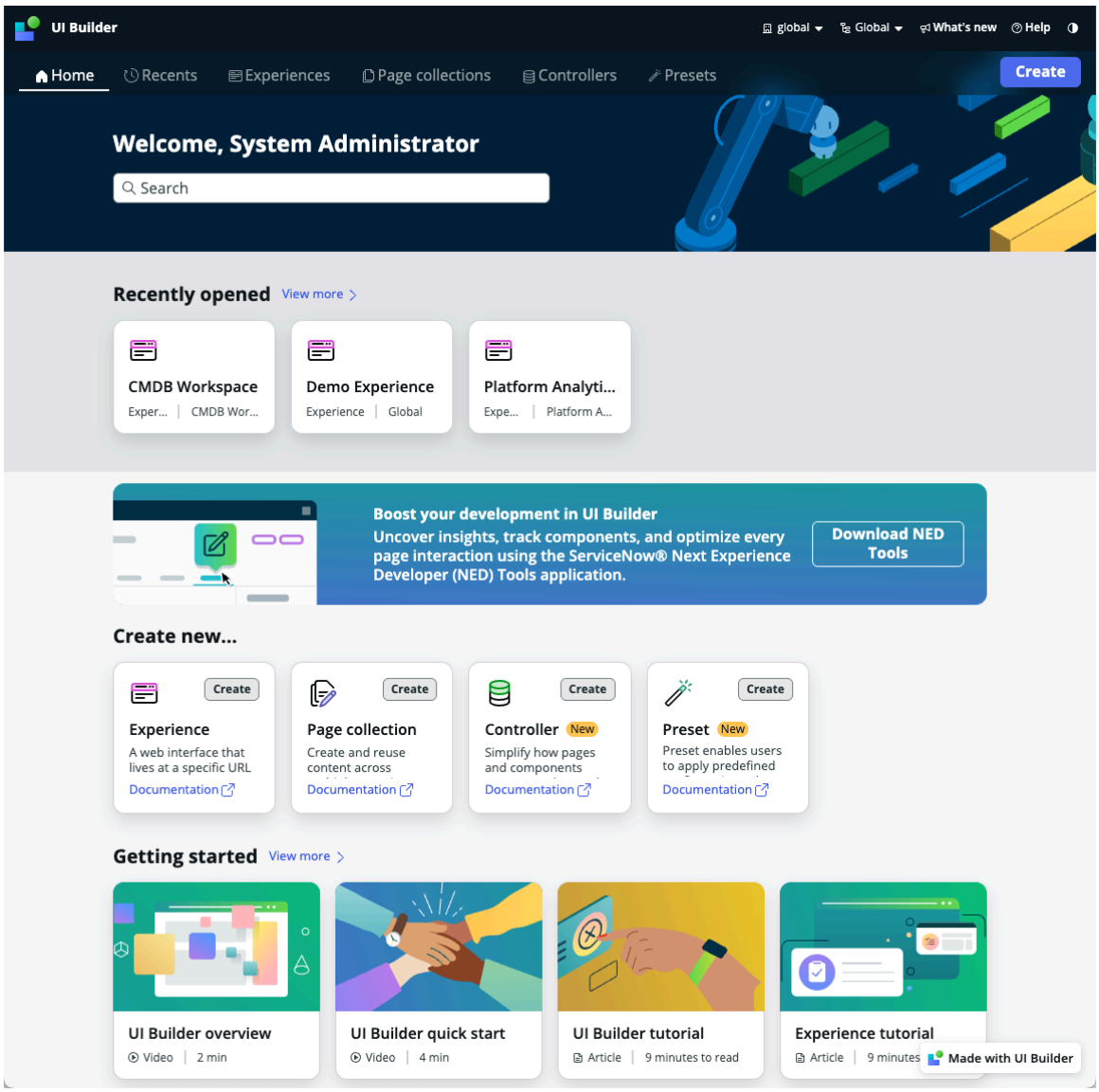
Learn how to navigate the UI Builder and explore some common use cases to learn by example.

Navigate the UI Builder application

Learn how to navigate through UI Builder.

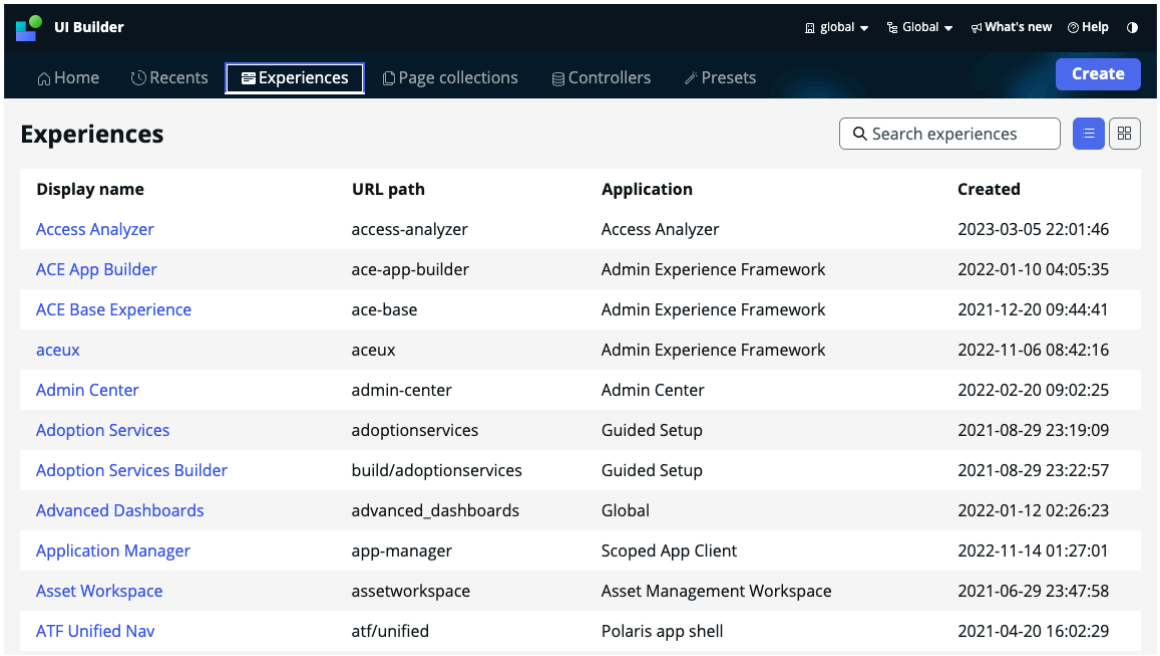
UI Builder home

The UI Builder home page provides easy access to recently opened experiences, options for creating experiences or page collections, and help for getting started using UI Builder. You can also use the search bar to quickly find your experiences by name.



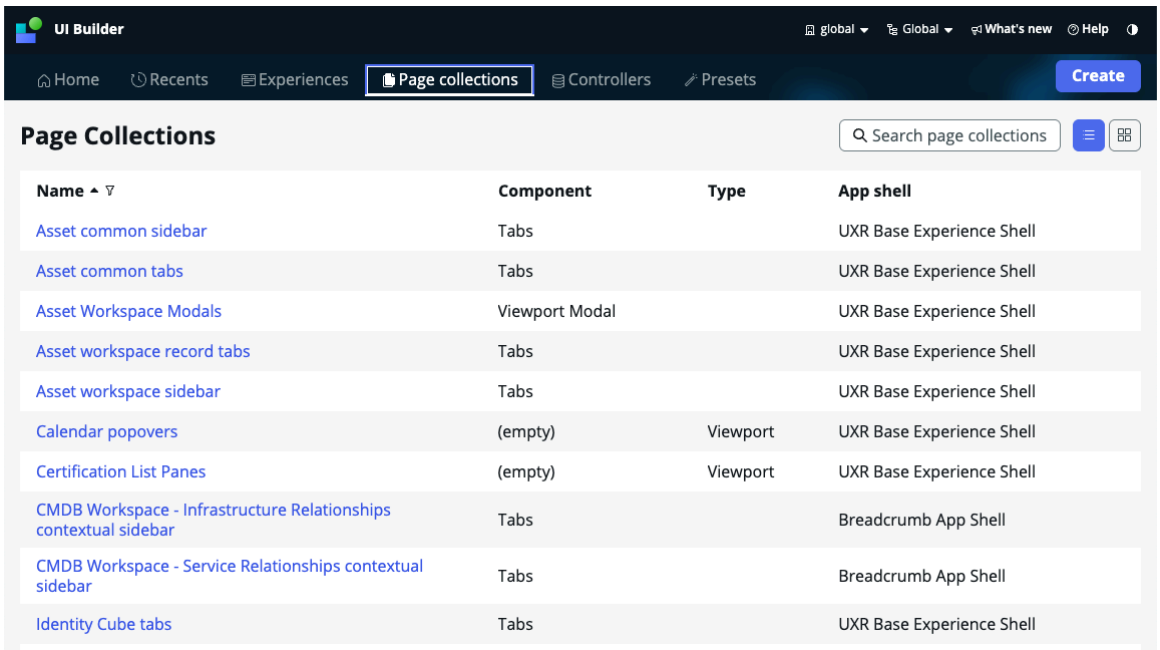
You can access the list of experiences in the top by selecting the **Experiences** tab in the top bar.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Display name	URL path	Application	Created
Access Analyzer	access-analyzer	Access Analyzer	2023-03-05 22:01:46
ACE App Builder	ace-app-builder	Admin Experience Framework	2022-01-10 04:05:35
ACE Base Experience	ace-base	Admin Experience Framework	2021-12-20 09:44:41
aceux	aceux	Admin Experience Framework	2022-11-06 08:42:16
Admin Center	admin-center	Admin Center	2022-02-20 09:02:25
Adoption Services	adoptionservices	Guided Setup	2021-08-29 23:19:09
Adoption Services Builder	build/adoptionservices	Guided Setup	2021-08-29 23:22:57
Advanced Dashboards	advanced_dashboards	Global	2022-01-12 02:26:23
Application Manager	app-manager	Scoped App Client	2022-11-14 01:27:01
Asset Workspace	assetworkspace	Asset Management Workspace	2021-06-29 23:47:58
ATF Unified Nav	atf/unified	Polaris app shell	2021-04-20 16:02:29

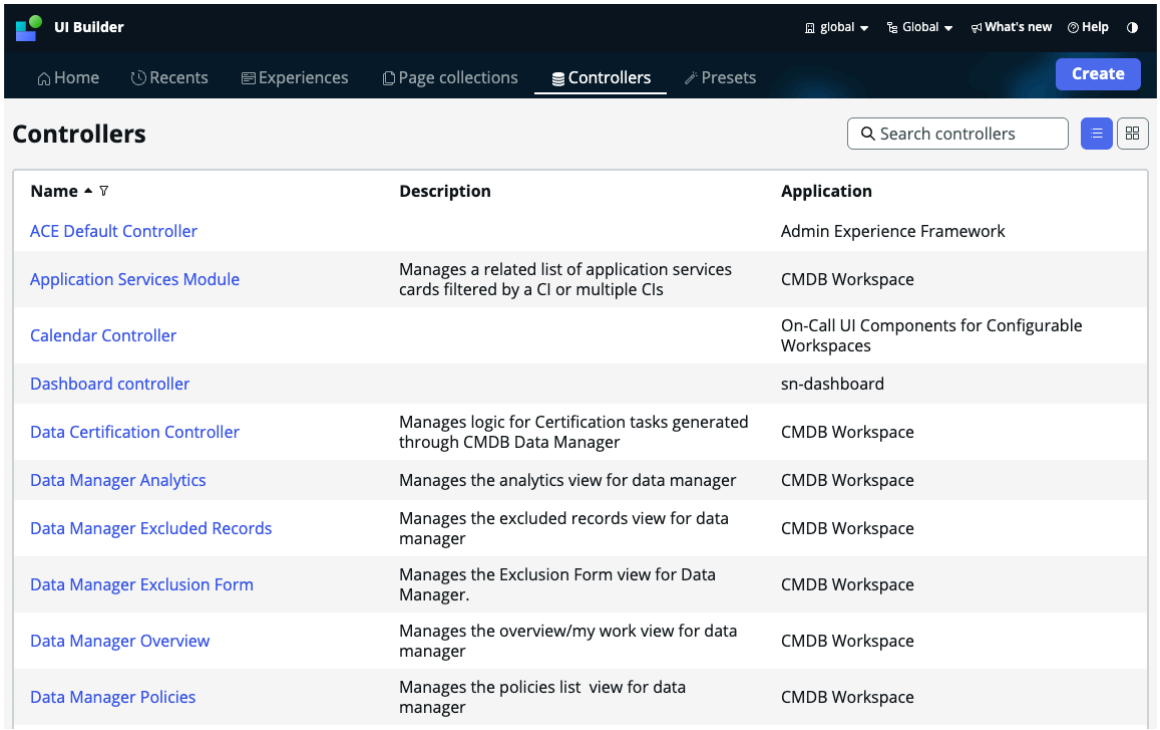
You can access the list of page collections in the top by selecting the **Page collections** tab in the top bar.



Name	Component	Type	App shell
Asset common sidebar	Tabs		UXR Base Experience Shell
Asset common tabs	Tabs		UXR Base Experience Shell
Asset Workspace Modals	Viewport Modal		UXR Base Experience Shell
Asset workspace record tabs	Tabs		UXR Base Experience Shell
Asset workspace sidebar	Tabs		UXR Base Experience Shell
Calendar popovers	(empty)	Viewport	UXR Base Experience Shell
Certification List Panes	(empty)	Viewport	UXR Base Experience Shell
CMDB Workspace - Infrastructure Relationships contextual sidebar	Tabs		Breadcrumb App Shell
CMDB Workspace - Service Relationships contextual sidebar	Tabs		Breadcrumb App Shell
Identity Cube tabs	Tabs		UXR Base Experience Shell

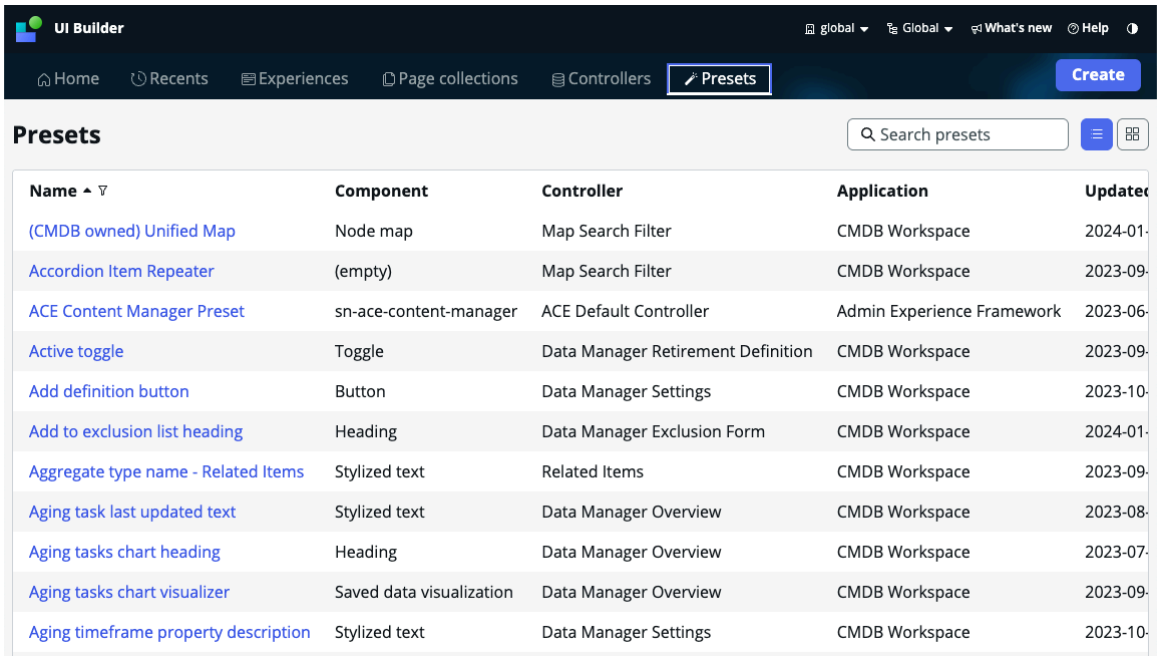
You can access the list of controllers in the top by selecting the **Controllers** tab in the top bar.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Name	Description	Application
ACE Default Controller		Admin Experience Framework
Application Services Module	Manages a related list of application services cards filtered by a CI or multiple CIs	CMDB Workspace
Calendar Controller		On-Call UI Components for Configurable Workspaces
Dashboard controller		sn-dashboard
Data Certification Controller	Manages logic for Certification tasks generated through CMDB Data Manager	CMDB Workspace
Data Manager Analytics	Manages the analytics view for data manager	CMDB Workspace
Data Manager Excluded Records	Manages the excluded records view for data manager	CMDB Workspace
Data Manager Exclusion Form	Manages the Exclusion Form view for Data Manager.	CMDB Workspace
Data Manager Overview	Manages the overview/my work view for data manager	CMDB Workspace
Data Manager Policies	Manages the policies list view for data manager	CMDB Workspace

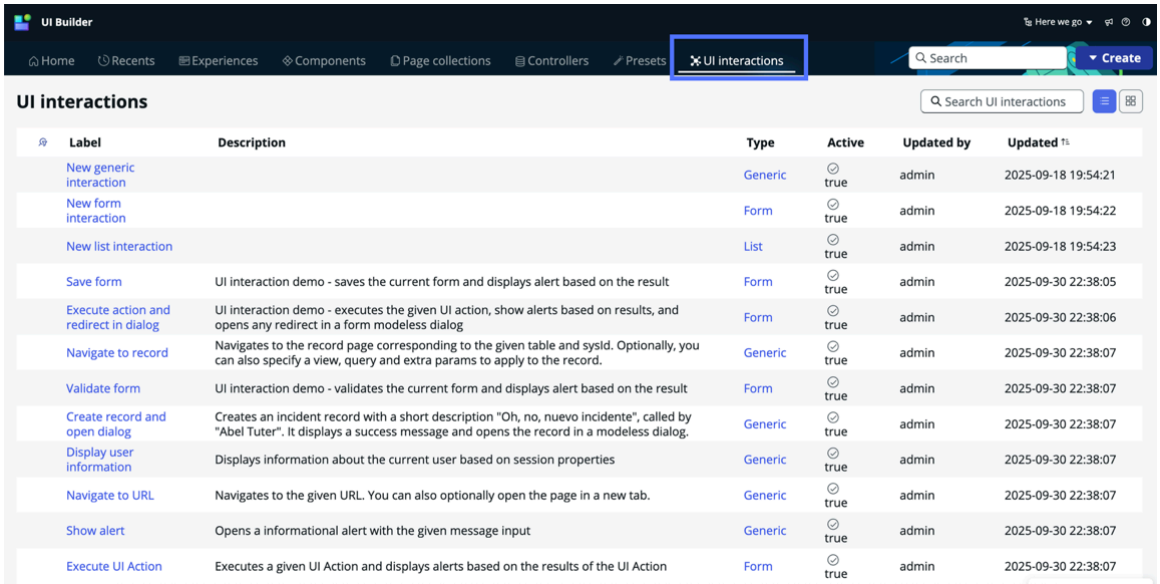
You can access the list of presets in the top by selecting the **Presets** tab in the top bar.



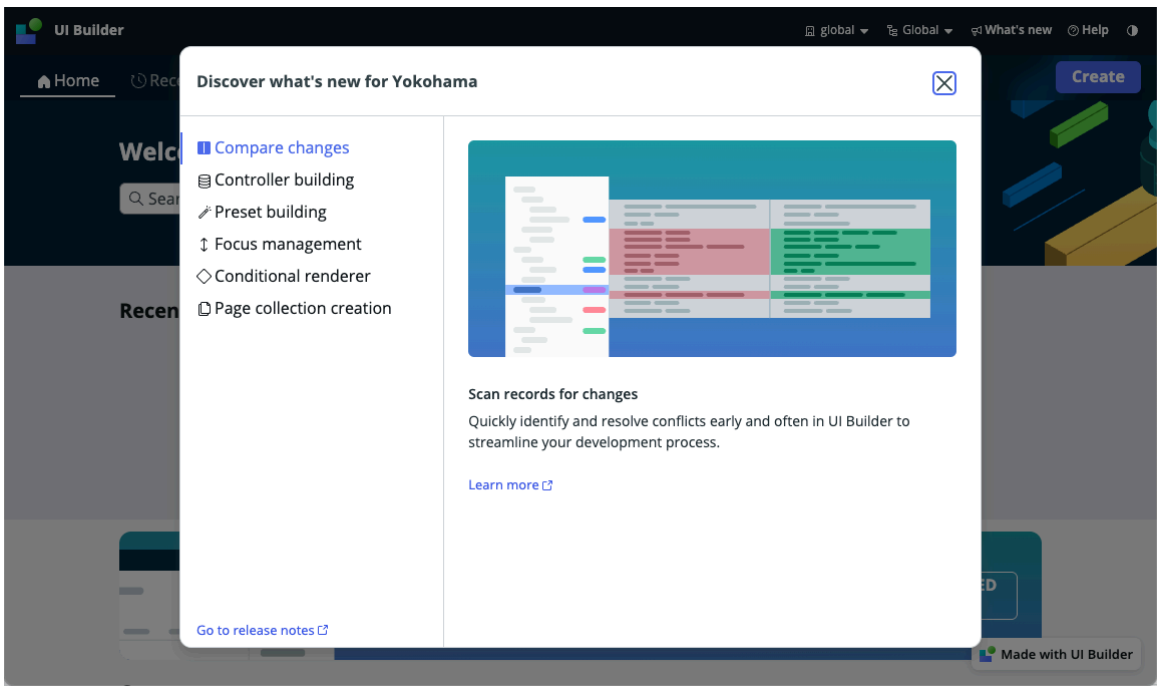
Name	Component	Controller	Application	Updated
(CMDB owned) Unified Map	Node map	Map Search Filter	CMDB Workspace	2024-01-
Accordion Item Repeater	(empty)	Map Search Filter	CMDB Workspace	2023-09-
ACE Content Manager Preset	sn-ace-content-manager	ACE Default Controller	Admin Experience Framework	2023-06-
Active toggle	Toggle	Data Manager Retirement Definition	CMDB Workspace	2023-09-
Add definition button	Button	Data Manager Settings	CMDB Workspace	2023-10-
Add to exclusion list heading	Heading	Data Manager Exclusion Form	CMDB Workspace	2024-01-
Aggregate type name - Related Items	Stylized text	Related Items	CMDB Workspace	2023-09-
Aging task last updated text	Stylized text	Data Manager Overview	CMDB Workspace	2023-08-
Aging tasks chart heading	Heading	Data Manager Overview	CMDB Workspace	2023-07-
Aging tasks chart visualizer	Saved data visualization	Data Manager Overview	CMDB Workspace	2023-09-
Aging timeframe property description	Stylized text	Data Manager Settings	CMDB Workspace	2023-10-

You can access the list of UI interactions in the top by selecting the **UI interactions** tab in the top bar.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Select the **What's new** link in the header to view the latest changes to UI Builder.

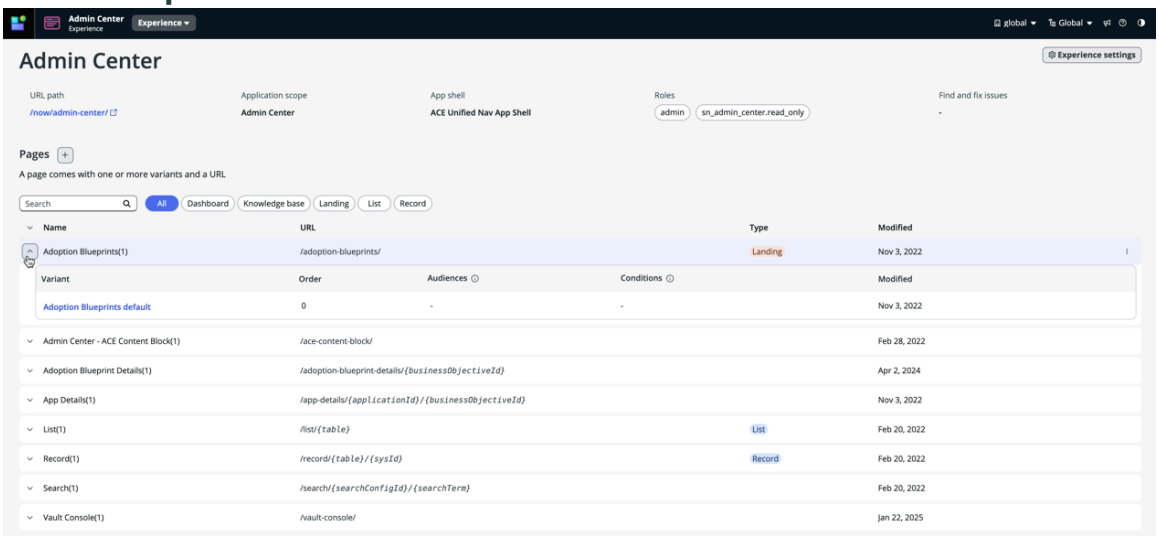


UI Builder experience view

The UI Builder experience view is a central place to view and understand the structure and details of an experience. Use the **Pages** list to see all pages and expand them to view associated variants. Locate pages or variants using the search field, which supports filtering by name, URL, page type, or variant.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

UI Builder experience view

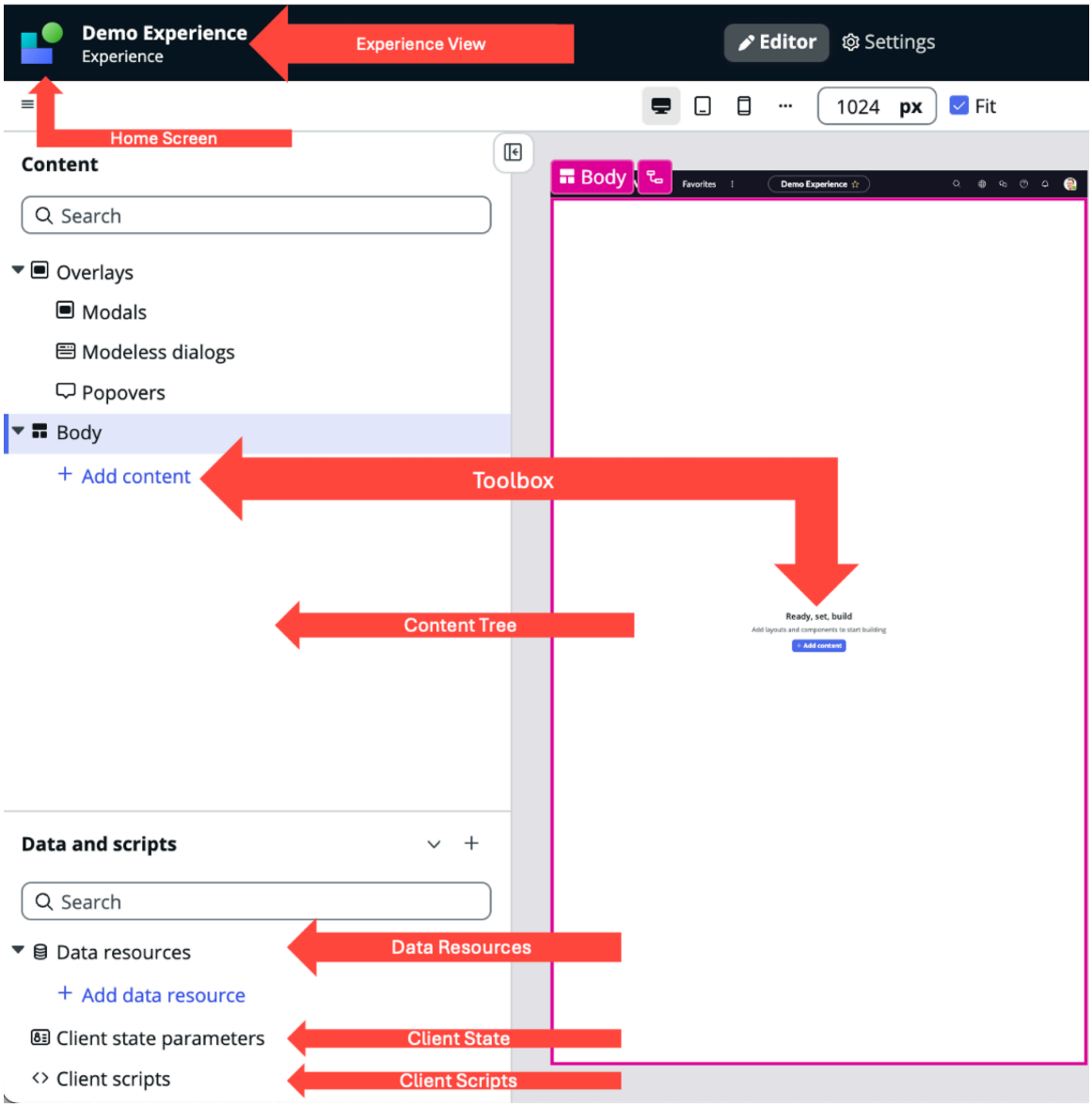


Structure of UI Builder

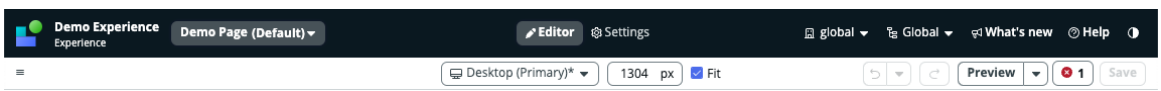
UI Builder is separated into the following areas.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Use the left vertical bar for access to the following in UI Builder:

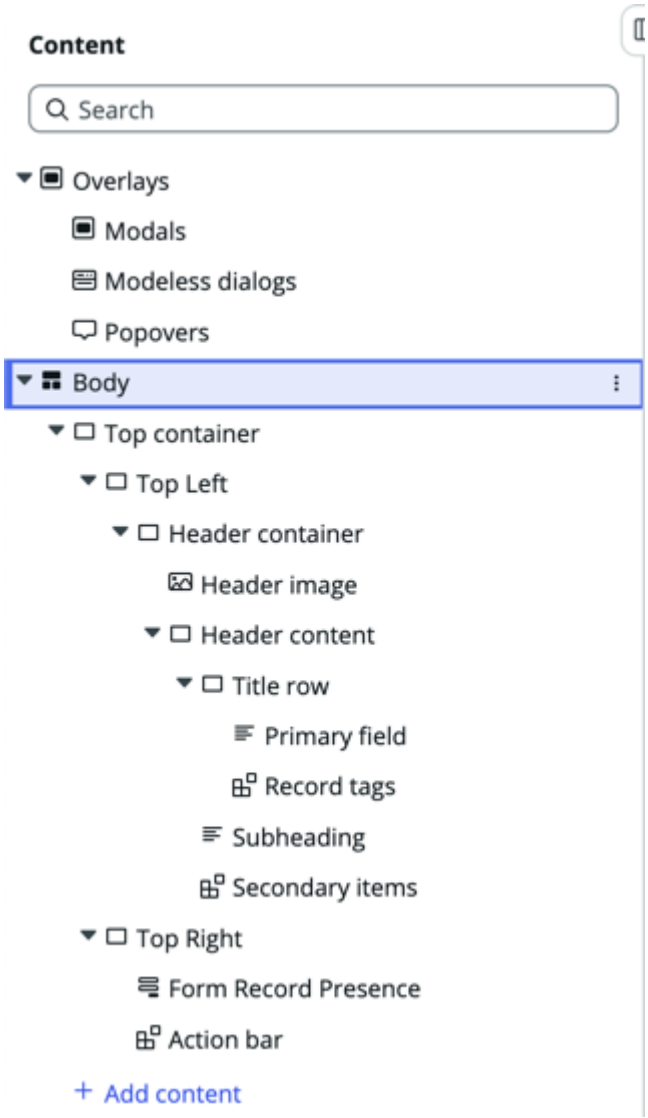


- Use the top bar to access pages in the experience and view Help. Change page settings, such as availability, order, and audience. Preview and open the URL path for the page you're editing to see what the page looks like for users. You can also select a domain and set application scope. View page errors and warnings by selecting the icon next to the save button. And finally, click the **Save** button often to save your page.



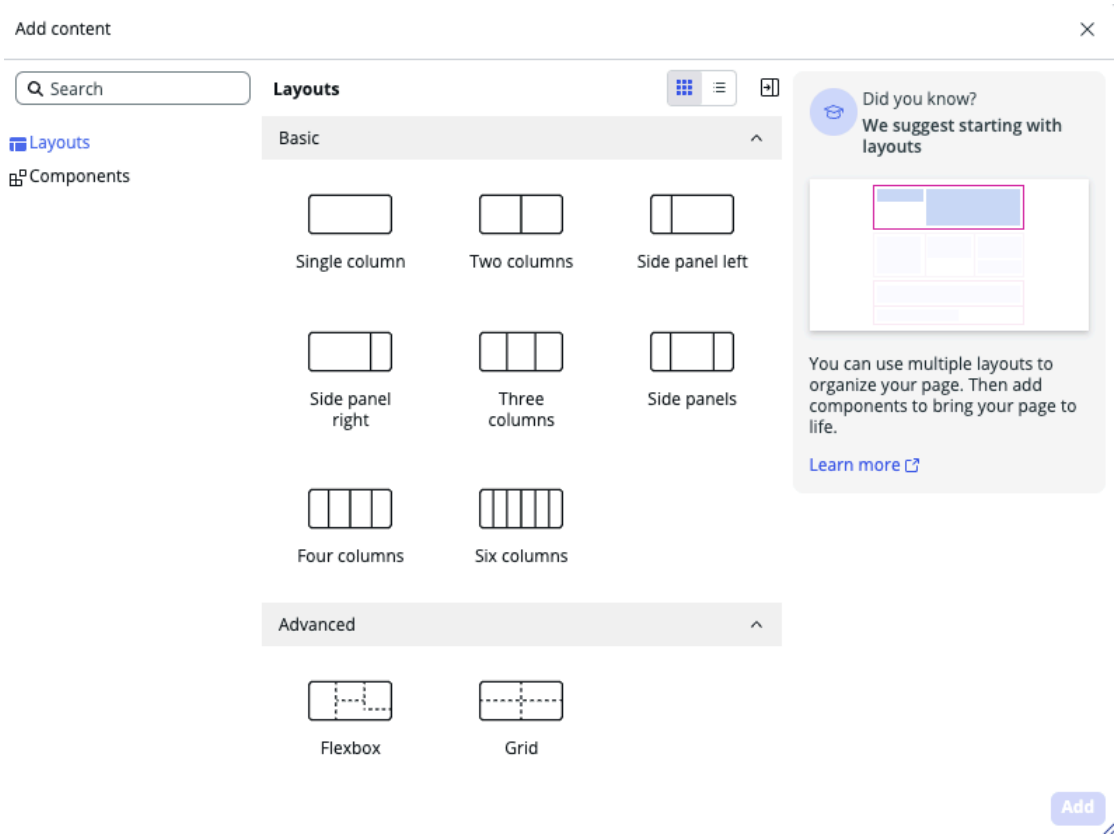
- Use the content tree to view and edit column layouts, columns, and components on your pages.

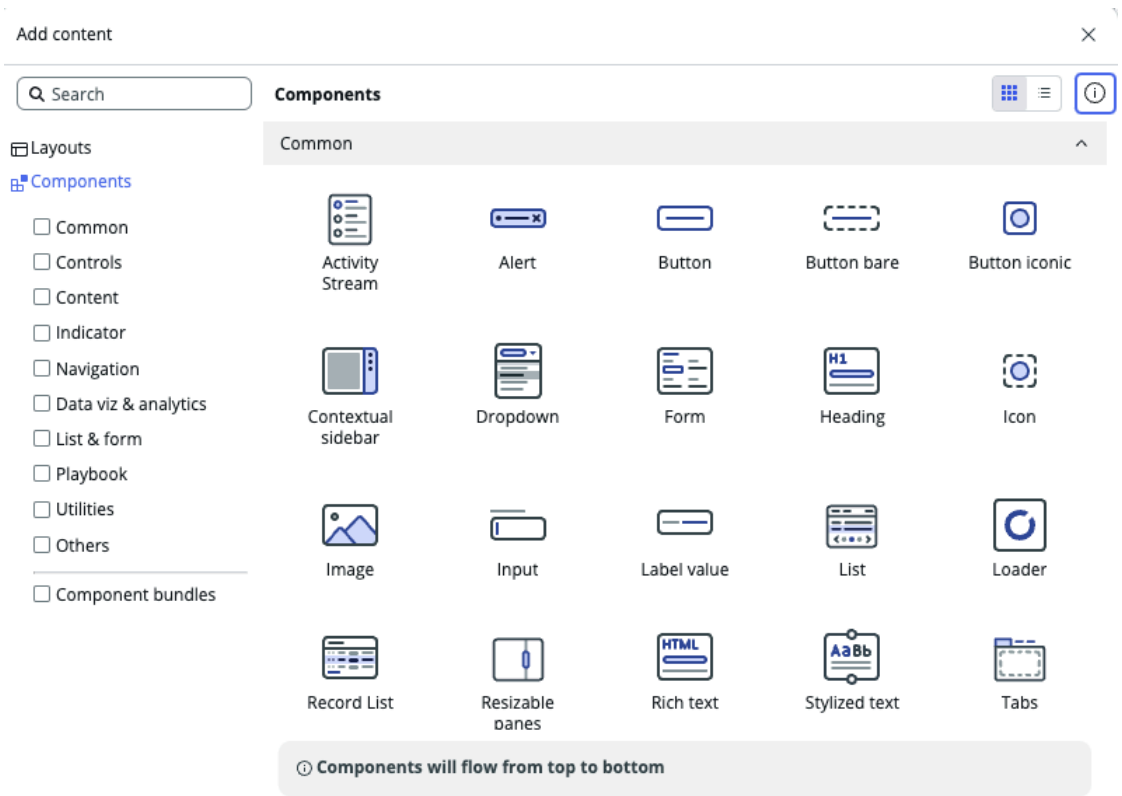
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



- Add column layouts and components to your page using the UI Builder toolbox.

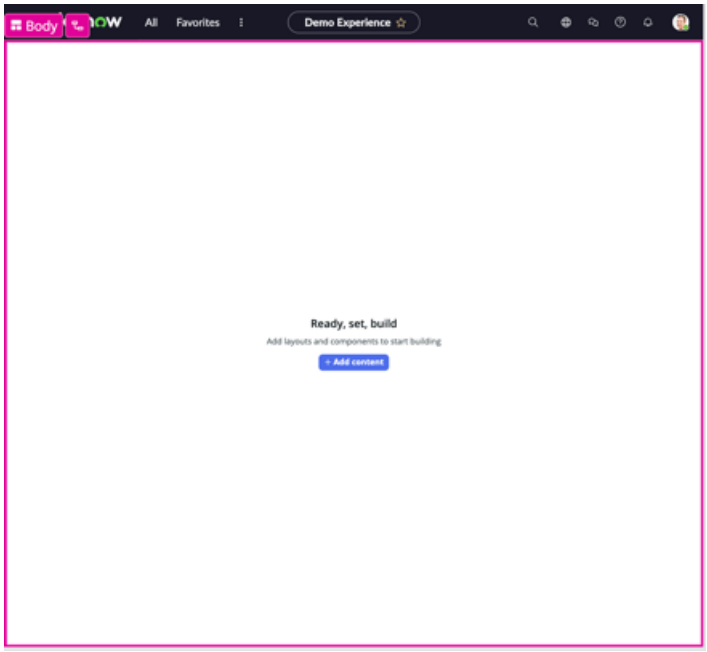
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



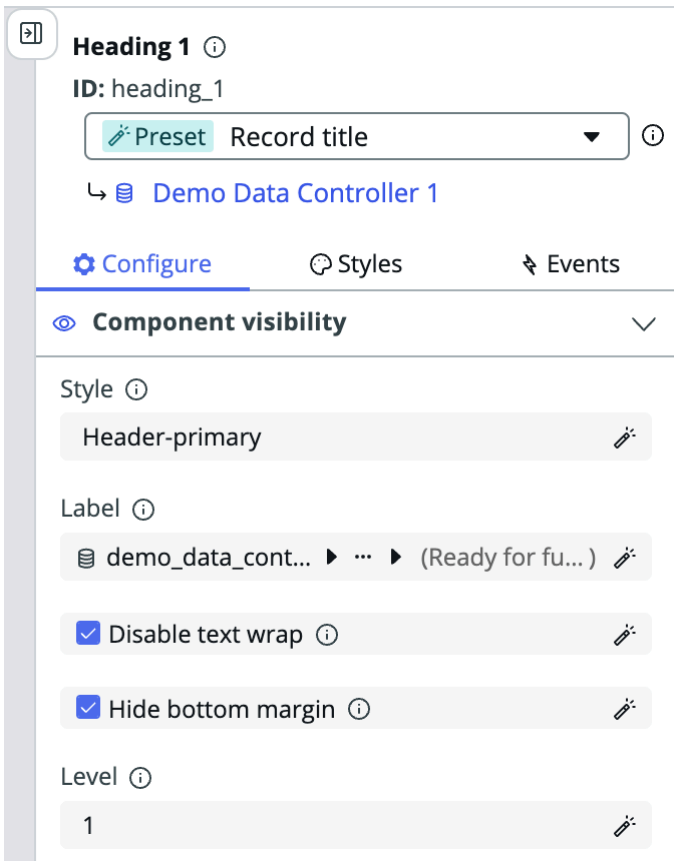


- Work in the UI Builder staging area to build and modify your pages.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

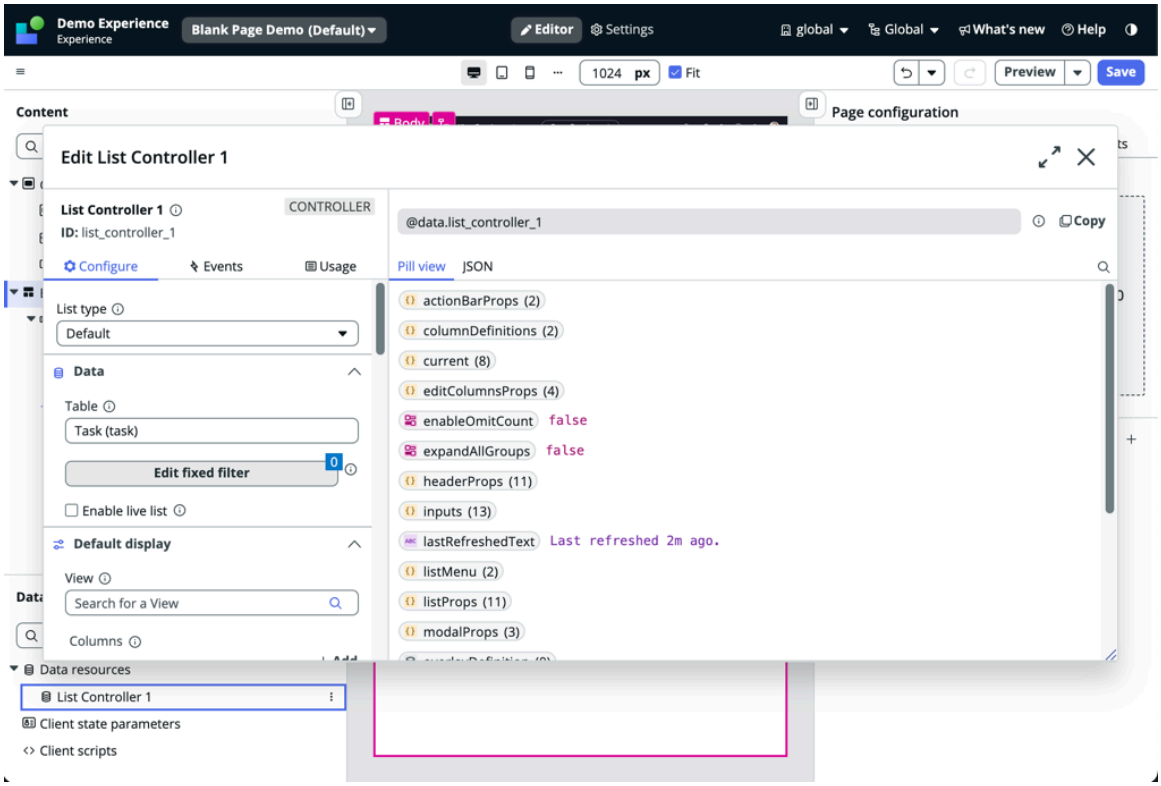


- Go to configuration panel to modify column layouts, columns, components, styling, and events.

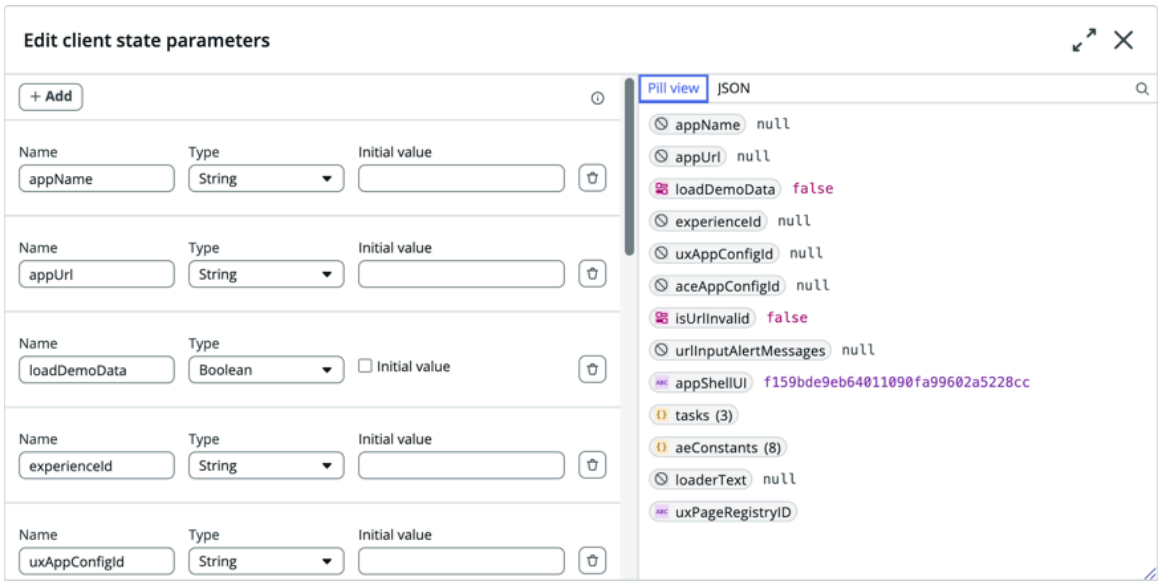


- Data resources: Bind data to your components using data resources to dynamically expose your data from tables, records, or other elements on your page. Data resources enable you to reuse your components. See Connect data to your components for more information.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

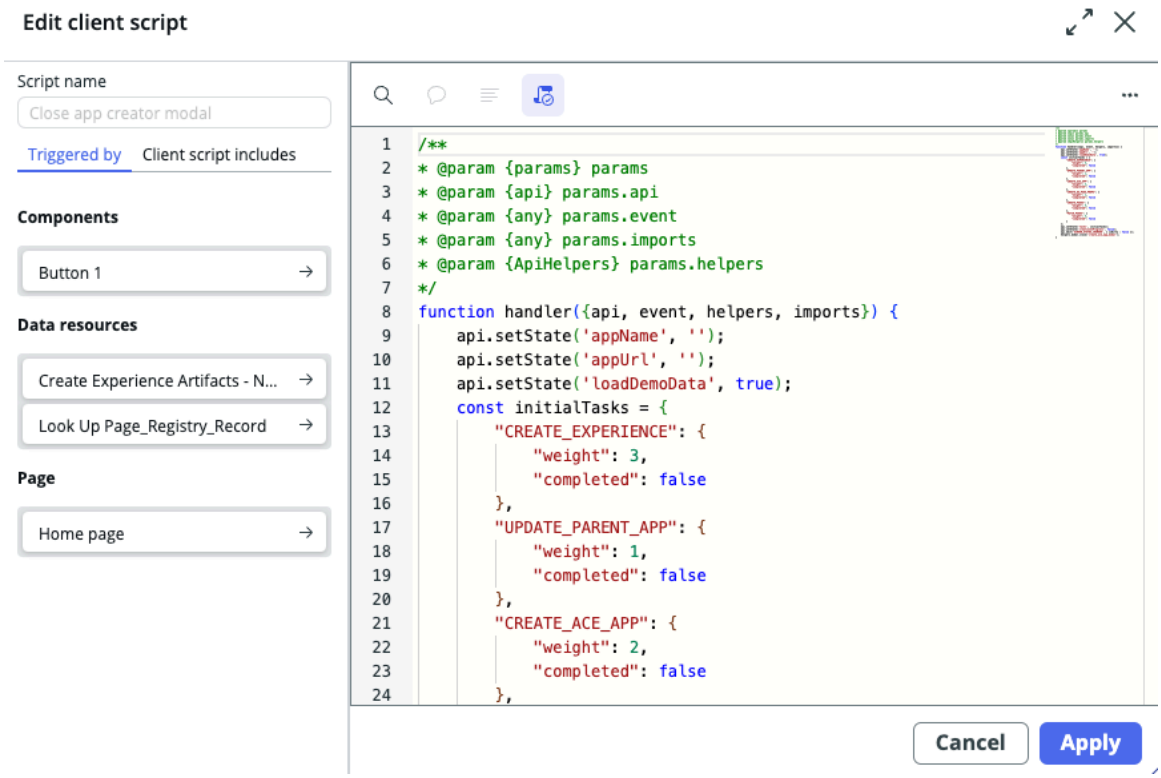


- Client state: Set the parameters for the client state for your page. The client state parameters consist of a name, a type, and an initial value. You can see the preview of the client state parameters.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Client scripts: Scripting automates aspects of the page, like event handling.

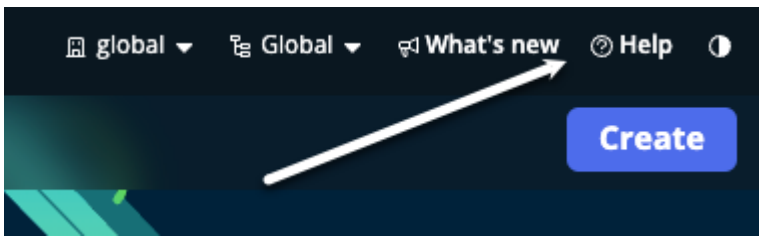


UI Builder Learning Center

Explore the UI Builder Learning Center for articles, videos, courses, and guided tours to help you get started and build knowledge. The Learning Center is a one-stop shop for learning fundamental features and basic guidelines in UI Builder.

Open the Learning Center

Navigate to **All** > **Now Experience Framework** > **UI Builder** > **Help**.



The Learning Center opens in an overlay.

Available information is listed on tiles. Each tile contains a title, the information type, and the time to complete. For example, a tile linking to a product documentation article that takes three minutes to read.

Select a tile to open a resource. Resources are in the following locations:

- Articles: https://www.servicenow.com/docs
- Videos: created by ServiceNow and available on public, online video platforms
- Courses: https://learning.servicenow.com
- Tours: embedded in your ServiceNow instance

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Guided tours in UI Builder

Learn about UI Builder guided tours, including how to use them to build your knowledge and practice using UI Builder.

What guided tours are

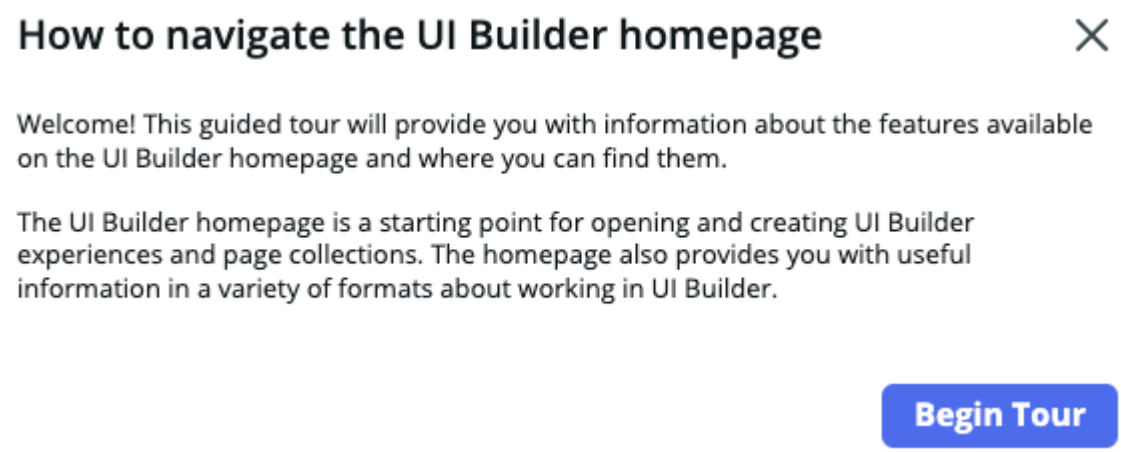
Guided tours are part of the default ServiceNow platform. There are guided tours available to take and you can use the Guided Tours application to create custom tours yourself in your own instances.

Tours contain interactive steps to help train and teach users. Some tours show various features in the user interface, such as an overview of a homepage. Other tours help complete a task, such as creating an experience or previewing an experience page. For more information about what guided tours are and how to create them, see Exploring Guided Tours.

How to access and use UI Builder guided tours

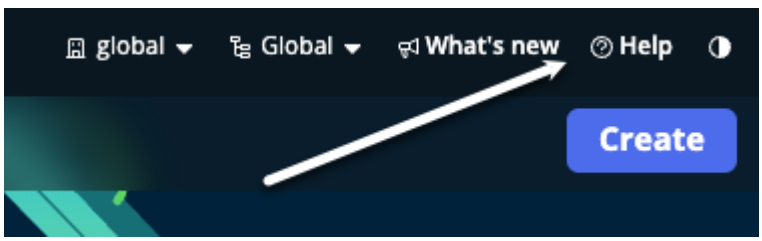
Guided tours about UI Builder may launch automatically. You can also start them manually from the **Getting started** section of the UI Builder homepage or the UI Builder Help panel.

When a guided tour starts automatically, an introduction is displayed. Select **Begin Tour**.



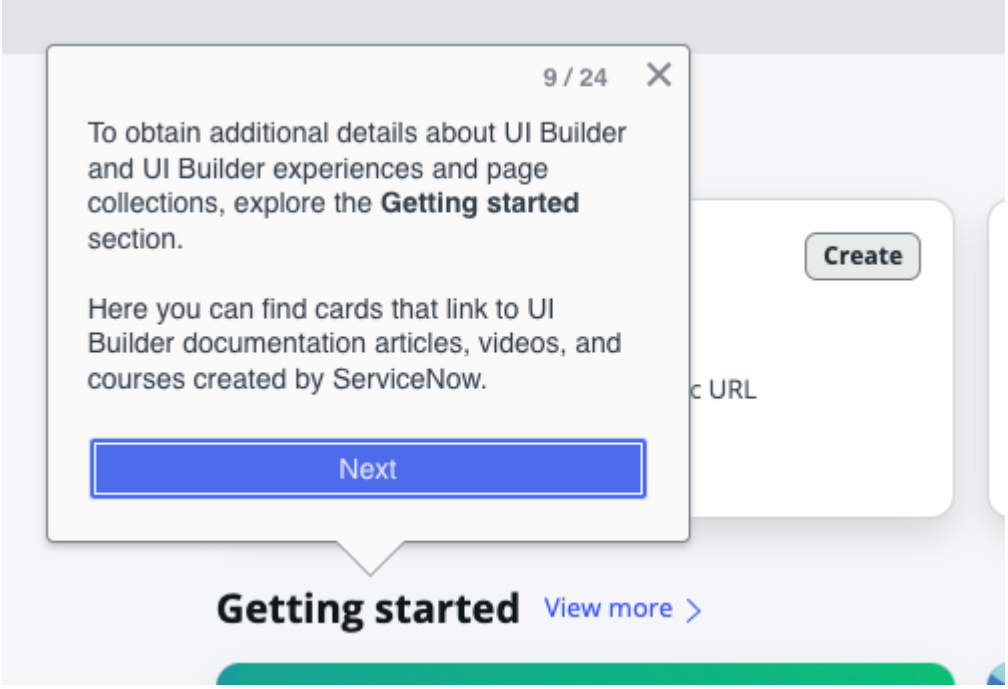
To begin a tour manually from the homepage, navigate to **All** > **Now Experience Framework** > **UI Builder** and select a tour in the **Getting started** section. If no guided tours appear by default in the section, select **View more** and then select a tour on the **Find relevant resources to get you unstuck** pop-up.

Alternatively, select **Help** and then select a tour on the **Find relevant resources to get you unstuck** pop-up.



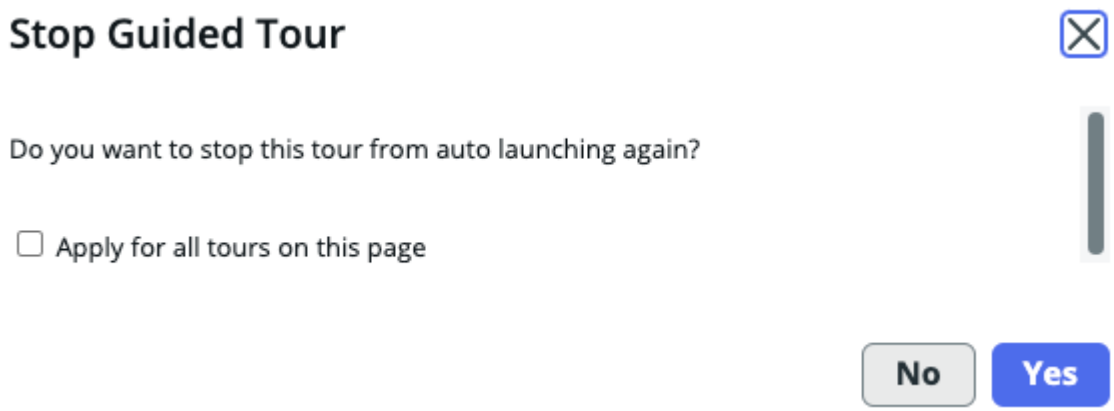
After a tour starts automatically or manually, you're led through a series of steps. Read the text for each step and perform the required actions or select **Next** to continue the tour.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



To stop a tour at any time, select the **X**.

Select **Yes** or **No** to stop the tour from auto-launching again. You can also apply your decision to all tours that start on the same page.



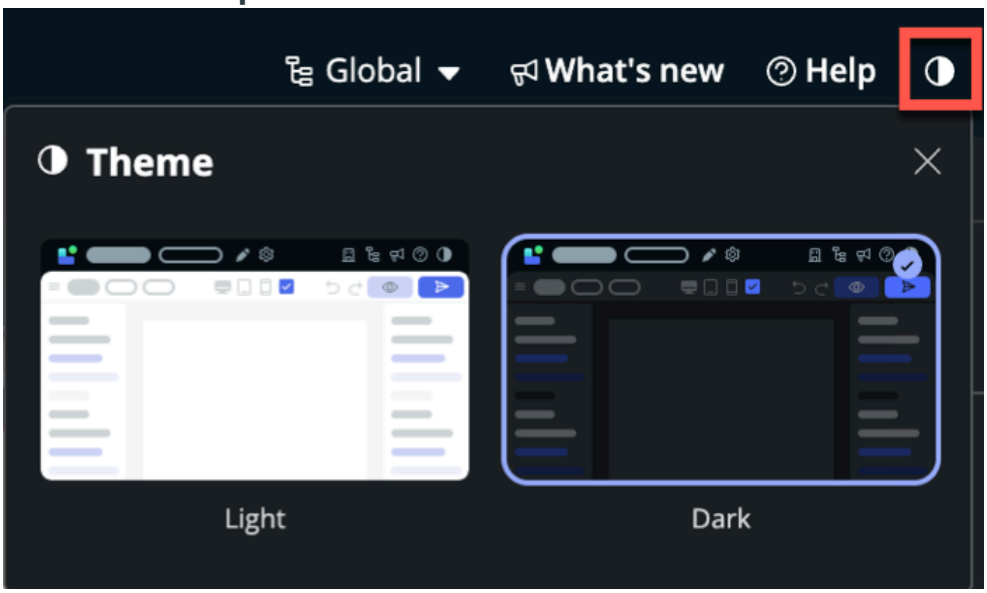
Dark theme in UI Builder

Learn how to switch to the dark theme in UI Builder.

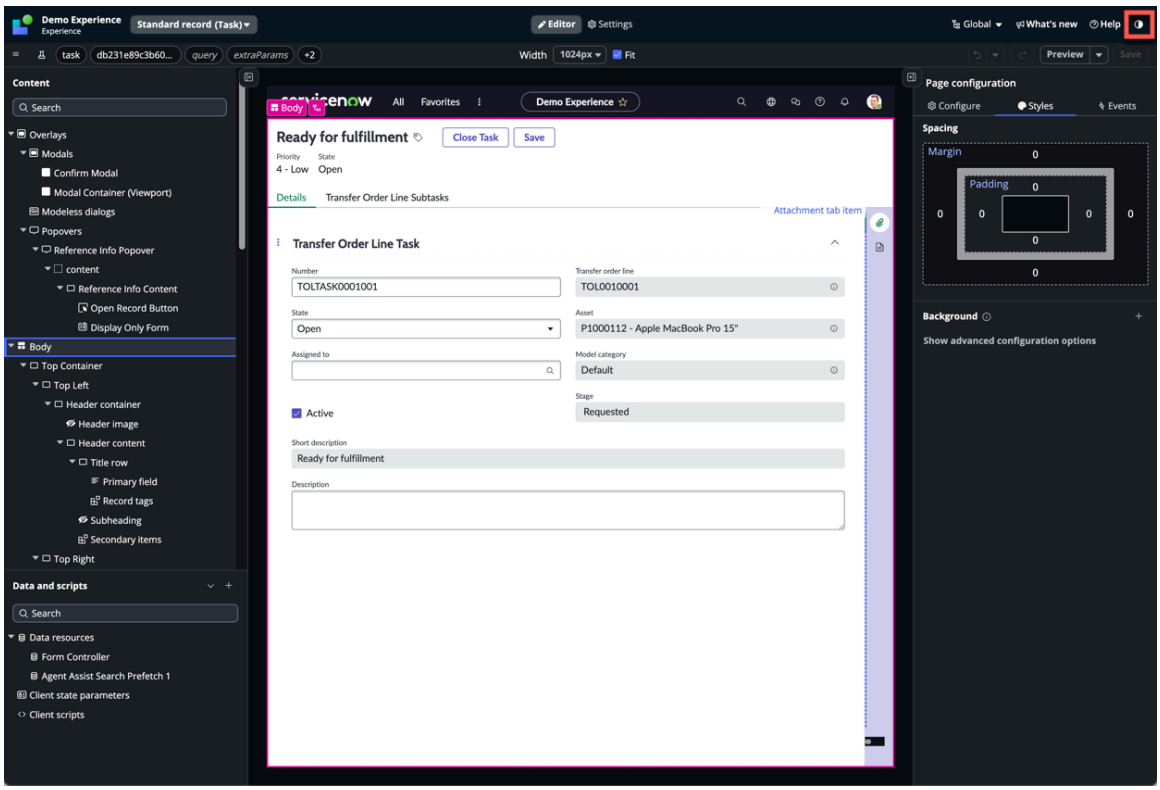
UI Builder includes a dark theme option, offering an eye-friendly alternative to the default light theme. You can enable dark theme by selecting the Choose theme icon ( ) in the banner.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

UI Builder theme options



The Dark theme is instantly applied to the UI Builder interface.



Note: The UI Builder stage uses the theme set in your Platform theme settings. For example, the stage might remain white even when the rest of the interface is dark. To enable UI16 pages to render with the dark theme on the stage, select the **Dark** option in the theme menu of your user preferences. For more information on navigating to the theme menu, see Open the theme menu.

Learn UI Builder by example

Exploring common use cases can be a great way to learn how to use UI Builder. Consider performing each of the tasks in this section in the order presented.

The following use cases are described here.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Create a demo experience to explore UI Builder

An experience in UI Builder is a collection of web pages for users to interact with an application. The experience includes routes, page variants, and the audience and conditions required for each variant, as well as experience settings.

Before you begin

Role required: ui_builder_admin

About this task

The UI Builder experience view is a central place to view and understand the structure and details of an experience. Use the experience view to see the structure and hierarchy of your experience. You can access UI Builder experience view by selecting an existing experience from the UI Builder home screen or by creating a demo experience, as described here.

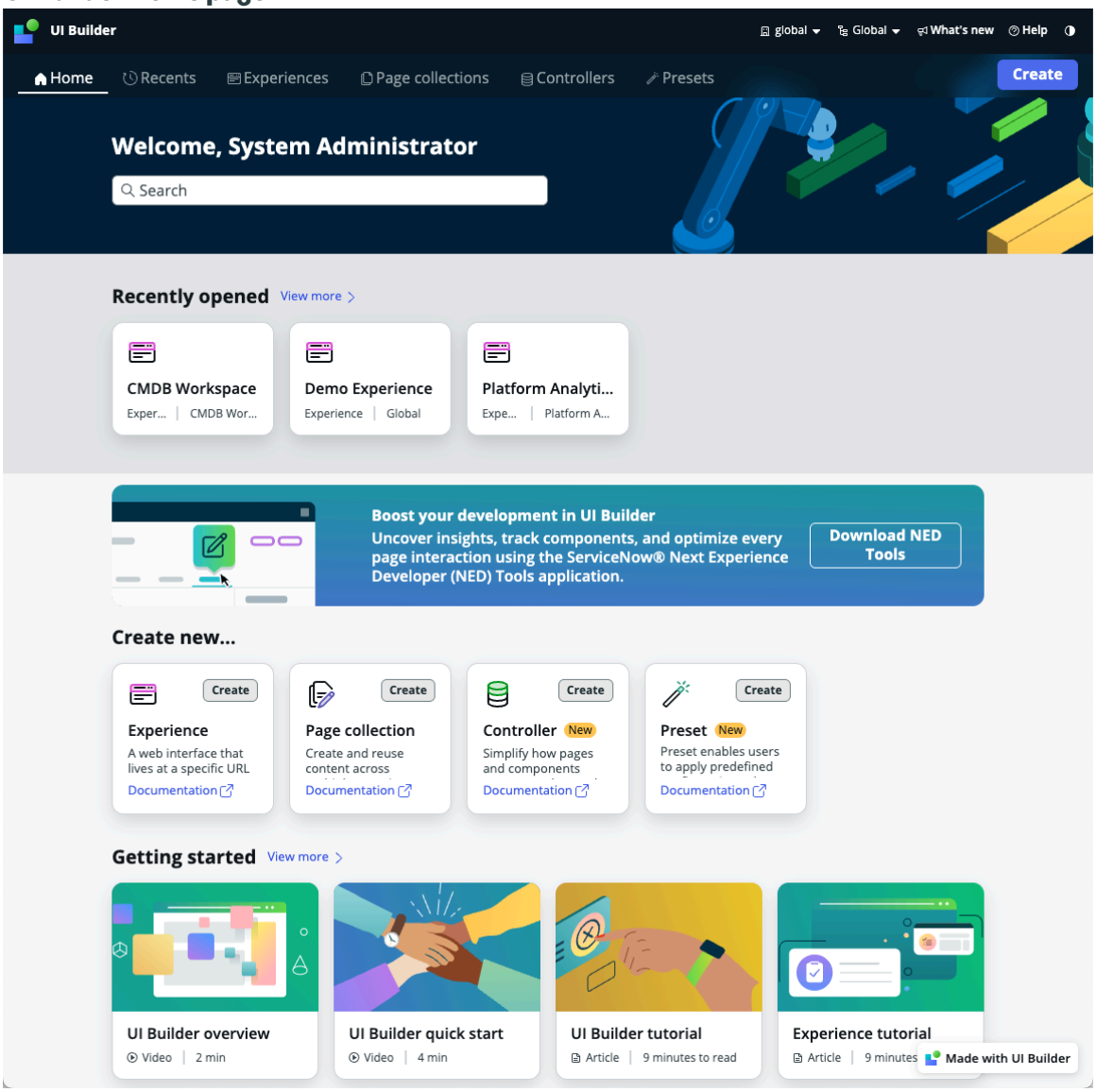
Note: UI Builder is not yet capable of building or configuring ServiceNow base system service portals, such as the Employee Center. For service portals, continue to use the Service Portal Designer.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.

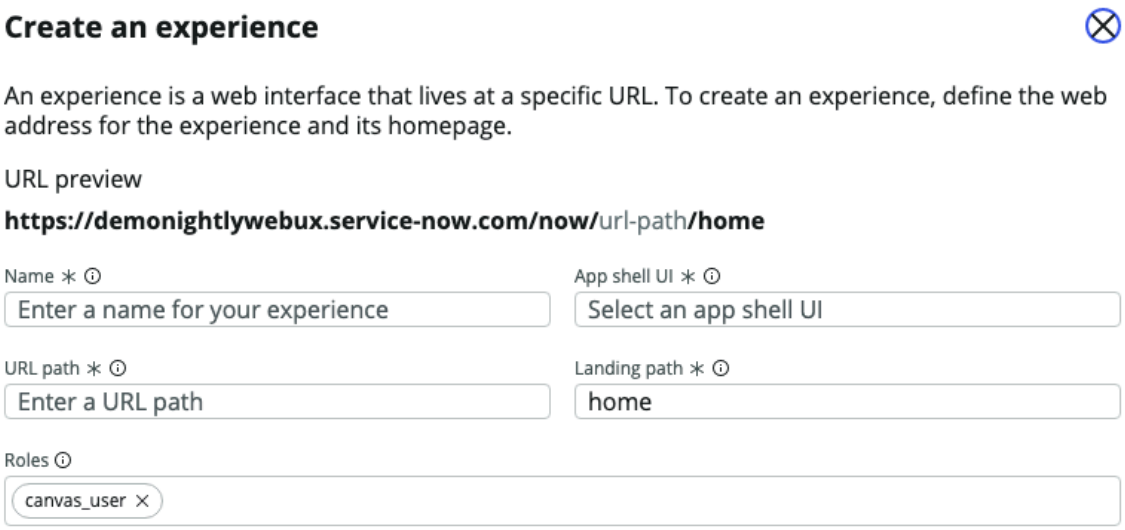
UI Builder Home page



2. From the UI Builder Home page, select **Create**.
3. Select **Experience**

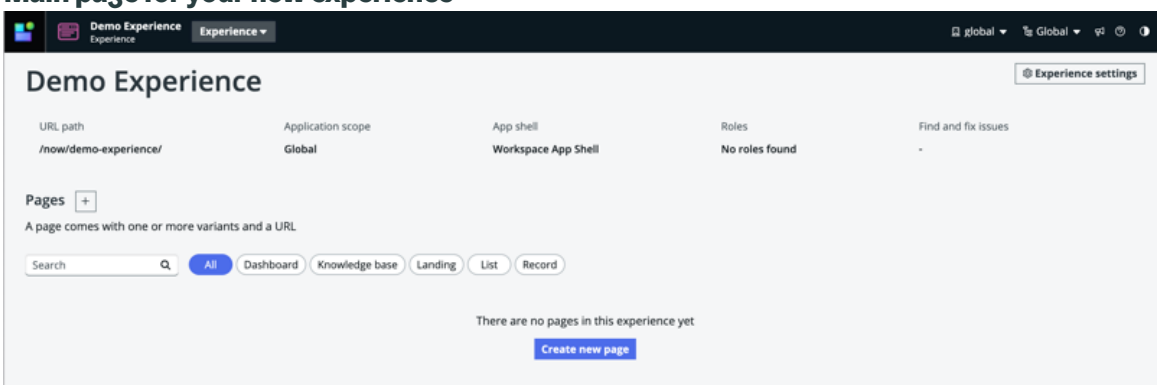
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Create an experience dialog box



4. Enter a **Name**; for example, **Demo Experience**.
5. Select an **App shell UI**; for example, **Workspace App Shell**.
6. Select **Create**.
7. Select **Open experience** to view the main page for your experience.

Main page for your new experience



Note: In addition to the name of your new experience, the screen includes the URL, application scope, admin panel, and the roles of users who can view the experience. Any pages or variants you create for the experience appear in the **Pages** section.

What to do next

Select the **Next topic** link to learn how to create a blank page for this experience.

Create a blank page

After creating a demo experience, you can create a blank page that you can then build on to define what the experience provides to your users.

This video shows you how to perform the following procedure.

Before you begin

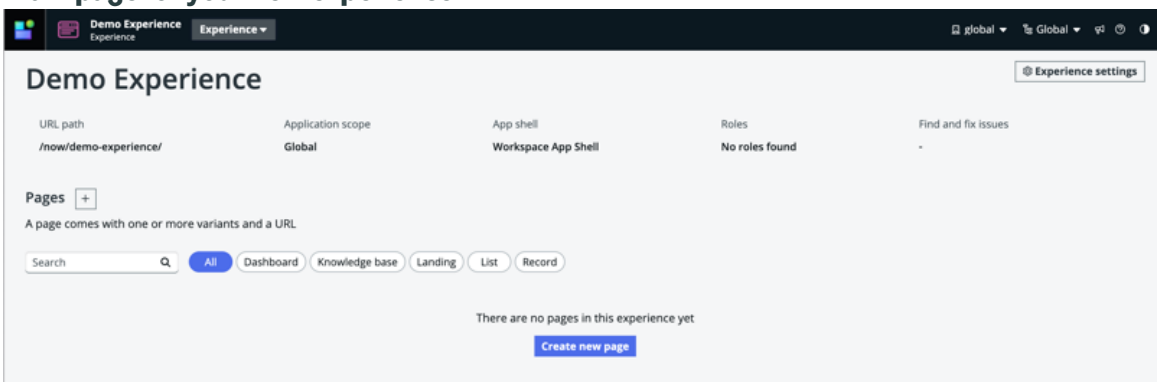
Role required: ui_builder_admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. If you aren't already viewing the experience you created in the previous procedure, navigate to **All** > **Now Experience Framework** > **UI Builder**, and select the experience you created.

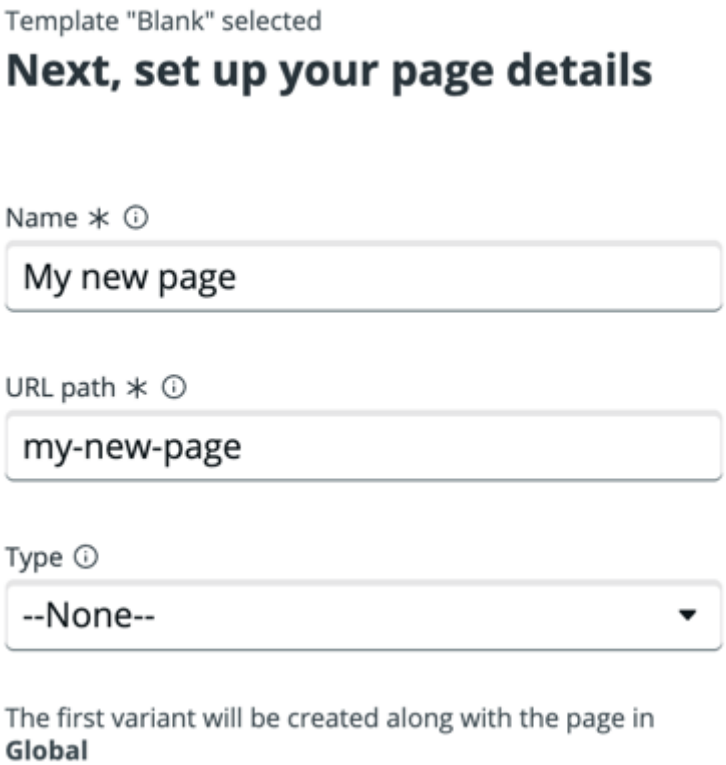
Main page for your new experience



2. From the main page for your experience, either select **Create new page** or select the plus (**+**) sign next to **Pages and variants**.
3. On the **First, select a template** screen, select **Create from scratch instead**.

   There are templates you can use, but we are going to create a page from scratch.

Page details



4. In the **Name** field, type **My test page**, and select **Continue**.

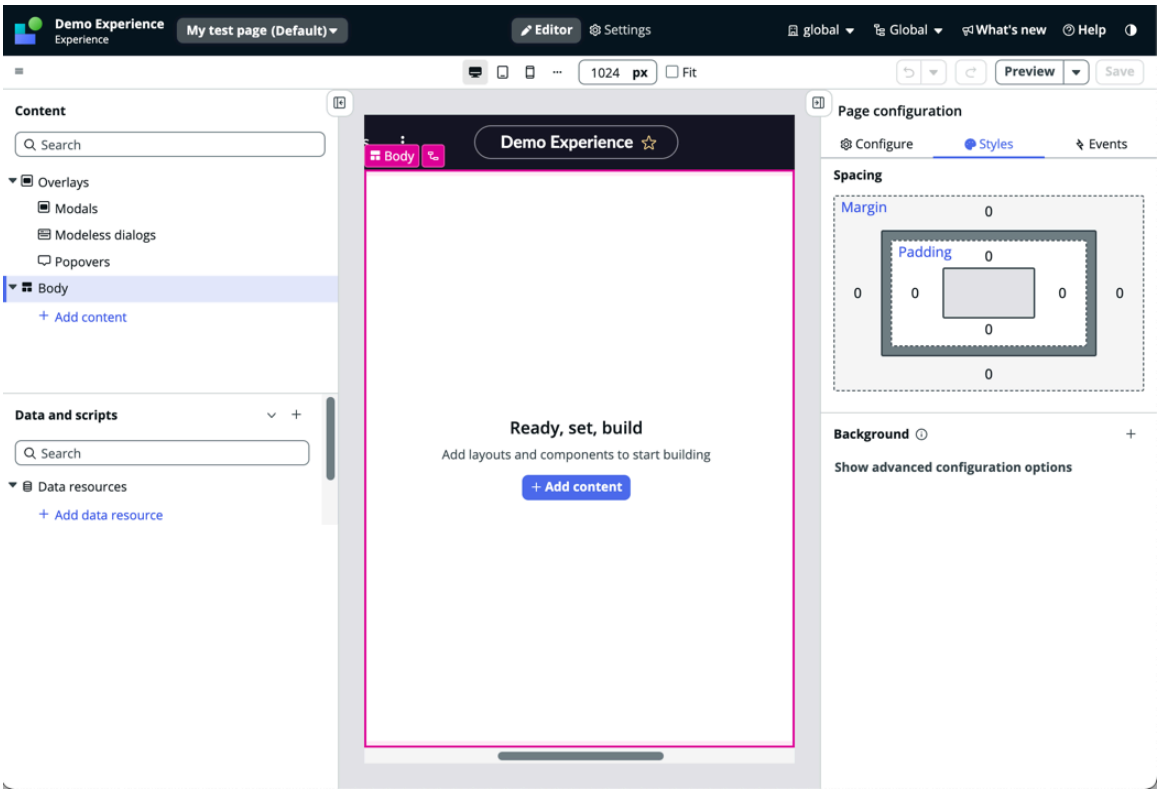
   Note: The **URL path** shows the URL that users can navigate to access the page.

5. Select a URL page type in the **Type** drop-down list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

6. Select **Looks good** on the next screen.
7. On the **Tell us about your variant** screen, do not make any changes.
8. Select **Continue**.
9. Select the **Build responsive** option (default) for greater control of how the page appears at different screen widths.
10. Select **Create**.
    The visual editor opens. Here, you can begin editing your new page.

UI Builder visual editor



What to do next

Select the **Next topic** link to learn how to create a button that opens a modal.

Create a record page using a template

After you've created your demo experience, you can create a record page from a template. A record page shows data from a table.

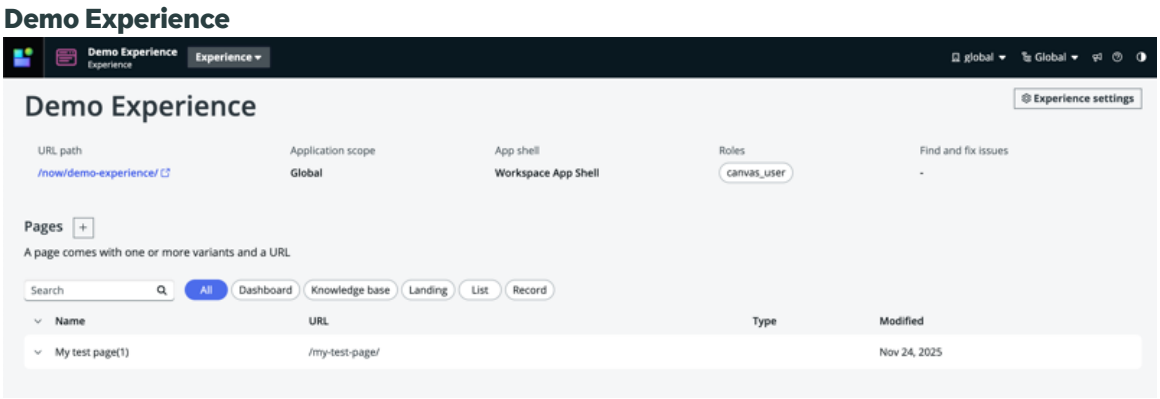
Before you begin

Role required: ui_builder_admin

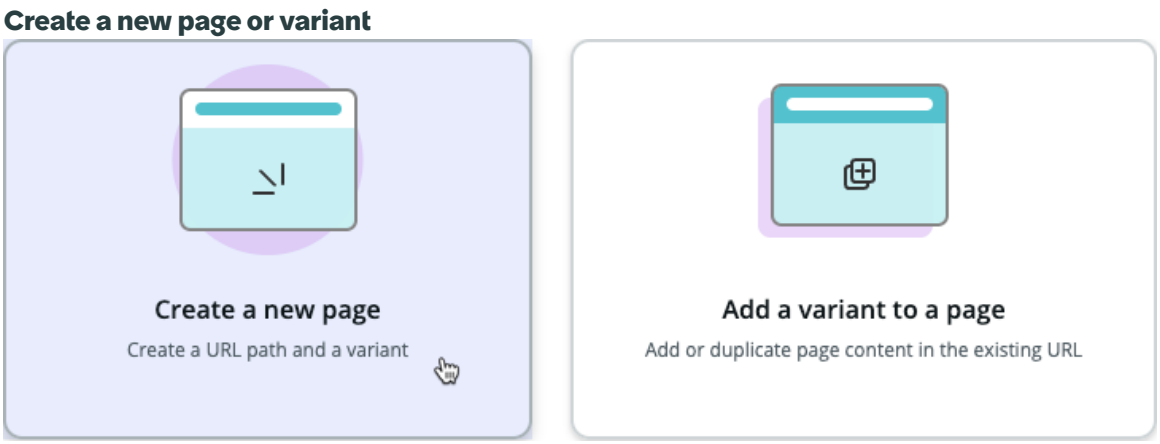
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Open the main page for your demo experience.



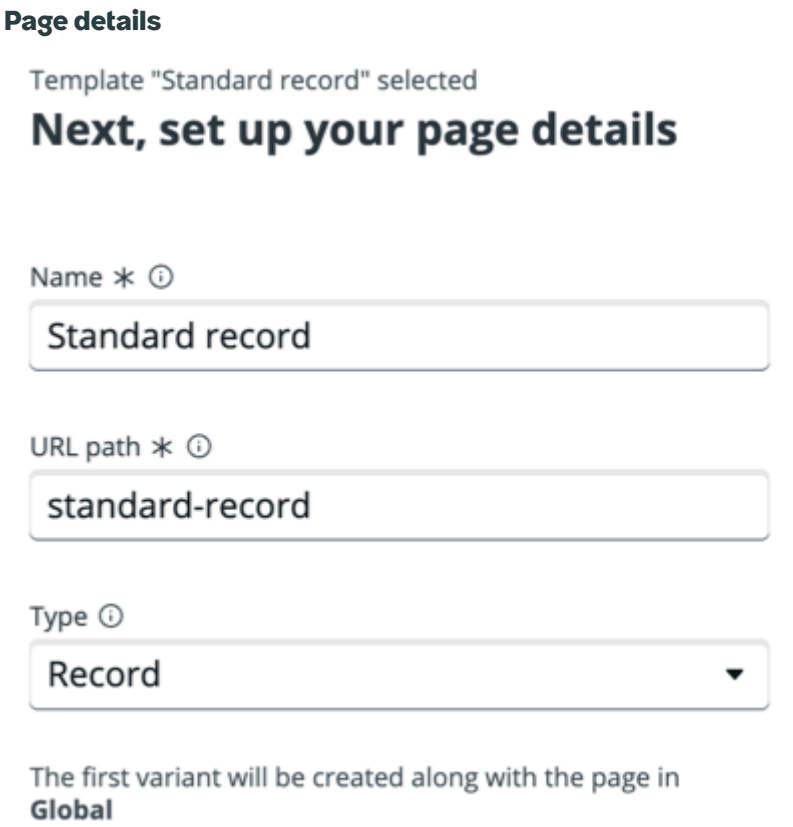
2. From the main page for your experience, select the plus (**+**) sign next to **Pages**.



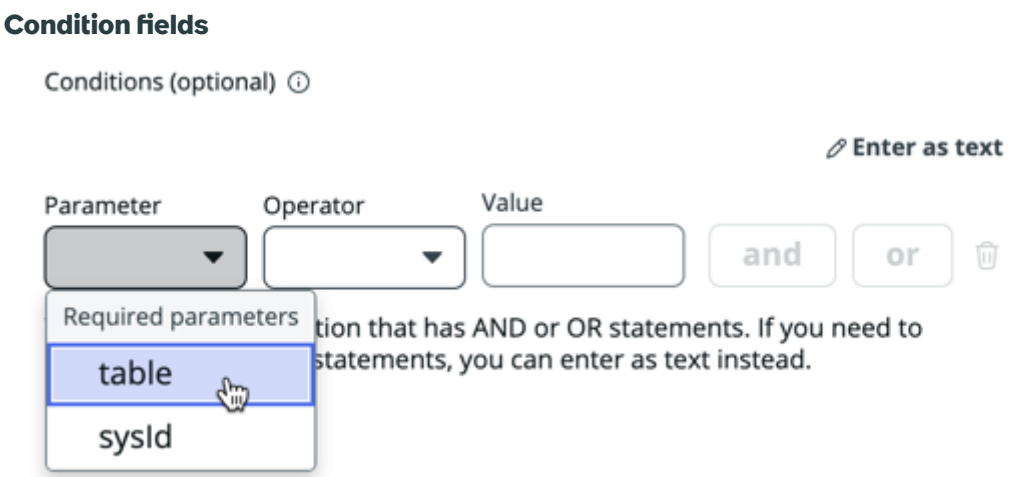
3. Select **Create a new page** to create a page that resides at a different URL.
4. On the **First, select a template** screen, locate the **Standard record** template and hover over **Use template**.

> **Note:** Optionally, you could select **Learn more** to read about the template before selecting it.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



5. In the **Name** field of the Page details screen, type `Task record page`, and select **Continue**.
6. Select a URL page type in the **Type** drop-down list. UI Builder automatically assigns a page type based on the template you select, but you can change this value if needed.
7. Select **Looks good** on the next screen.
8. On the **Tell us about your variant** screen, enter a condition to determine when the page is visible.
   a. In the **Parameter** field, select **table**.
   b. In the **Operator** field, select **is**.
   c. In the **Value** field, enter `task`.



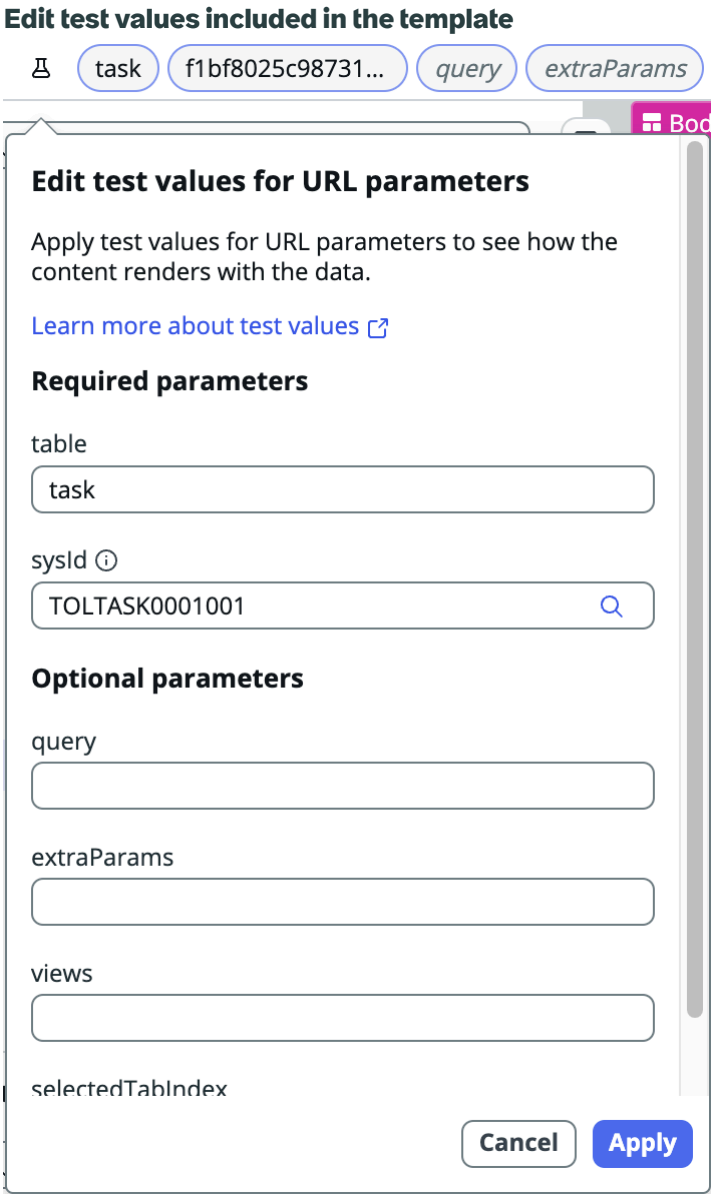
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

This page is visible to users accessing a record from the Task table.

9. Select **Create**.
   The visual editor opens. Here, you can begin editing your new page.

   Note: The new page includes LOTS of layout, preconfigured components, data, and modals. Additionally, it includes test values you can modify for your own particular use cases. Using templates can be a significant time-saver when creating new pages.

10. You can select the area above the Page content pane to view information about test values included in the template.



What to do next

Select the **Next topic** link to learn how to define audiences who can view your pages in UI Builder.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Create a button that opens a modal

After you've created your demo experience and added a blank page, you can edit the page variant as needed. For the sake of this demo, you can create a button and a modal, and configure the button to open the modal.

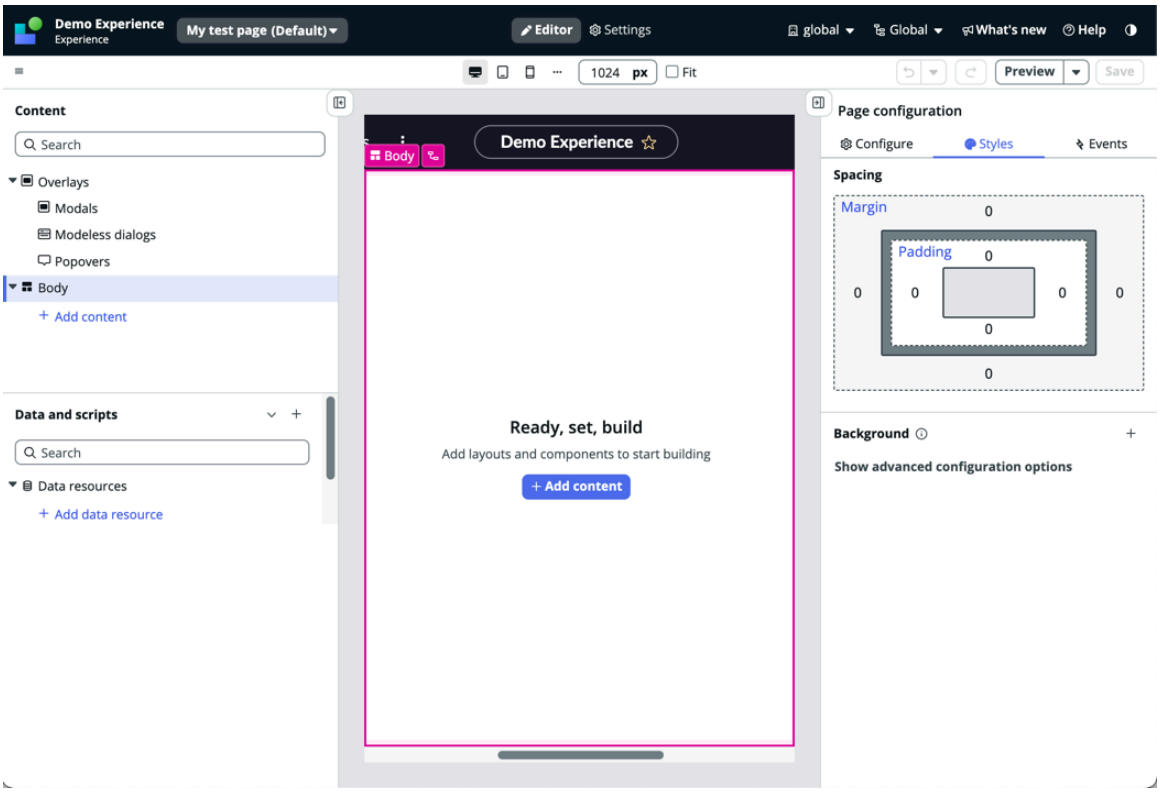
Before you begin

Role required: ui_builder_admin

Procedure

1. Open the UI Builder page for your demo experience.

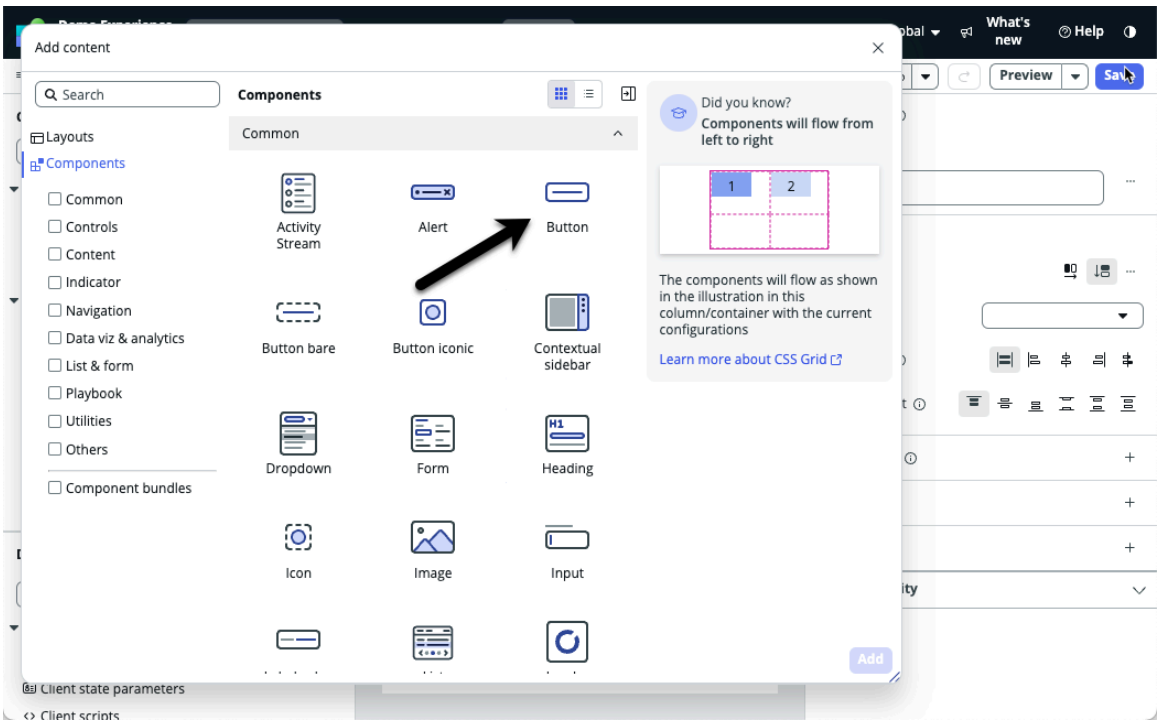
UI Builder visual editor



2. Click the **+ Add content** button on the stage to open the toolbox.
3. Select a **Single column** layout.
4. Next, click the **+** icon in the column to open the toolbox.
5. Select the **Button** component to add it to the stage.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add a button

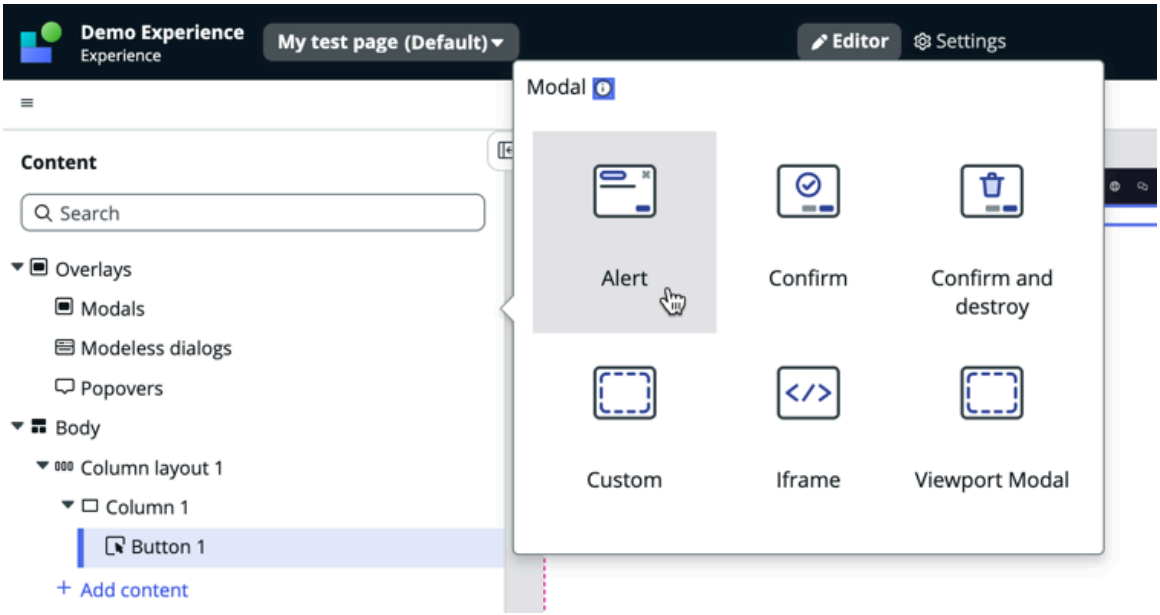


Note: When you have selected the component, the Page configuration pane includes some presets you can use to automatically configure components on compatible pages. For the sake of this exercise, however, you will be configuring the component manually. For more information on presets, see Customize UI Builder pages using components.

6. Select **Add**.
7. In the Page configuration pane, select **Configure the component manually**.
8. In the Page content pane, select **Button 1** and, in the Configuration pane, change the button label to **Open modal**.
9. Select **Save**.
10. In the Page content pane, click the plus icon next to **Modals** and select an **Alert** modal.

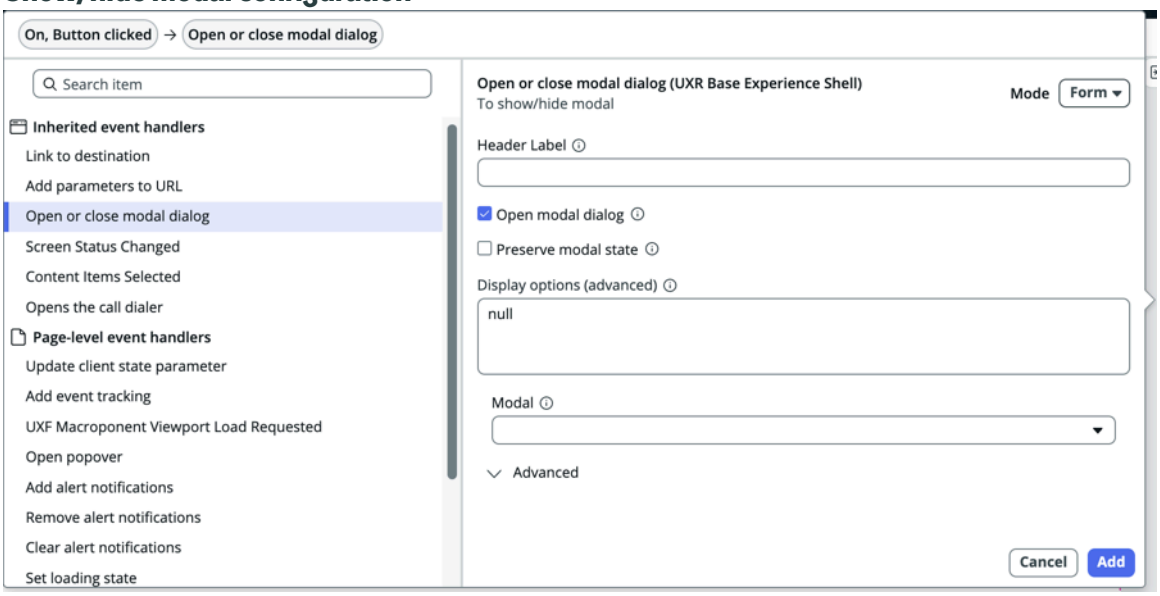
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add an Alert modal



11. Select **Save**.
12. In the Page contents pane, select **Button 1** and, in the Configuration pane, select the **Events** tab.
13. Select **+ Add event handler** and, under **Inherited event handlers**, select **Open or close modal dialog**.

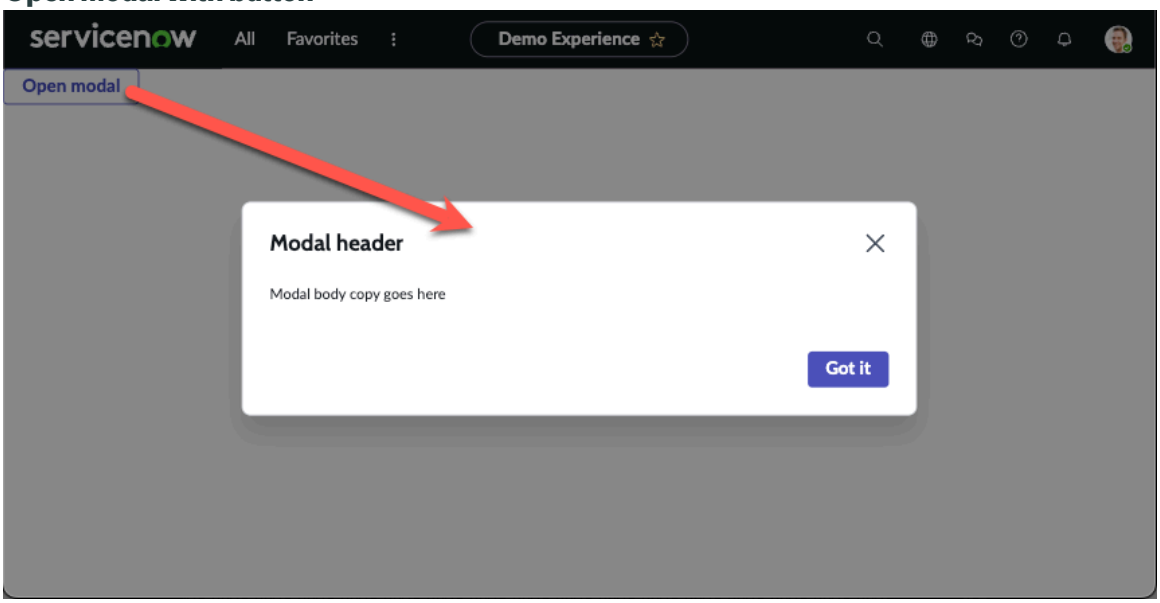
Show/hide modal configuration



14. Activate the **Open modal dialog** and, select **Alert 1** in the **Modal** field, and select **Add**.
15. Select **Save**.
16. Select **Preview**.
17. When the preview opens, select **Open modal**.
    The modal you defined opens.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Open modal with button



18. Select **Got it** in the modal, and then select the browser back button to return to the experience main screen.

What to do next

Select the **Next topic** link to learn how to create a page using a template.

Define an audience for your variant

An audience represents a group of users in your organization. You can define who can access this page variant by adding one or more predefined audiences.

Before you begin

Role required: ui_builder_admin

About this task

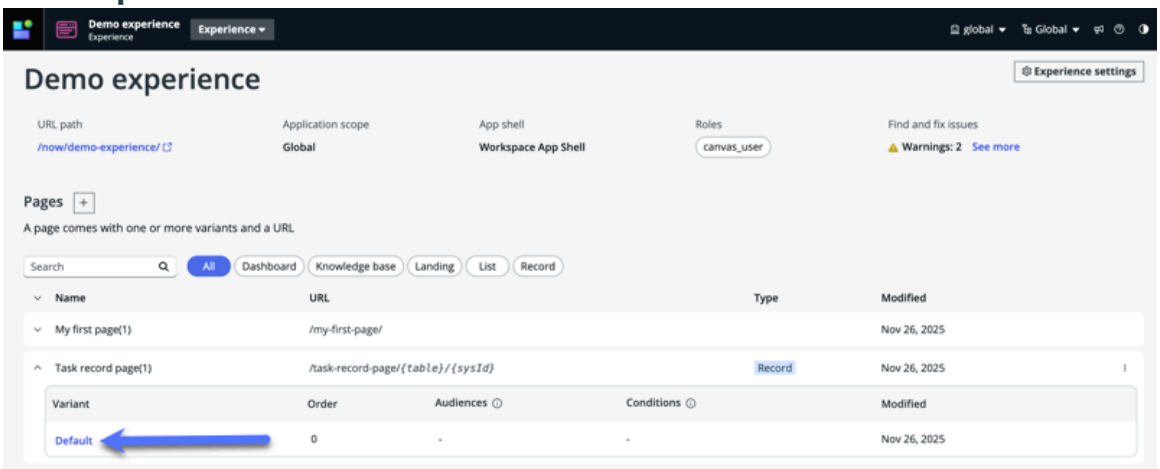
In the previous procedure, you created a page variant that can be viewed by anyone accessing a record from the Task table. For this procedure, you will duplicate the variant and configure it so that it is accessible only to audiences with the Admin role.

Procedure

1. Open the main page for your demo experience.
2. Select the experience you created, expand a page and select **Default**.

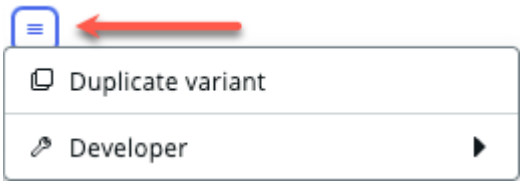
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Demo Experience



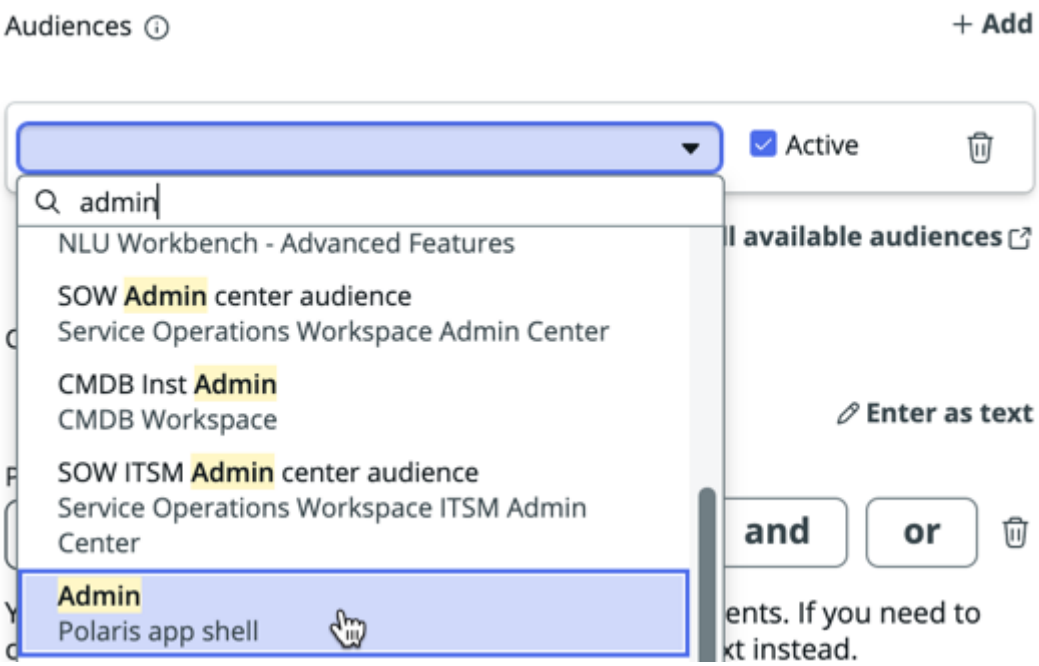
3. Select the **Open menu** icon.

Open menu



4. Select **Duplicate variant**.
5. On the **Tell us about your variant** screen, enter **Admin only** in the **Name** field.
6. Select **+ Add** next to **Audiences**, and select the **Admin** audience.

Select the Admin audience



7. Select **Create**.
   The variant is duplicated and only users with the Admin role can view this page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

What to do next

Select the **Next topic** link to learn how to apply conditions to the variant so that the variant is visible only when the defined conditions are met.

Define conditions for your variant

You can define conditions to determine when a page variant is shown. The conditions are based on setting an order, and declaring the criteria that must be met for the page variant to display.

This video shows you how to perform the following procedure.

Before you begin

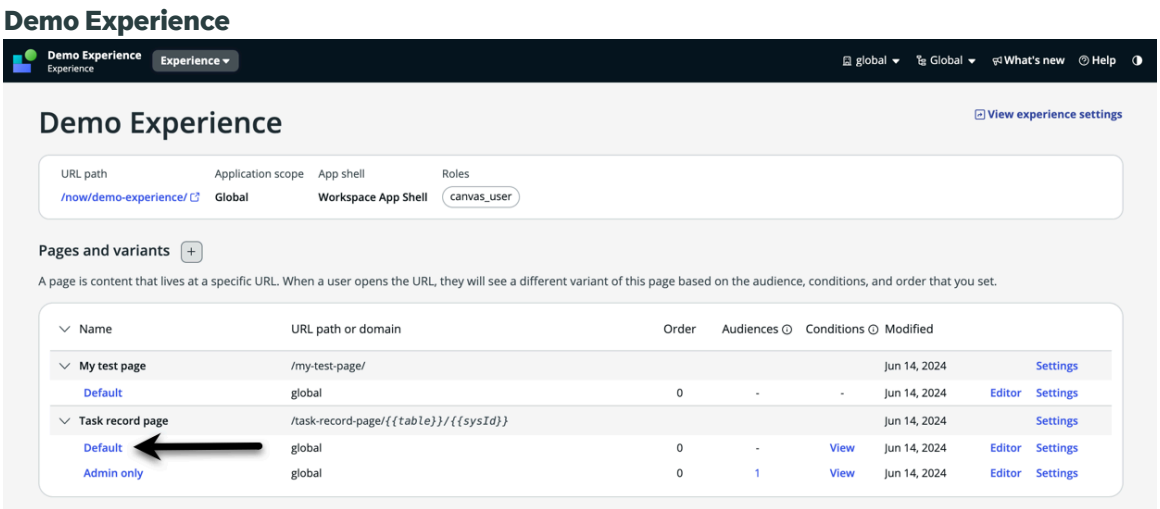
Role required: ui_builder_admin

About this task

If you have multiple page variants that all have the same conditions, the variants go by the order setting.

Procedure

1. Open the main page for your demo experience.
2. Select the experience you created, and select **Default**.



3. On the Default variant under the **All users record page**, select the **Menu** icon (⋮), and select **Duplicate variant**.
4. On the **Tell us about your variant** screen, enter **Incident record page** in the **Name** field.
5. Since the intent of this task is to restrict this page to users who are opening a record in the Incident table, enter the following under Conditions:

   a. In the **Parameter** field, select **table**.

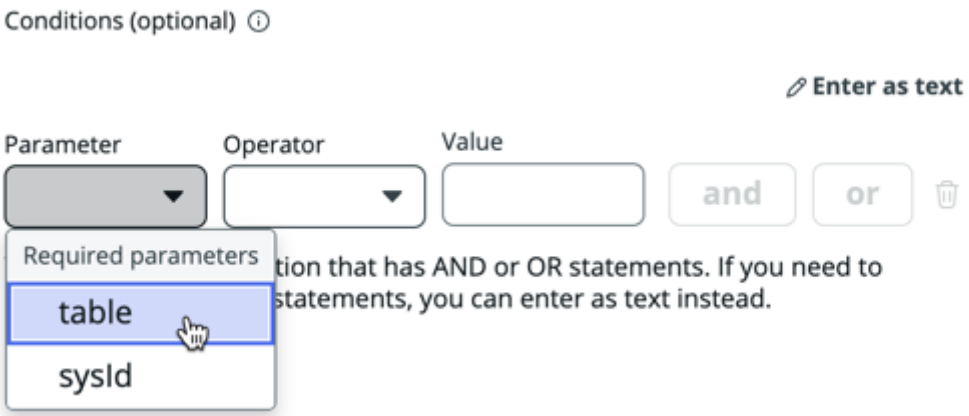
   b. In the **Operator** field, select **is**.

   c. In the **Value** field, replace the text with `incident`.

      When you created this page earlier in the series, you set this condition to the Task table. Now, you are configuring the page to display only to users accessing a record from the Incident table.

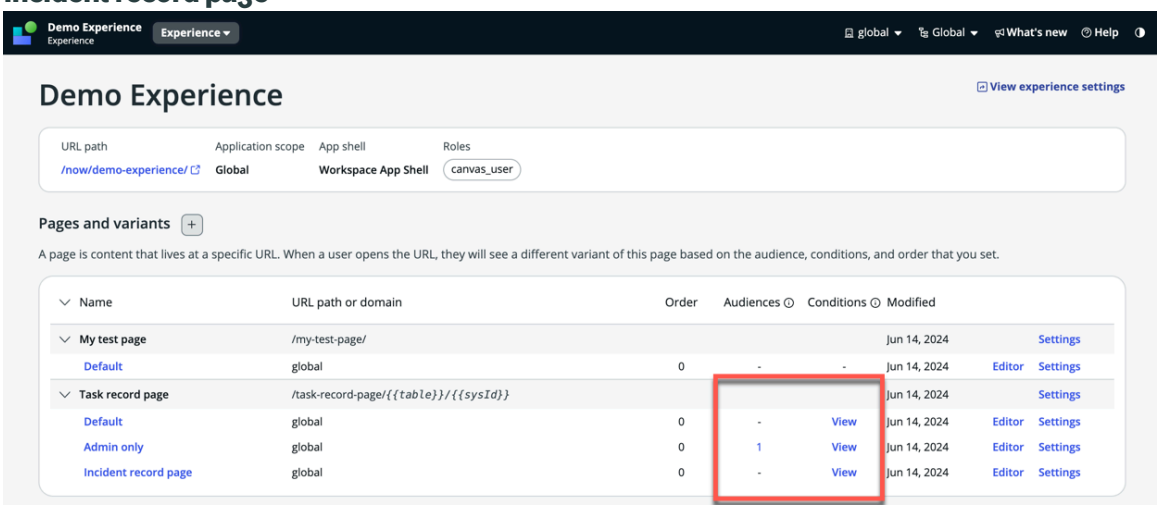
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Condition fields



6. Select **Create**.

Incident record page



Note: You can view the structure of your pages and variants in this experience. The **Conditions** column shows a **View** link for each page. You can select **View** to see the condition for that variant (that is, table=incident). The **Audiences** column for the Admin only variant shows a 1. You can select the **1** to view the role required to view that variant.

Customize forms within a form component

Customize your form components by accessing Form Builder in UI Builder.

Before you begin

Role required: admin

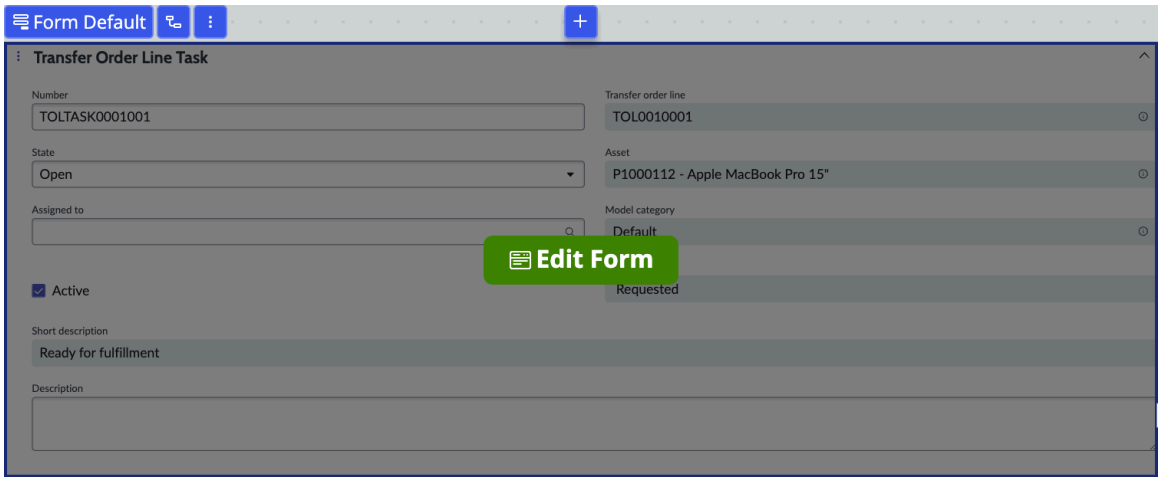
About this task

You can edit form components without leaving UI Builder. Access Form Builder from within a form component on the UI Builder stage. After you save the form, your edits remain, and anyone who views the form sees the modified version. Changes made in Form Builder are reflected on the stage.

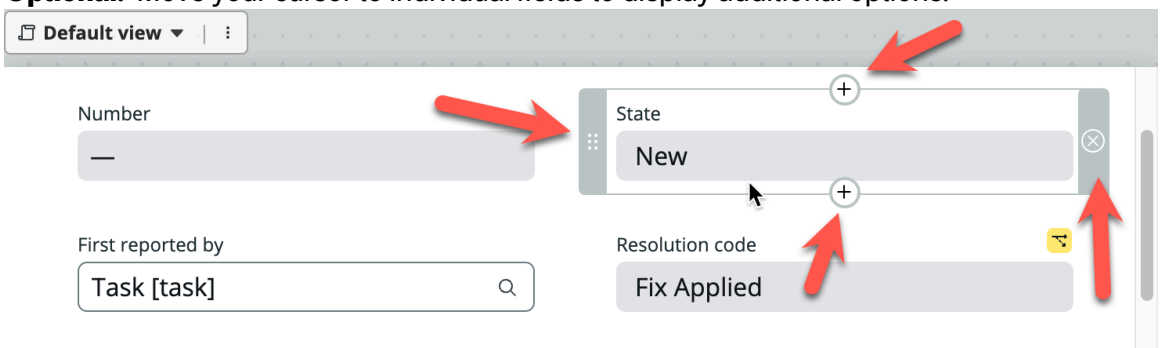
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Add a form component to your page.
   a. In the content tree, select **+ Add content**.
   b. Select **Form**.
2. On the stage, move your cursor to the form component to display the **Edit Form** button.



3. Select the **Edit Form** button.
   Form Builder opens in a full-screen modal.
4. **Optional:** Move your cursor to individual fields to display additional options.



5. **Optional:** Insert fields by moving your cursor to the **+** button, which displays the **+ Add** button, then selecting **+ Add**.
   You can also drag additional fields onto the form from the **Fields** column.

   Example
   For example, you can add the **Due date** field below the **Configuration item** field, then select **Save** and **Preview** to save and preview the form.
6. **Optional:** Move a field by dragging the edge of a field to another place in the form and then selecting **Save**.

   Example
   For example, select the **State** field and place it below the **Duplicate of** field, then select **Save**.
7. **Optional:** Delete a field by selecting the **X** button on the field.
8. Select **Save**.
9. Close Form Builder to see your edits appear on the stage.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

What to do next

You can enhance the functionality of a custom form by configuring additional features. For more information, see Forms in Table Builder.

Learn UI Builder using other ServiceNow resources

Learn more about UI Builder using resources outside of the product.

Learn more about UI Builder	Additional ServiceNow resources
UI Builder is a web user interface builder. Use UI Builder to build pages for CRM Workspace, App Engine Studio generated workspaces and portals, or custom web experiences using Next Experience Components and custom web components.	ServiceNow Dev Program YouTube channel
	Community YouTube channel
	ServiceNow Community site
	Developer Advocate blog
	ServiceNow Developer site
	Components documentation

Learn about audiences

Learn how to apply the correct audiences to your UI Builder pages.

Audiences

Understanding your audiences in UI Builder is an important part of creating pages. The audience defines who can see your pages. You create pages for targeted experiences tailored to audiences and roles such as agents and managers. For example, you may want to create a page for agents to solve issues for your employees. You want to ensure that only people who have the agent role can see the page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Audiences are generally made of allow/deny lists based on role and domain. They can also target the following:

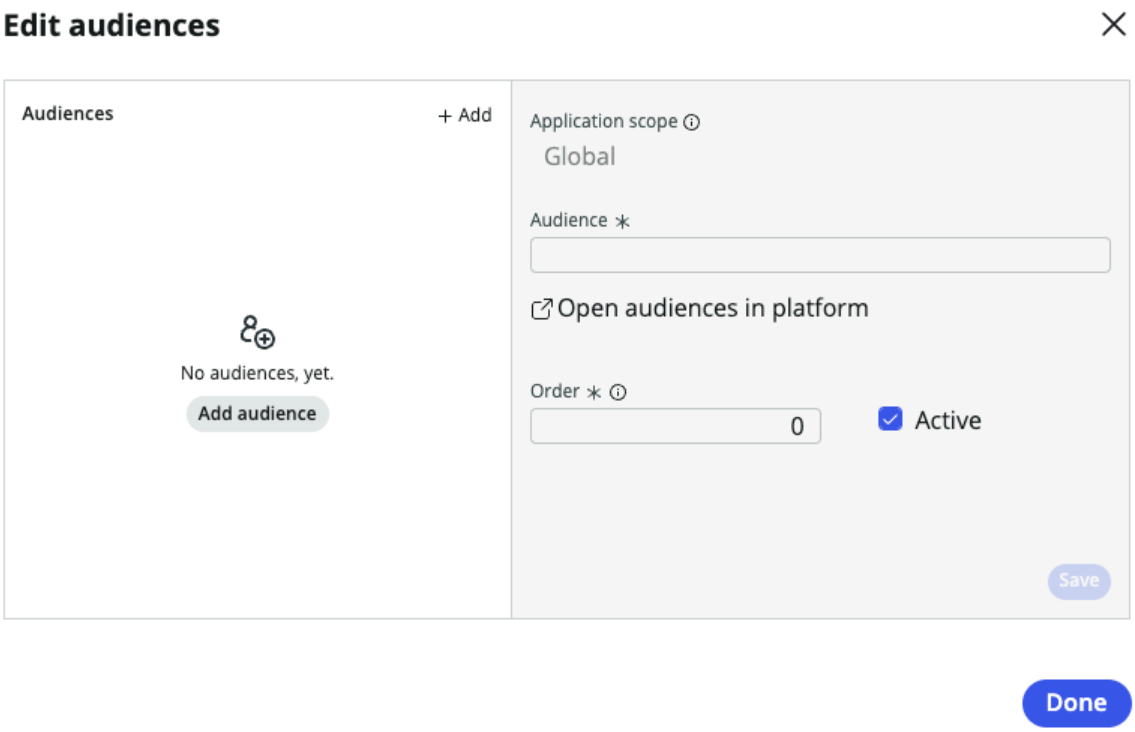
- Role
- Group
- User
- Company
- Department
- Location
- Script

You can set audiences to fit a specific role based on one or more criteria. For example, you could create an audience for an ITSM user in Europe who is not a manager.

The *glide.ux.user_criteria_enabled* property needs to be set to **true** to configure access for users based on role, department, group, location, or company. See Enable the user criteria property, for more information.

Set an audience in UI Builder when you create your page or page variant. You can also set the priority of the audience record. The lower the number the higher the priority. Ensure the **Active** check box is selected to make this page active.

Add audiences modal

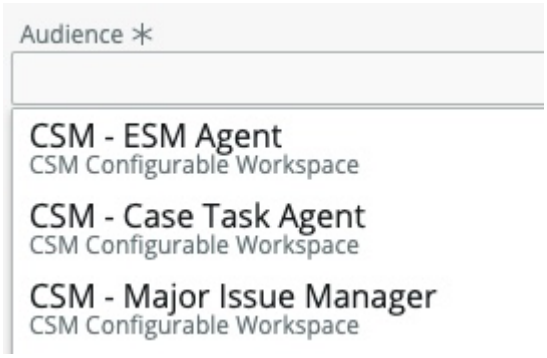


If you do not see an audience that you need, you can click **Open audiences in platform** to define an audience in the ServiceNow AI Platform®.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Choosing an audience from the list in the **Audience** field.

Audience list



- Selecting **Open audiences in platform** to edit or create an audience record in the ServiceNow AI Platform®.
- If you have multiple audiences, define the **Order** for each audience record. The order defines the importance of each audience record. For example, a CSM Manager could have a higher priority than a CSM Consumer Agent. To give higher priority to an audience, enter a lower number.
- Ensuring the **Active** check box is selected to make this page active.

Learn about security and roles

Set up the security and roles for your UI Builder instance.

Security and roles in UI Builder are controlled through your application scope, domain separation, and protection policy settings.

Roles

Roles control access to features and capabilities in UI Builder. The admin role provides access to all features and capabilities. After access has been granted to a role, all the groups or users assigned to the role are granted the access. Roles can contain other roles, and any access granted to a role is granted to any role that contains it. For more information, see Managing roles.

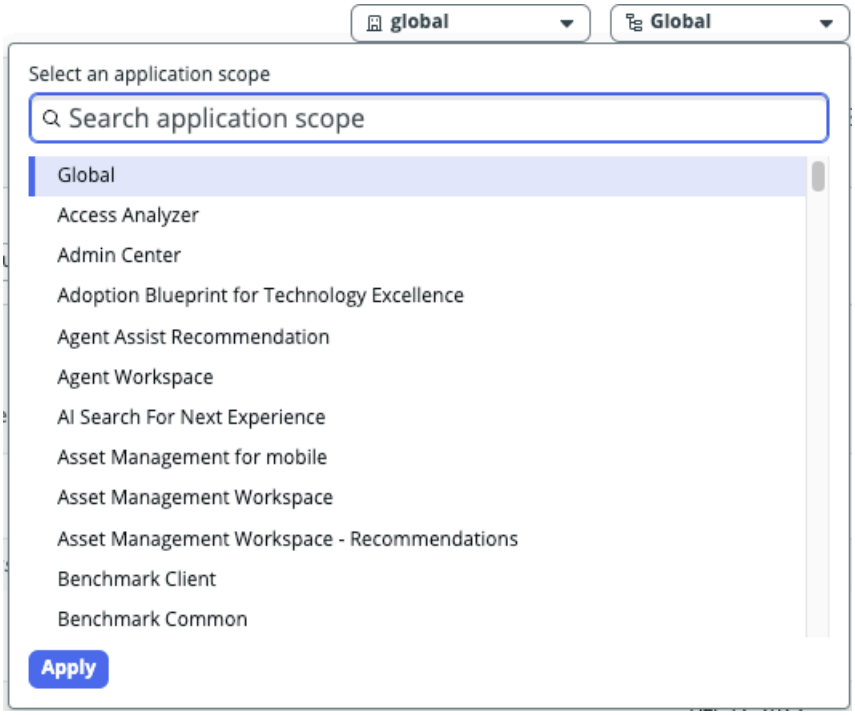
Application scope

Application scoping protects applications by identifying and restricting access to application files and data. Administrators can specify what parts of an application are accessible to other applications, which helps to protect data and application files. In UI Builder, System administrators and developers set application scope when creating a page. Note, you cannot change the application scope after creating the page, so choose your scope carefully.

Changing the scope in UI Builder also changes the scope in the ServiceNow AI Platform®, and vice versa.

When creating a page, it's important for admins and developers to be aware of the scope they are in for the workspace or portal experience. Choose the correct application scope for your experience. The scope picker is to the right of the URL field. The scope defaults to the scope that the user is currently in within the ServiceNow AI Platform®.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



If you change to a different scope while in a page, you're notified that you are in a different application scope from the one the page was created in. For more information, see Application scope.

Delegated developers for UI Builder

If you have the application-specific admin role or the system-level admin role, you can delegate application development in ServiceNow® Studio to designated developers at the UI Builder application level.

Protection policy

A protection policy prevents anyone from modifying and/or copying an application file or its related record. A protection policy is typically used when the author of an application is different than the company that uses the application. UI Builder notifies you if you try to modify a protected record. For more information, see Application file protection policy.

Learn about domain separation

Domain separation is supported for UI Builder . Domain separation enables you to separate data, processes, and administrative tasks into logical groupings called domains. You can control several aspects of this separation, including which users can see and access data.

Support level: Standard

- Includes all aspects of **Basic** level support.
- Application properties are domain-aware as needed.
- Business logic: The service provider (SP) creates or modifies processes per customer. The use cases reflect proper use of the application by multiple SP customers in a single instance.
- The instance owner must configure the minimum viable product (MVP) business logic and data parameters per tenant as expected for the specific application.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Sample use case: An admin must be able to make comments required when a record closes for one tenant, but not for another.

For more information on support levels, see Application support for domain separation.

Overview of Domain Separation in UI Builder

UI Builder is a web user interface builder. UI Builder enables developers to build new pages or customize existing pages for web-based workspace experiences using Next Experience Components and custom web components. In addition, UI Builder supports Domain Separation, which is the ServiceNow® instance-wide multi-tenant architecture.

Enable developers or dashboard builders in domain-separated environments to safely create UI application screens or dashboards while in the same browser window. Domain Separation in UI Builder works similarly to application scope to help administrators safely create or edit in a multi-tenant environment.

It's important to understand a key principle to maintaining a stable, healthy, and scalable ServiceNow® instance, where Domain Separation is installed.The primary principle is standardization. Standardization means a common configuration that most the instance operates by. When an instance has hundreds or thousands of domains, managing them successfully requires rigorous governance. Domain-specific configurations should be used only if they are deemed necessary by the instance owners. Generally, most instances should follow the common instance configuration. Doing so provides a more uniform experience across the instance. It also lets instance owners minimize code sprawl that slows the adoption of new ServiceNow® features included as part of release upgrades.

How domain separation works in UI Builder

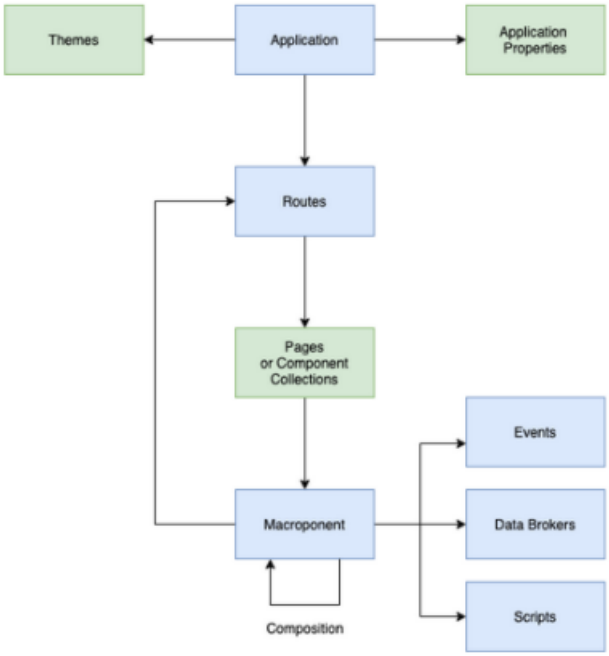
Domain separation in UI Builder works similarly to application scope to help administrators safely create or edit in a multi-tenant environment.

UI Builder is compromised of a framework of interlocked components that you use to create web-based workspaces, dashboards, or portals. While the application supports domain separation, it does not mean every component or table is domain separated, which is important for instance owners to understand.

If the current domain does not match the domain of the variant or dashboard, the record is read-only. If a user has access to the domain, they can choose to switch their domain to the domain of the record. Alternatively, users can edit the record. Editing the record temporarily forces the user session into that record's domain. They can then make edits without fear of accidentally creating an override.

The following diagram shows what is (in green) and is not (in blue) domain separated in UI Builder.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Not shown in the diagram are viewports, declarative actions, and screen applicabilities, which are domain supported as process.

Data and Process/UI separation are important when considering domain separation architecture. UI Builder fully supports data and process/UI separation, and any data (records) displayed in the web-based workspace, dashboard, or portal experiences.

For example, a change request that belongs to the domain of Acme only shows for users who have access to the domain of Acme in an experience built using UI Builder. Conversely, if an application does not support data separation, its records won't be domain separated by the workspace or portal experience.

Process/UI separation tables that form the underpinning framework in UI Builder are process separated, and a sys_override column exists on those tables. For example, if a page is created in Global, any changes to the logic created and saved in a sub domain results in an override.

For items that are not domain separated, any change to the logic globally affects any page or dashboard that references its content. Understanding domain separation is critical when interacting with these elements.

Domain Selection menu, messaging, and managing overrides

When designing a workspace, dashboard, or portal experience using the UI Builder (including Dashboard Builder), a system administrator or ui_builder_admin has access to a **Domain Selection** menu in UI Builder. A system administrator or ui_builder_admin should switch to the proper domain prior to creating, editing, or overriding a variant or dashboard page.

By default, the ui_builder_admin role does not have access to the **Domain Selection** menu. The **Domain Selection** menu must be coupled with a role that grants access, such as ITIL, or it can be added via system property. For more information, see Enable domain selection menus in Core UI.

In addition, the **Domain Selection** menu also shows **Expand/Collapse Domain Scope**, that is displayed while the system administrator or ui_builder_admin is in Global. Select **Expand** to show any variant or dashboard that has been overridden, or exists as a standalone in a sub domain. Select **Collapse** to only show variants or dashboards created in Global.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Lastly, domain hierarchy is available from the **Domain Selection** menu. For deep-domain hierarchies, the user may have to collapse the branches of the domain hierarchy to physically select the domain. In these environments, perform a search to find the domain.

UI Builder has governance controls for editing and overriding variants or dashboards, similar to the way application scope is handled. Both application scope and domain scope are handled concurrently in UI Builder.

For example, if a variant was created in Global, but the current domain of the system administrator is set to Acme, then that variant is read-only. As long as that screen is not in a private scope that prevents editing, the system administrator or ui_builder_admin have two options. They can temporarily transact into the global domain if they have access to Global. Or, they can create an override.

You can edit the domain separation to make quick changes to the variant or dashboard and its content. When you edit the domain, you temporarily transact into the same domain scope as the variant or dashboard. Going into the same scope prevents accidental overrides when modifying certain settings (such as Name, Order, Event Mappings, Page Definition configurations) tied to the variant. While in edit mode, not all settings are available in page management. For full capabilities, switch into the correct domain prior to editing the record.

Create Override allows a system administrator or ui_builder_admin to create an override of an existing variant or dashboard. Create an override of a variant or dashboard to perform an extensive copy of the page definition content, minus screen conditions and audiences in the user's currently selected domain. The sys_override column is then updated appropriately.

Viewports, which are variants in and of themselves, are domain separated, and are typically nested inside page definition content. Some viewports may not copy over. For example, a viewport (displayed as a tab set) that was created as an override in a domain of a Global viewport would not be carried in the page definition content during the override creation process.

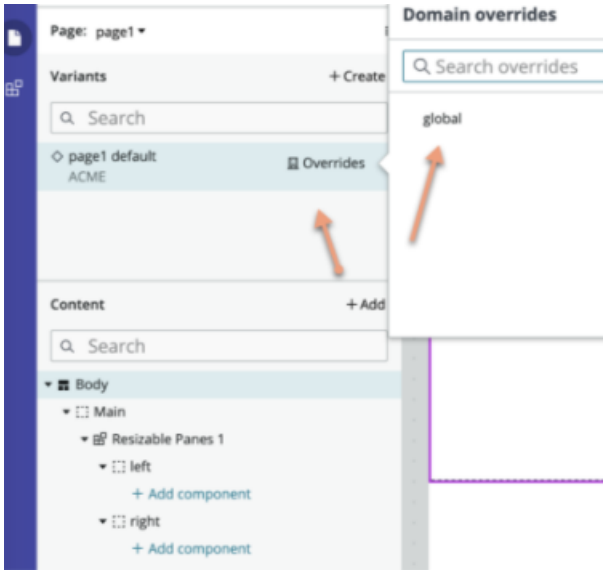
As screen conditions and audiences may be specific to a domain, this content is not carried over during the override creation process. A screen prompts the system administrator or ui_builder_admin to create screen conditions and audiences.

A user cannot create an override of a variant or dashboard in Global if the item exists in a sub domain, or if an override exists for that variant or dashboard in the same sub domain.

After the override and the conditions and audiences are set, the content and configurations can be configured as needed. As standard to domain separation, the override is no longer affected by any changes done to the original variant or dashboard. The workspace, dashboard, or portal experience displays these overridden configurations if the user's current domain session is within the affected domain or sub domains where this override was created. Audiences further determine what a user may or may not see.

In addition, a user can access the domain hierarchy to view existing overrides from higher domains. For example, Global <- Top <- Acme <- Current domain. If no overrides exist, the default variant or dashboard are displayed. The exception is if the default variant or dashboard is in a child domain or a peer domain.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



If you select **Expand Domain Scope** while in Global, all variants and overrides in sub domains are shown as previously mentioned.

System administrators and ui_builder_admin can see what has been created in the ServiceNow® platform.

Viewports and domain separation

Viewports are variants that can be nested in page definition content. They can be created as a common configuration in Global, or can be overridden per sub domain.

Declarative actions and domain separation

Declarative Actions can be overridden per domain as well. A system administrator or ui_builder_admin should select the appropriate domain prior to creating a domain specific declarative action override.

Related topics

Domain separation for service providers

Working in UI Builder

Get a full understanding of everything that you can do when creating a page for your workspace or custom portal experience in UI Builder.

Using UI Builder resources

The following information helps you along your UI Builder journey:

Configure how users interact with your applications in UI Builder

Learn what an experience is in UI Builder. Understand what an experience contains.

What is a UI Builder experience

An experience in UI Builder is a collection of web pages for users to interact with an application.

Many websites use the same basic structure on every page. For example, every page contains the same header, the same footer, and the same menus. These common page elements are called a shell. Experiences can have a shell, or wrapper, for each page in the experience. A shell

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

gives an experience a consistent look and feel as users navigate from page to page. Shells are recommended but not required. For more information, see Define UI experiences using app shells.

Ways to create an experience

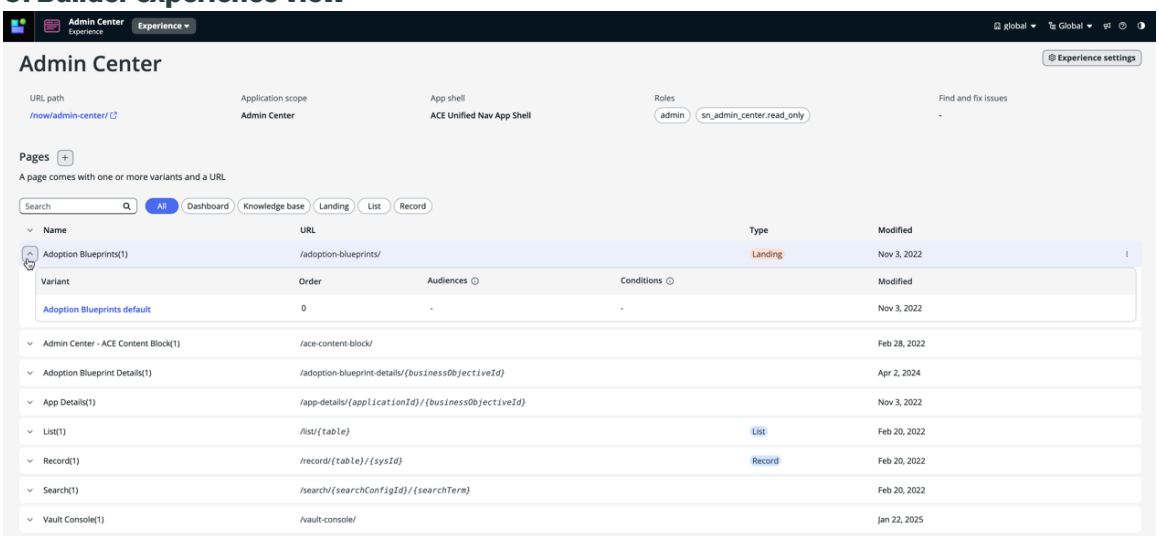
You can Create an experience for UI Builder from scratch or use UI generation to assist you in building an experience.

Experience view in UI Builder

The UI Builder experience view is a central place to view and understand the structure and details of an experience. This includes routes, page variants, and the audience and conditions required for each variant, and experience settings. Use the experience view to see the structure and hierarchy of your experience.

- View the experience URL, application scope, app shell, and roles for the experience.
- Edit experience settings.
- Quickly view and compare variants.
- Create pages and page variants.
- View a list of page collections.
- Edit page and page variant settings.
- Duplicate page variants.

UI Builder experience view



Create an experience for UI Builder

Learn how to create a workspace or portal experience for UI Builder in the ServiceNow platform.

Before you begin

Role required: ui_builder_admin

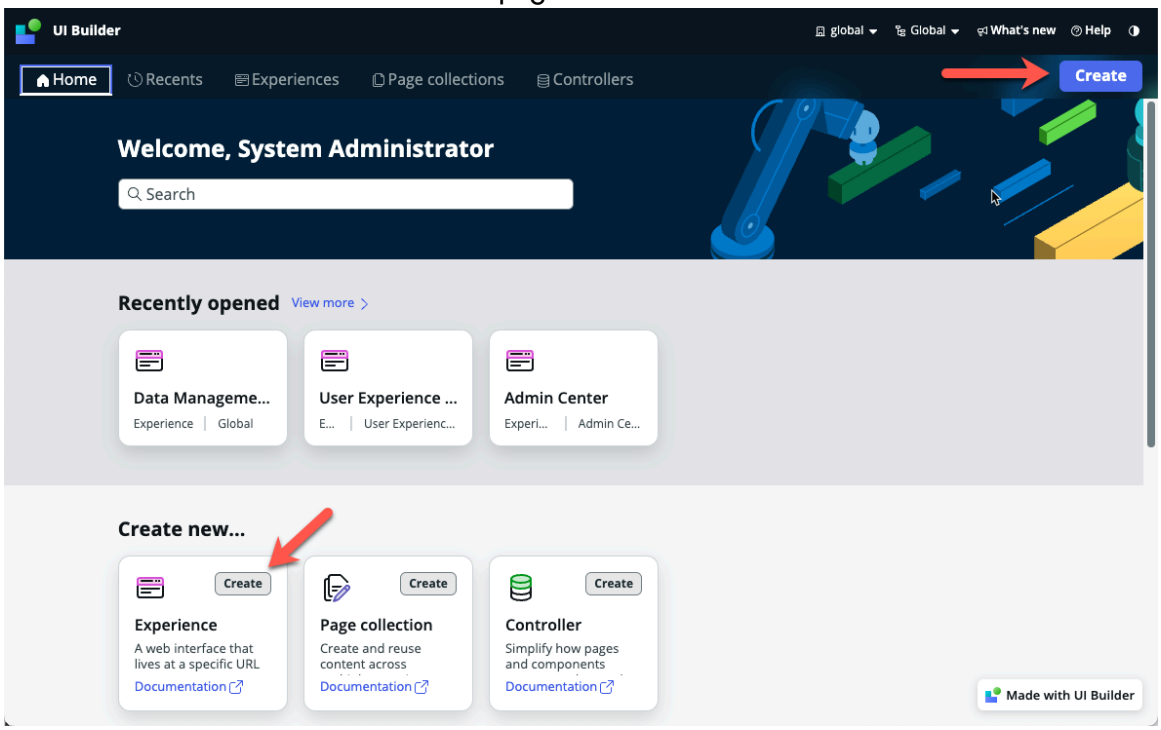
About this task

This task shows you a basic example of how to create a workspace or portal experience for UI Builder. This example creates an experience called Demo Experience to demonstrate the process.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

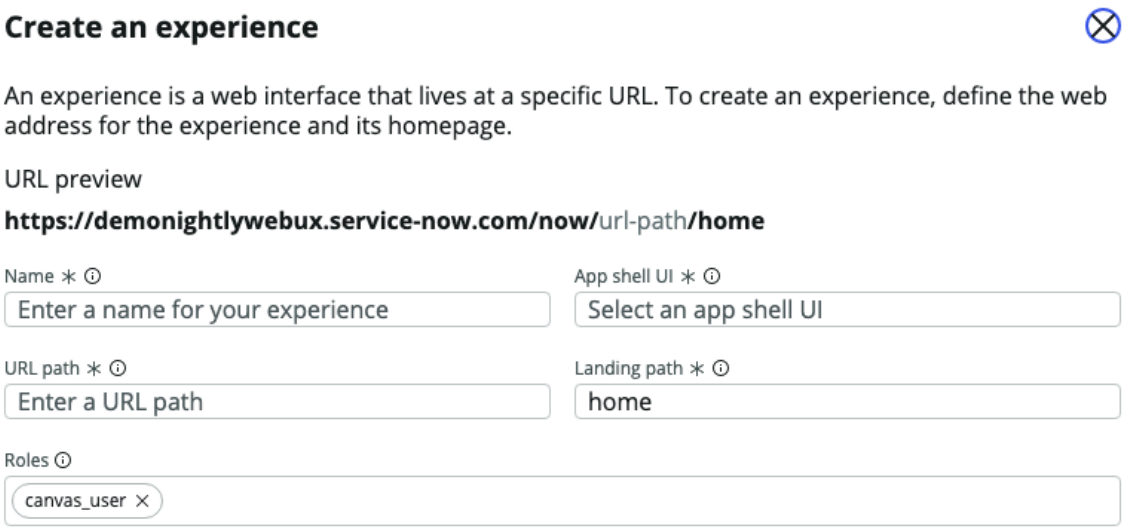
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Select **Create** from the UI Builder home page.



3. Select **Experience**.
4. In the form, fill in the fields.

Create an experience dialog box



Create an experience form

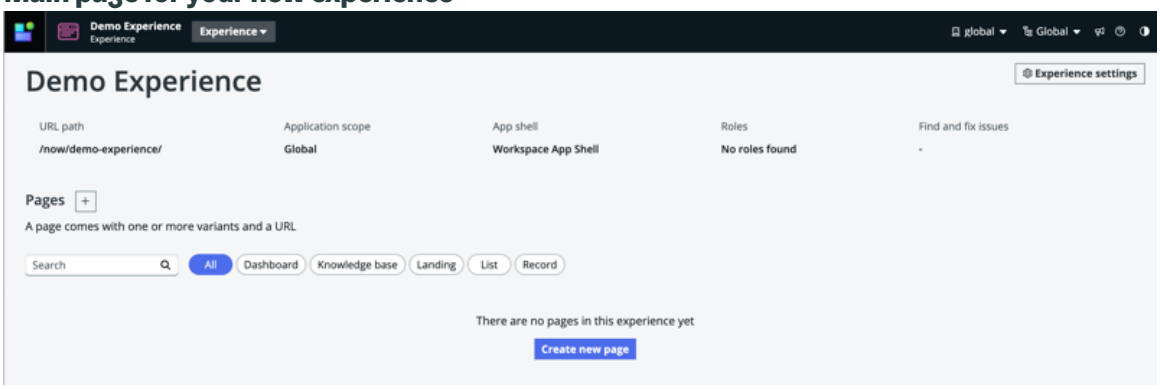
Field	Description
Name	Add a name to track your experience internally. The experience name is visible to users from the browser tab. For this example, it is MyApp.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
App Shell UI	You must choose an app shell for the experience to work. For example, you could choose a workspace or portal app shell. The app shell is the wrapper of the page contents, which is similar to the functionality of a web page. The app shell can show things like the logo of your company, user preferences, the search icon. For more information, see Define UI experiences using app shells.
URL path	Add a path for your experience. The path appends to the end of the URL parameter of your experience. For example, if you used `MyApp` as the title for your experience, you could type `my/app` for the path.
Landing path	The landing path is the prefix that people use to reach your experience homepage. To designate a page as the homepage, you must create a page that has a matching path.
Roles	Only users with these assigned roles can access the experience. If you leave this field empty, the experience is open to all logged-in users by default.

5. Select **Create** to create your experience.
6. Select **Open experience** to view the main page for your experience.

Main page for your new experience



Note: In addition to the name of your new experience, the screen includes the URL, application scope, admin panel, and the roles of users who can view the experience. Any pages or variants you create for the experience appear in the **Pages** section.

What to do next

Now that you have created an experience you can add and customize pages. For more information, see Manage UI Builder pages and page variants.

Configure UI Builder workspace experiences

You can change the UI Builder workspace experience settings to fit your company goals.

By changing the experience settings, you can affect how your users interact with your workspace experience, how your workspace looks, and how users navigate to and around your workspace.

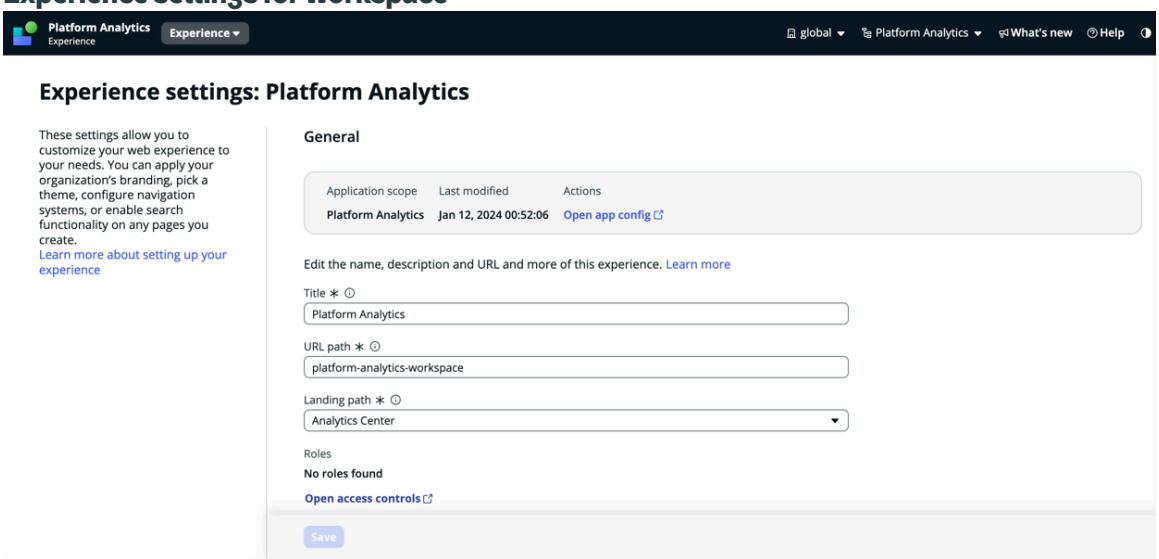
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you can edit the experience settings, you must be in the correct application scope. If you're in a different scope, the experience settings are read-only. To change your application scope, go to the main header, select the application picker (Global), and then select the application scope that you want. For more information about the application scope, see Learn about security and roles.

Learn how you can change these workspace experience settings:

- Change or add the workspace general settings for your workspace experience. For example, you can change the title, the description, and the path of the workspace.
- View the brand and theme setting in your workspace experience to see the theme currently assigned to your experience. The theme sets the look and feel of the experience.
- Change the workspace side navigation settings to add pages to the side navigation in your workspace experience.
- Modify the workspace utility settings to turn on or turn off notifications for your workspace experience. Also, you can change the global search settings for your workspace experience. For example, you can choose to show or hide the search bar in your workspace experience or change the search source that determines where the search results come from.

Experience settings for workspace



Change the general settings in your workspace experience

Change the general settings for your workspace experience in UI Builder to fit the needs of your organization. For example, you can modify the title, description, and path for your workspace experience.

This video show you how to perform the following procedure.

Before you begin

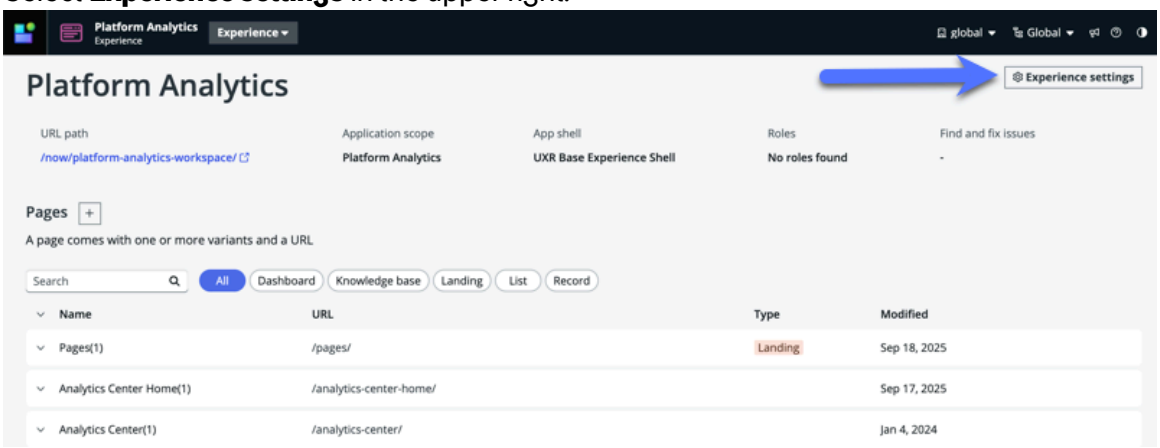
You must be in the correct application scope to edit the experience settings. If you're in a different scope, the experience settings are read-only. To change your application scope, go to the main header, select the application picker (Global), and then select the application scope that you want. For more information about the application scope, see Application scope.

Role required: ui_builder_admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

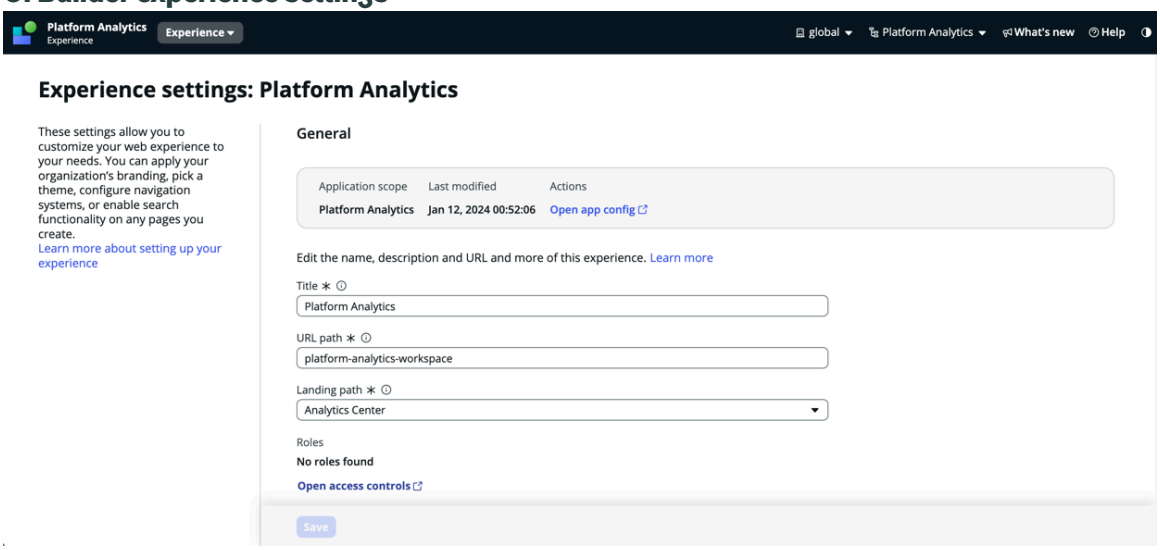
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   For more information, see Configure how users interact with your applications in UI Builder.
3. Select **Experience settings** in the upper right.



4. Change the title, description, and URL path of your workspace experience.
   - The title lets everyone know what the workspace is called. Take care when choosing a name for your workspace.
   - The description lets your users know the details of the workspace.
   - The path must be unique. The path can include digits (0-9), letters (A-Z, a-z), and a few special characters (" - ", " . ", " _ ", or " ~ ") with the words separated by a forward slash or hyphen.

UI Builder experience settings



5. **Optional:** Select **Advanced settings** to edit the workspace record on the ServiceNow AI Platform.

   a. Update the record in the platform when you're finished.
   When you go back to your workspace experience settings in UI Builder, you see the changes that you made to the general settings.

6. Click **Save**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

View the brand and theme setting in your workspace experience

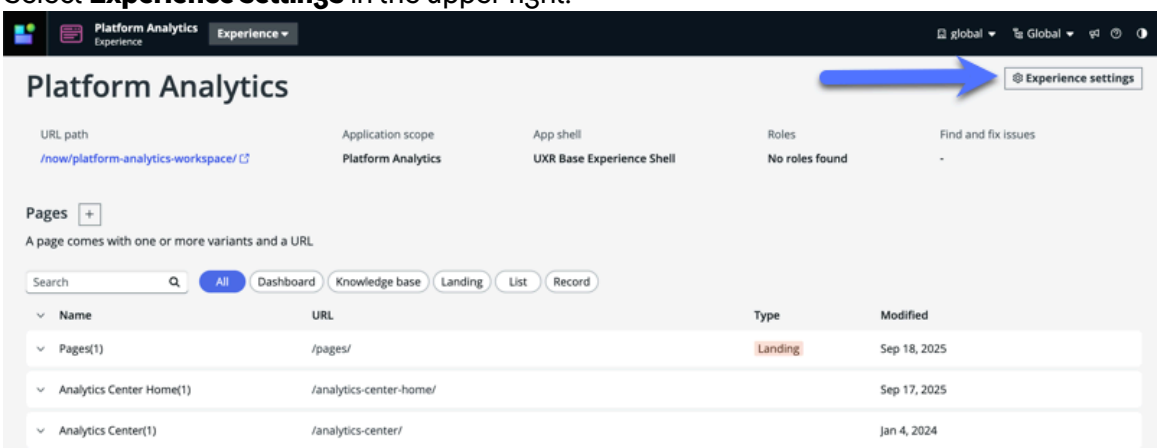
View the theme setting for your workspace experience in UI Builder. The theme sets the visual style of the experience and provides a consistent look and feel across all pages.

Before you begin

Role required: ui_builder_admin

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Select **Experience settings** in the upper-right.



4. Scroll to the **Branding and theming** section and note what is listed in **Themes**. By default, the Polaris theme is applied to UI Builder experiences.

What to do next

If you want to customize the theme, there are two options:

- **Theme Builder** enables you to create, edit, manage, and apply Next Experience themes using a friendly visual interface. For more information, see Configuring Next Experience with Theme Builder.
- If you are an admin user comfortable working in style records and JSON code, you can create a custom theme to override the default Polaris style. This option gives you the most control over typefaces, colors, and images (including the banner and logo) in a theme. For more information, see Configuring Next Experience themes and preferences.

Change the navigation and menu settings in your UI Builder workspace experience

Select pages for side navigation in your UI Builder workspace experience. From any page in the workspace experience, users can navigate to the pages you selected.

Before you begin

This task has the following prerequisites:

- You have an existing workspace experience on UI Builder. For more information, see Configure how users interact with your applications in UI Builder.
- This workspace experience was created with the Workspace App Shell.
- This workspace experience includes at least two pages.
- You want users to be able to navigate from a page to other pages that you specify.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Role required: ui_builder_admin

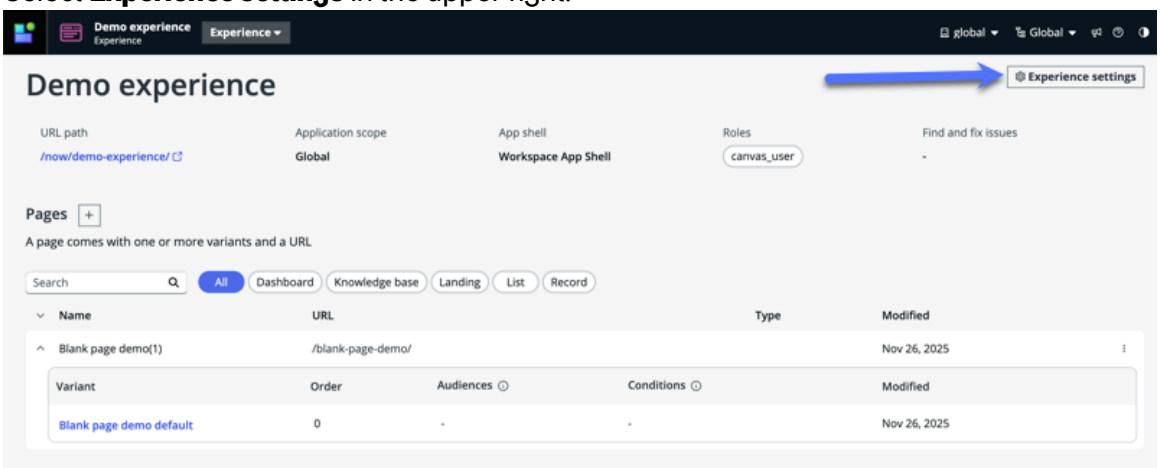
About this task

You can add up to seven pages to your side navigation.

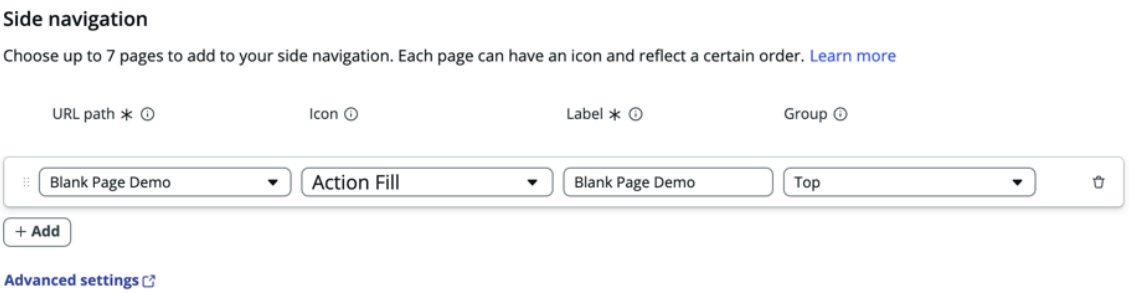
You must be in the correct application scope to edit the experience settings. If you are in a different scope, the experience settings are read-only. To change your application scope, go to the main header, select the application picker (Global), and then select the application scope that you want. For more information about the application scope, see Learn about security and roles.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open your workspace experience in the UI Builder.
3. Select **Experience settings** in the upper-right.



4. Scroll down to **Side navigation**.
5. Select **+ Add**.



6. Fill in the following fields:

Field	Description
URL path (required)	Select the page that you want to navigate to. **Tip:** The landing page is usually the first page that you want a link to.
Icon	Select an icon for the link. Default: No icon

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Label (required)	Enter a name for the page, which appears in a tooltip for the link.
Group	Place the link in the top or the bottom group of the side navigation panel. Default: Top

7. Add more pages as needed.
8. Select **Save**.
9. Select the experience name (for example, Demo Experience) in the UI Builder header.
10. Select the **URL path** link to open the experience in a new tab and check that the side navigation is displayed on the left side.

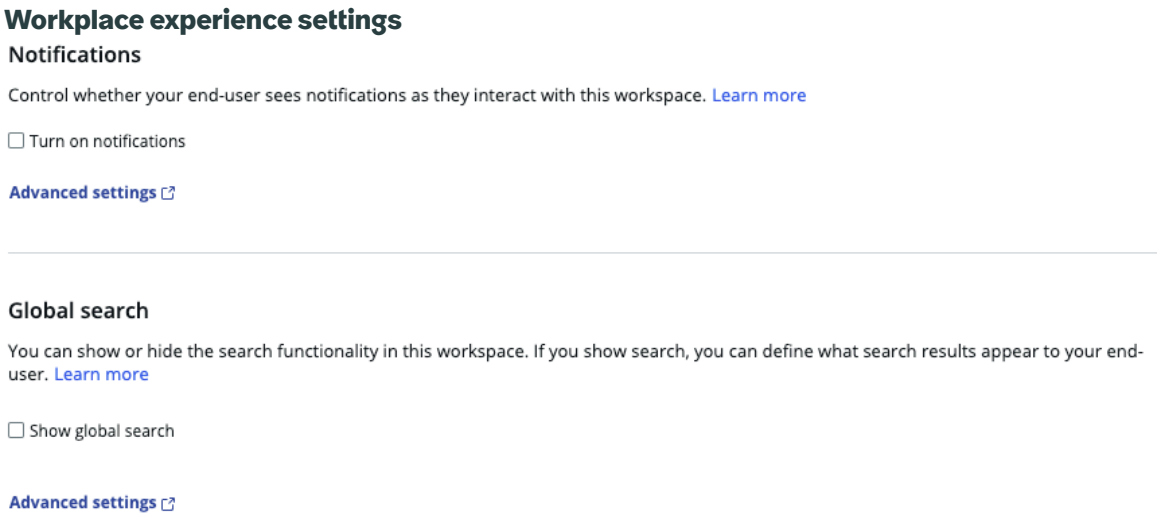
> **Tip:** If the side navigation doesn't work as expected, consider reviewing the UX Page Property [sys_ux_page_property] record underlying the side navigation. Reopen the Side Navigation experience settings, as described in Steps 3-4. Select **Advanced side navigation settings** to open the record. Verify that the fields have the correct values, such as Type=`json`.

Set up notifications in workspace utility settings

Set up the advanced notification and search settings in your workspace experience so that you can control how your users see notifications and search options.

Before you begin
Role required: ui_builder_admin

About this task
Set up the advanced notification and search settings in your workspace experience. You can turn notifications on or off to control how people see notifications in your workspace experience.



You can also control whether the search functionality is visible to the users of your workspace experience. If you choose to display a search option, you can define what search results are returned by choosing a source for the search.

Before you can edit the experience settings, you must be in the correct application scope. If you're in a different scope, the experience settings are read-only. To change your application

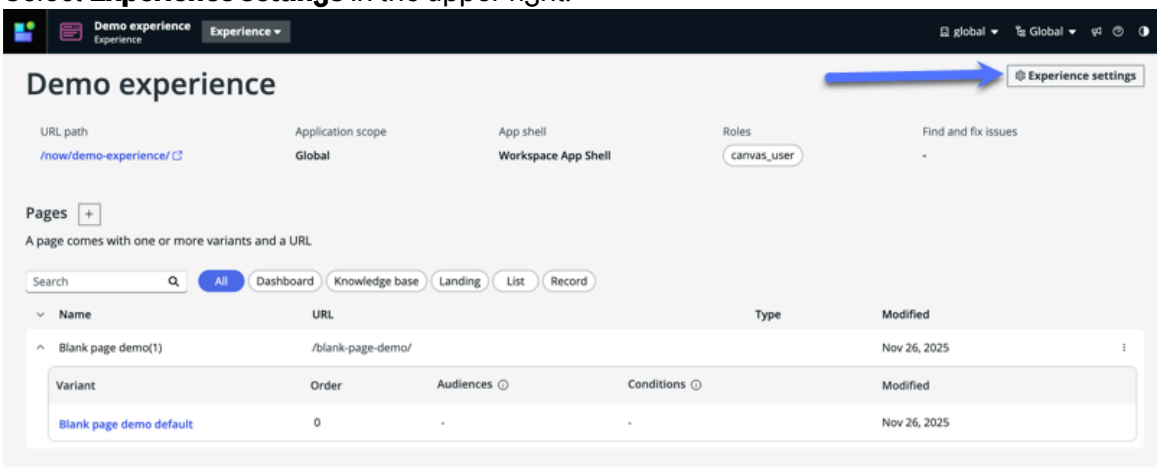
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

scope, go to the main header, select the application picker (

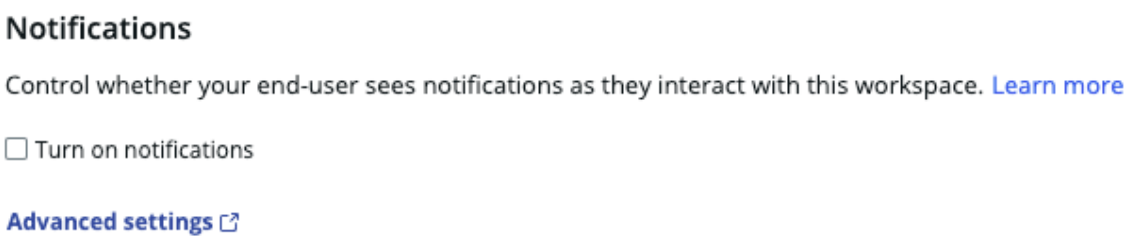
), and then select the application scope that you want. For more information about the application scope, see Learn about security and roles.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. For more information, see Configure how users interact with your applications in UI Builder.
3. Open or create a page in your workspace experience.
4. Select **Experience settings** in the upper-right.



5. Scroll down to **Notifications**.



6. Select the **Turn on notifications** option.
7. Click **Save**.
8. **Optional:** Click **Advanced settings** to go to the ServiceNow AI Platform® and edit the JSON values of the record.

> **Note:** If you have the Record component on a page, you can have a record open in the workspace experience when a notification is selected. Follow the instructions in Record UIB Setup on the ServiceNow developer site and define a featureRoutes page property.

Display global search in a workspace experience

Show or hide the search functionality in your workspace.

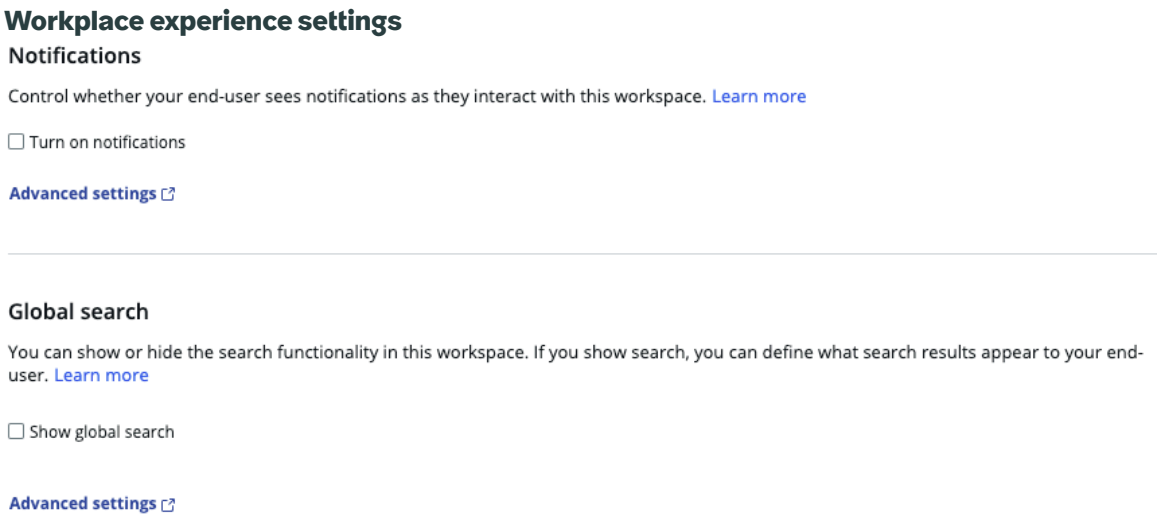
Before you begin

Role required: ui_builder_admin

About this task

You can control whether the search functionality is visible to the users of your workspace experience. If you choose to display a search option, you can define what search results are returned by choosing a source for the search.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

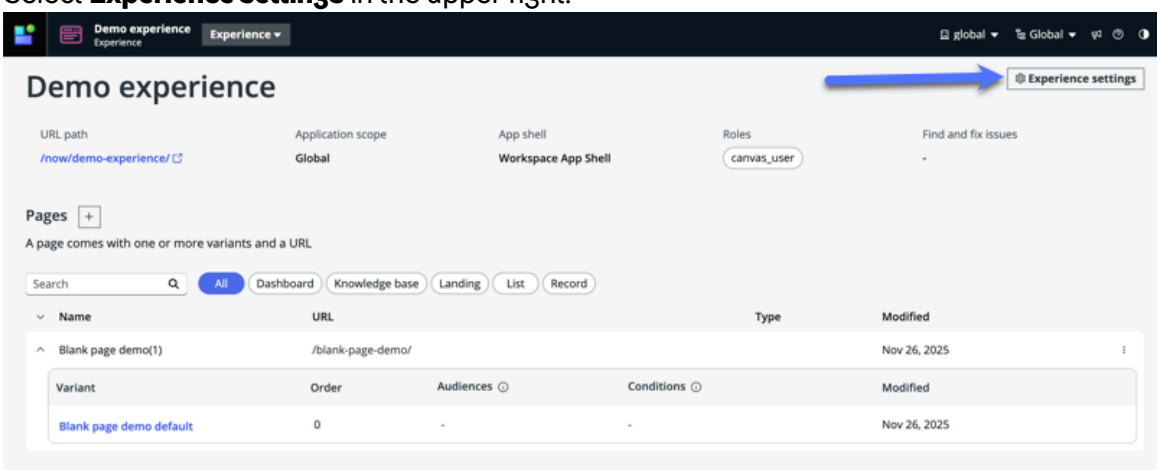


Before you can edit the experience settings, you must be in the correct application scope. If you're in a different scope, the experience settings are read-only. To change your application scope, go to the main header, select the application picker (

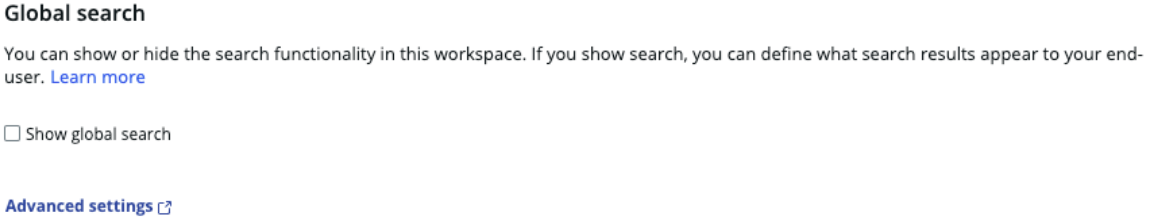
), and then select the application scope that you want. For more information about the application scope, see Learn about security and roles.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. For more information, see Configure how users interact with your applications in UI Builder.
3. Open or create a page in your workspace experience.
4. Select **Experience settings** in the upper-right.



5. Scroll down to **Global search**.



6. Select the **Show global search** option.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

7. Click **Save**.
8. **Optional:** Click **Advanced settings** to go to the ServiceNow AI Platform® and edit the JSON values of the record.

Configure UI Builder portal experiences

You can change your UI Builder custom portal experience settings to affect how your users interact with your portal experience.

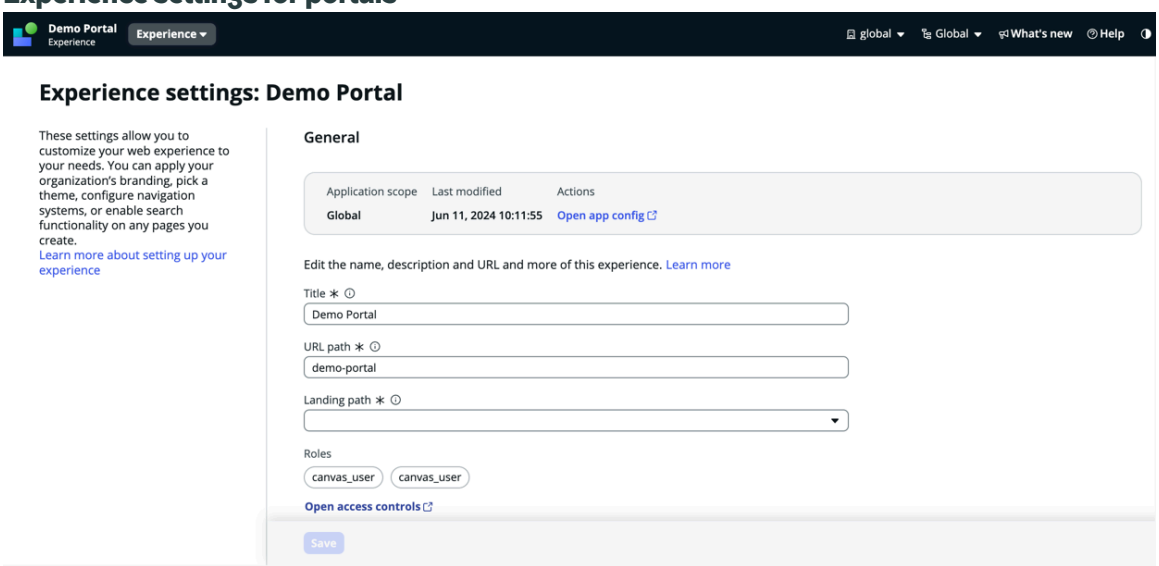
Use experience settings to change the settings for the custom portal experience that you are working in. These settings affect how your users interact with your portal experience, how your portal looks, and how users navigate to and around your portal.

You must be in the correct application scope to edit experience settings. If you are in a different scope, the experience settings are read-only. To change your application scope, in the main header, select the application picker (Global), and then select the application scope that you want. For more information about the application scope, see Learn about security and roles.

Learn how to change the following portal experience settings.

- Modify or add portal general settings for your portal experience. Change the title of your portal, the description, and the path of the portal.
- View the brand and theme setting in your portal experience to see the theme currently assigned to your experience. The theme sets the look and feel of the experience.
- Use portal navigation and menu settings to set up the navigation and menu settings in the app shell of your portal experience. The app shell is the wrapper of the portal contents. For example, the app shell can show things like the logo of your company, user preferences, the search icon, the configuration icon, and the user menu. For more information about app shells, see Define UI experiences using app shells.
- Modify the portal search settings to show or hide the global search functionality on public or private pages in your portal experience.

Experience settings for portals



Change the general settings in your portal experience

Change the general settings for your custom portal experience in UI Builder. For example, you can modify the title, description, and path for your portal experience.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

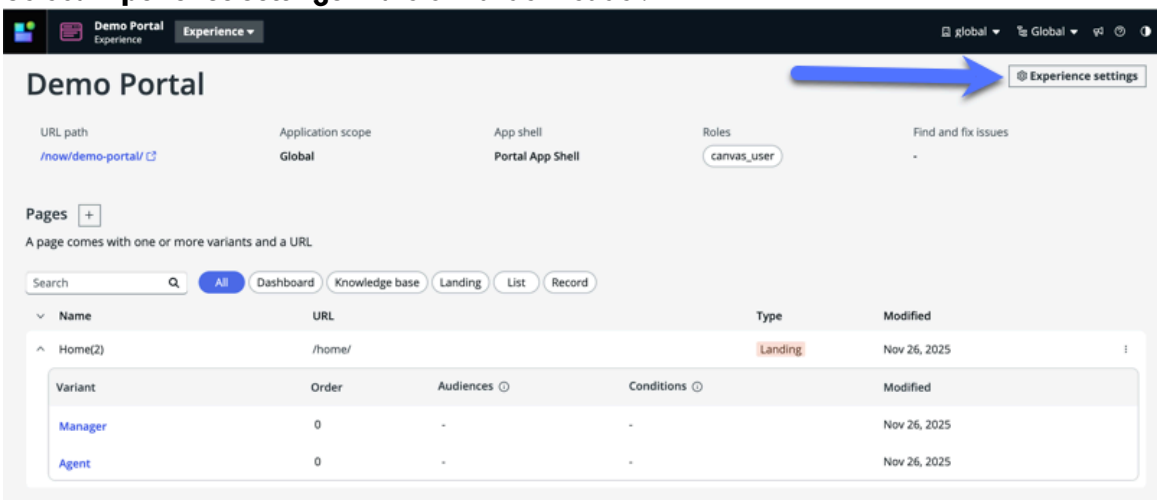
Before you begin

You must be in the correct application scope to edit the experience settings. If you're in a different scope, the experience settings are read-only. To change your application scope, go to the main header, select the application picker (Global), and then select the application scope that you want. For more information about the application scope, see Learn about security and roles.

Role required: ui_builder_admin

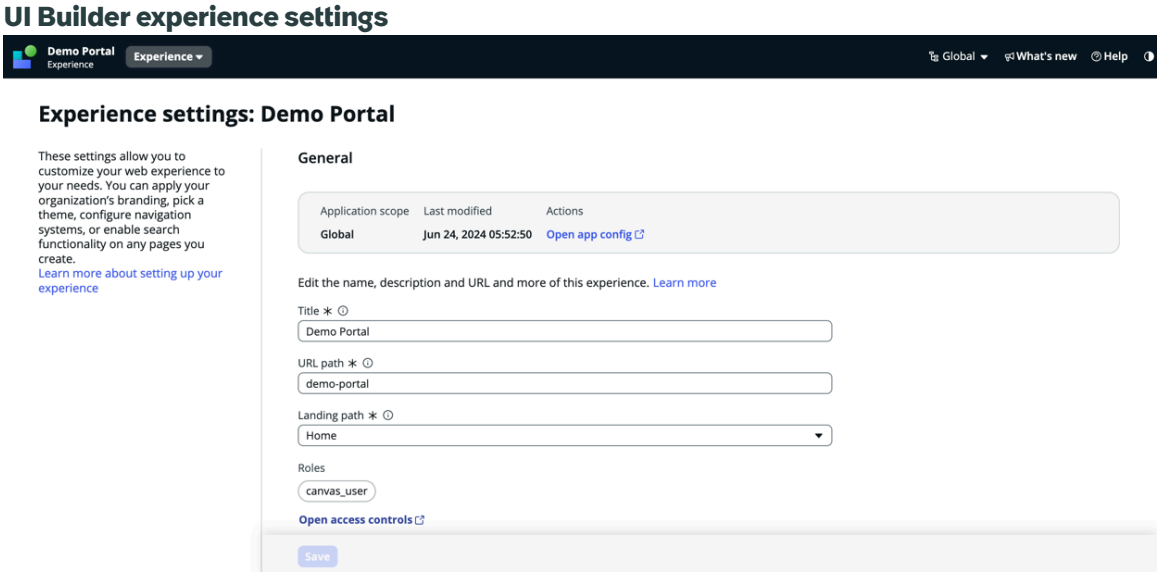
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open a portal experience to work in or create an experience by selecting **Create** > **Experience**. For more information, see Configure how users interact with your applications in UI Builder.
3. Select **Experience settings** in the UI Builder header.



4. Change the title, description, and path of your portal experience.
   - The title lets everyone know what the workspace is called. Take care when choosing a name for your workspace.
   - The description lets your users know the details of the workspace.
   - The path must be unique. The path can include digits (0-9), letters (A-Z, a-z), and a few special characters (`" - "`, `" . "`, `" _ "`, or `" ~ "`) with the words separated by a forward slash or hyphen.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



5. **Optional:** Select **Advanced settings** to edit the workspace record on the ServiceNow AI Platform.

   a. Update the record in the platform when you are finished.
   When you go back to your workspace experience settings in UI Builder, you see the changes that you made to the general settings.

6. Click **Save**.

Result

When you go back to your portal experience settings in UI Builder, you see the changes that you made to the general settings.

View the brand and theme setting in your portal experience

View the theme setting for your portal experience in UI Builder. The theme sets the visual style of the experience and provides a consistent look and feel across all pages.

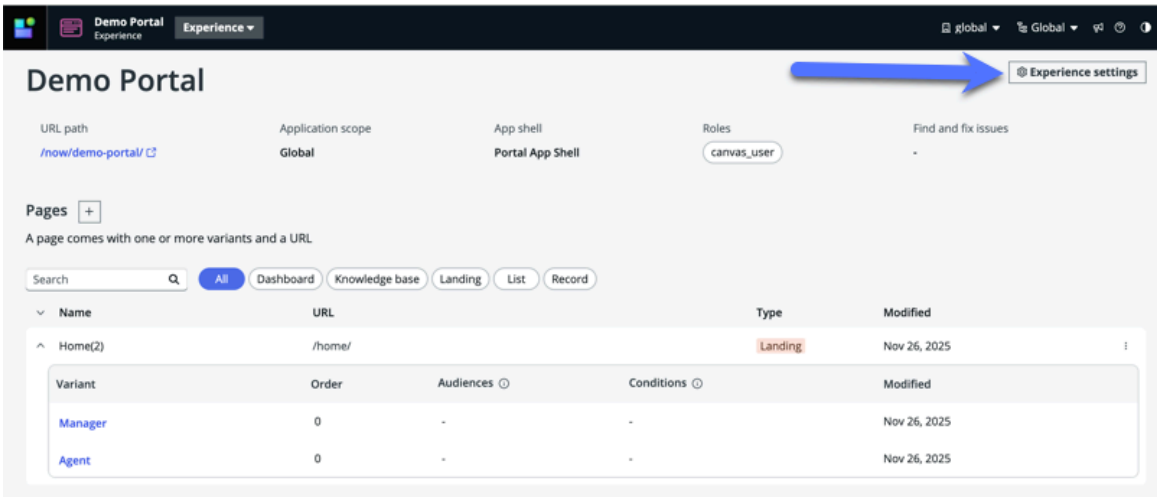
Before you begin

Role required: ui_builder_admin

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open a portal experience to work in or create an experience by selecting **Create** > **Experience**. See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Select **Experience settings** in the UI Builder header.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



4. Scroll to the **Branding and theming** section and note what is listed in **Themes**.
   By default, the Polaris theme is applied to UI Builder experiences.

What to do next

- **Theme Builder** enables you to create, edit, manage, and apply Next Experience themes using a friendly visual interface. For more information, see Configuring Next Experience with Theme Builder.
- If you are an admin user comfortable working in style records and JSON code, you can create a custom theme to override the default Polaris style. This option gives you the most control over typefaces, colors, and images (including the banner and logo) in a theme. For more information, see Configuring Next Experience themes and preferences.

Change the navigation and menu settings in your portal experience

Change the header, footer, and menu settings for your custom portal experience in UI Builder to fit the needs of your organization.

Before you begin

This task has the following prerequisites:

- You have an existing portal experience on UI Builder. For more information, see Configure how users interact with your applications in UI Builder.
- This workspace experience contains at least two pages.
- You want users to be able to navigate from a page to other pages that you specify.

Role required: ui_builder_admin

About this task

Set up the navigation and menu settings in the app shell of your portal experience. The app shell is the wrapper of the portal contents. For example, the app shell can show things like the logo of your company, user preferences, the search icon, the configuration icon, and the user menu. You can configure this app shell in the ServiceNow AI Platform. For more information about app shells, see Define UI experiences using app shells.

You must be in the correct application scope to edit experience settings. If you are in a different scope, the experience settings are read-only. To change your application scope, in the main header, select the application picker (

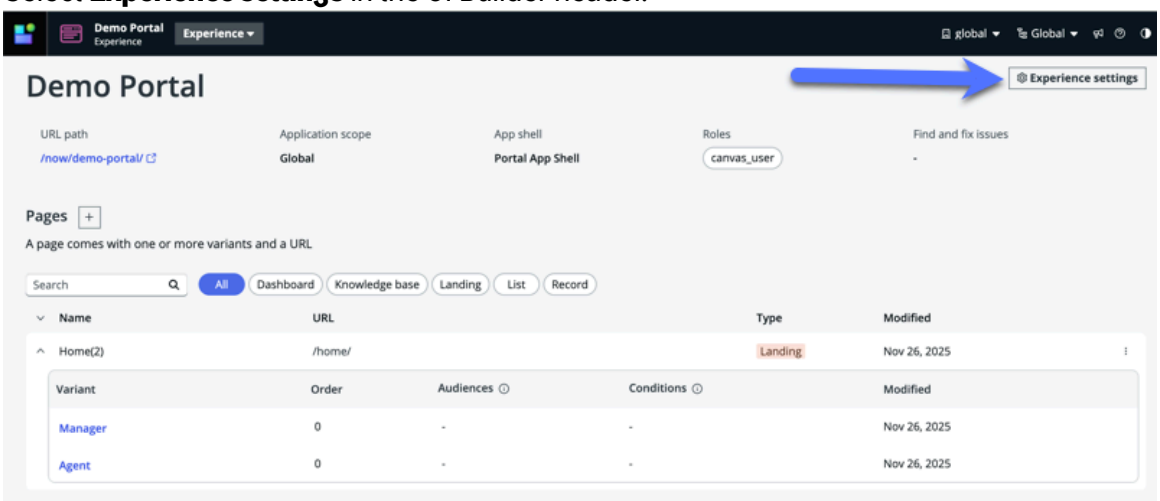
), and then select the application scope. For more information about the application scope, see Learn about security and roles.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

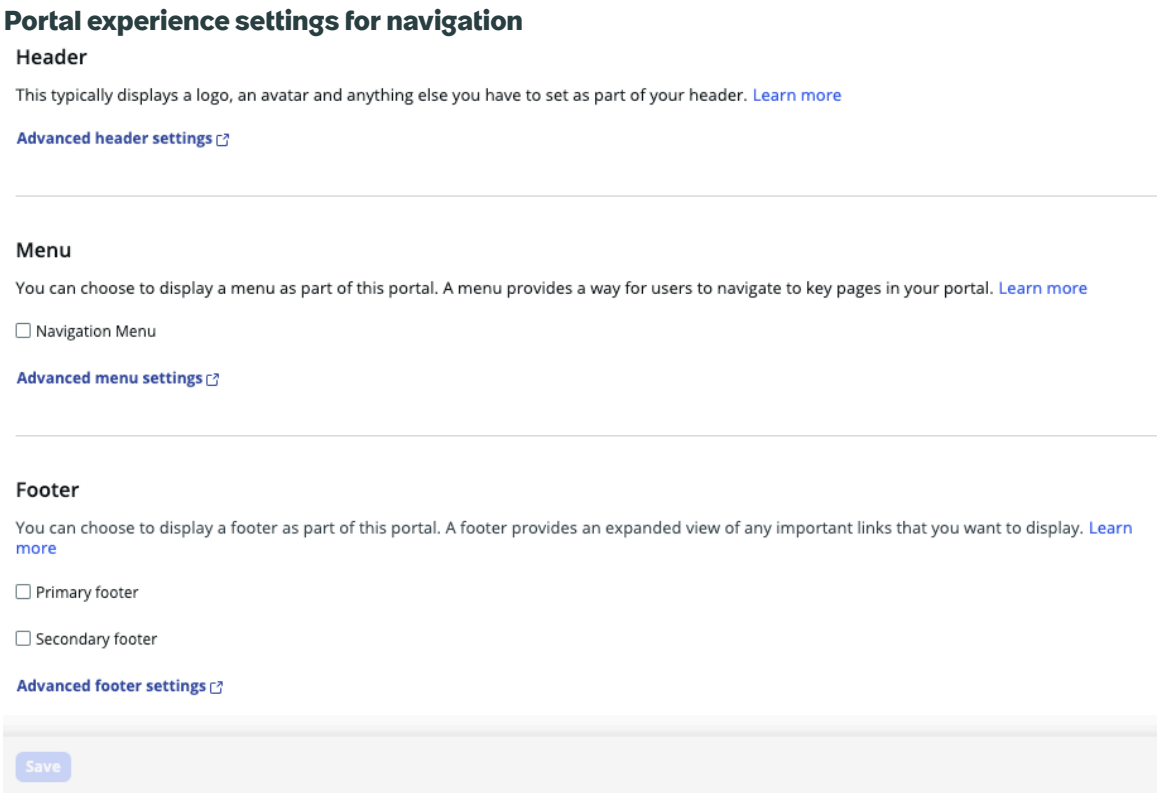
Note: For advanced information about UX page properties, including JSON examples, see https://www.servicenow.com/community/next-experience-articles/workspace-app-shell-ux-page-properties/ta-p/2331956.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open a portal experience to work in or create an experience by selecting **Create** > **Experience**. For more information, see Configure how users interact with your applications in UI Builder.
3. Select **Experience settings** in the UI Builder header.



4. To change the settings for the header, footer, and navigation menu for the app shell of the portal experience you're working in, select the advanced settings links.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Action	Procedure
Change the header	To go to the ServiceNow AI Platform®, click **Advanced header settings**. Here, you can change what information shows in the header of your portal experience. You can change things like a company logo, user preferences, the search icon, the configuration icon, and so on.
Show the menu	**a.** To add a menu to your portal experience, select **Navigation Menu**. **b.** To remove the navigation menu, deselect **Navigation Menu**. **c.** To go to the ServiceNow AI Platform, select **Advanced navigation menu settings**. Here, you can change what menu and menu items are displayed in the navigation menu. For more information, see Define UI experiences using app shells.
Display a footer	**a.** To add a primary footer to your portal experience, select **Primary footer**. The footer can show a footer logo, content, and other important links for your portal. **b.** To go to the ServiceNow AI Platform, click **Advanced navigation menu settings**. **c.** To add a second footer to your portal experience, select **Secondary footer**. A secondary footer includes additional text links, social media elements, and copyright information.

5. Click **Save**.

Show or hide the search settings for your portal experience

Show or hide the global search functionality on the public or private pages for your custom portal experience in UI Builder so that your users can search on the portal.

Before you begin

Role required: ui_builder_admin

You can control whether the global search functionality is visible to users of your portal experience. You can enable or disable the global search for either public or private pages.

You must be in the correct application scope to edit the experience settings. If you are in a different scope, the experience settings are read-only. To change your application scope, in the main header, select the application picker (

), and then select the application scope. For more information about the application scope, see Learn about security and roles.

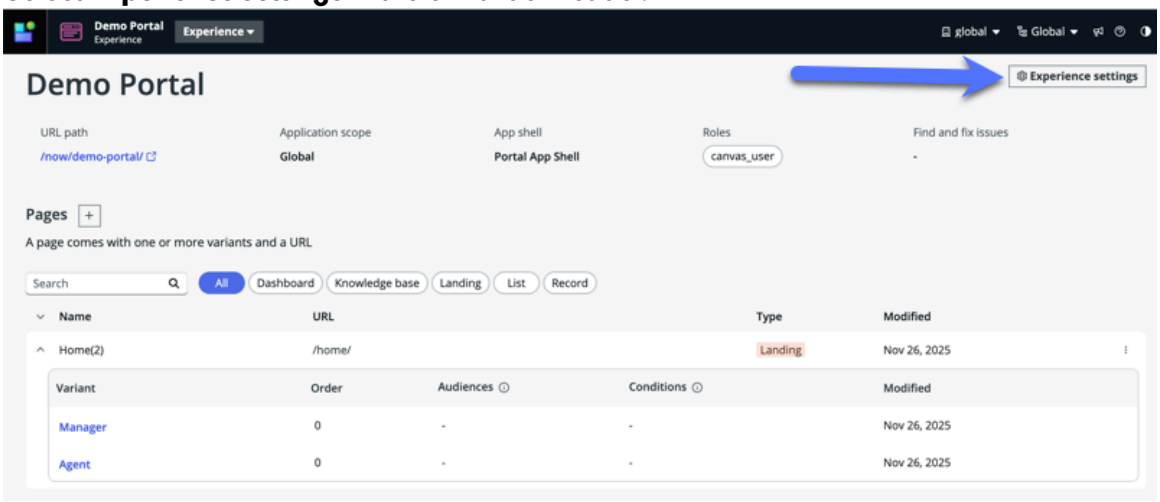
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open a portal experience to work in or create an experience by selecting **Create** > **Experience**.

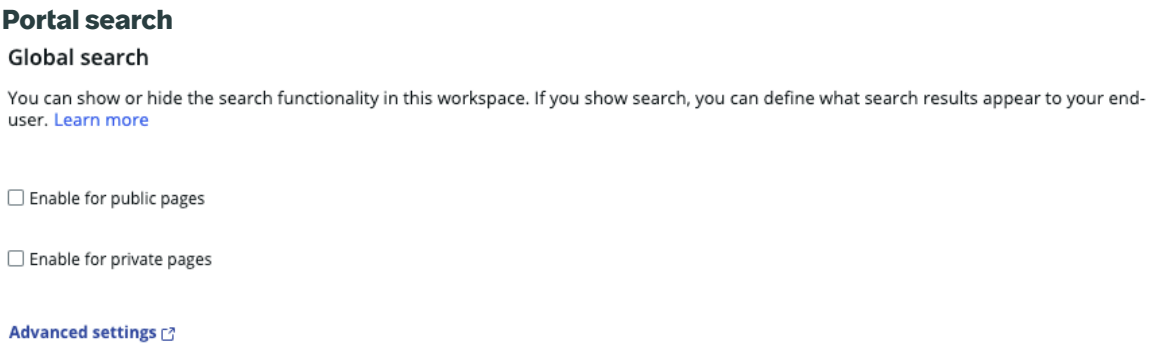
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For more information, see Configure how users interact with your applications in UI Builder.

3. Select **Experience settings** in the UI Builder header.



4. Scroll down to **Global search**.
5. Select the **Enable for public pages** or **Enable for private pages** to show or hide the global search in public or private pages.



6. **Optional:** Click **Advanced settings** to go to the ServiceNow AI Platform to make changes to the application record on the ServiceNow AI Platform®.
   To enable or disable the search for a page, you must create a separate UX Page property. The UX Page property overrides the global search setting.
7. Click **Save**.

Define UI experiences using app shells

Understand what app shells are, what app shells are available for UI Builder, and why you would pick one over another.

App shells

An app shell is the wrapper of the contents of an experience page. An app shell includes functionality similar to a web page. For example, an app shell can show things in a header or footer of your experience, such as a logo for your company, user preferences, a search icon, configuration icon, user menu, and so on.

An app shell is required for UI Builder.

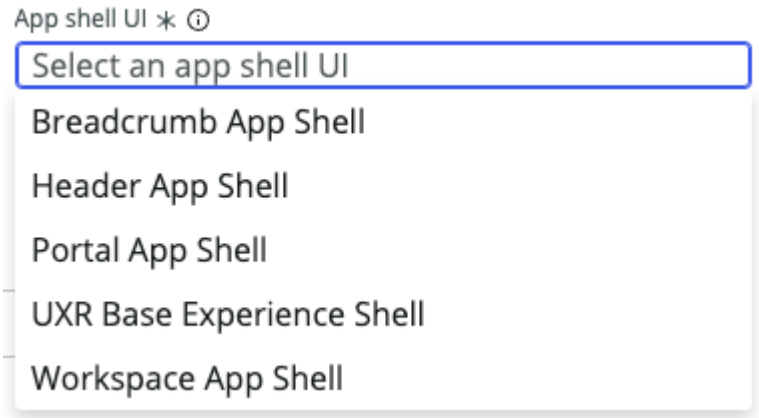
The app shell defines whether your experience has a workspace or portal design. A workspace is a graphical user interface that puts multiple tools on one page for handling requests from users.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

A portal is a page where users can add requests, such as order items, track their tickets, and so on.

You choose which app shell to apply to your experience when you create an experience in the ServiceNow® platform. For more information on how to create an experience, see Configure how users interact with your applications in UI Builder.

You can choose from various app shells when creating an experience.



App shell descriptions

App shell	Description	Example
Breadcrumb app shell	The breadcrumb app shell includes a header, breadcrumb style navigation, and a primary navigation bar for a workspace experience. Use the breadcrumb app shell for focused workflows and wizard flows that don't require multi-tasking.	
Header app shell	The header app shell includes a header and user menu for a workspace experience. Use the header app shell for experiences that require minimal navigation and few actionable items in the header.	
Portal app shell	The portal app shell includes the header and footer for a portal experience. The portal app shell record is a reference implementation of how the menu, utilities, logo, and login are configured for a portal experience.	Experience created with Portal app shell
UXR Base Experience app shell	The UXR Base Experience shell creates an experience with the Next Experience banner. The UXR Base Experience shell provides base functionality support for experiences that are configured within a parent app, including experience configuration, context binding, routing, screen loading, and	

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

App shell descriptions (continued)

App shell	Description	Example
	caching. The UXR Base Experience shell has no visual or experience-specific components. The UXR Base Experience shell can also be used for experiences that have no shell customization. For example, Identity Center directly uses UXR Base Experience Shell as its app shell.	
Workspace app shell	The workspace app shell includes the headers and footers for a workspace experience. Use the workspace app shell to provide tools that users need to find, research, and resolve issues. To manage the visual style of experiences created with this app shell, see Configuring Next Experience with Theme Builder.	

Manage UI Builder pages and page variants

Learn what a page is in UI Builder. Understand the building blocks of a UI Builder page, such as column layouts and components.

UI Builder quick start

Create pages in UI Builder as part of a workspace or custom portal experience. UI Builder pages consist of layouts and components. You build a page using column layouts and components to guide a user through an experience. For example, you could build a page to manage travel requests for your employees. The page could have column layouts with components that contain lists of all travel requests submitted and approved. You can add buttons that let users add and submit travel requests. The way you build your page is limitless.

Note: One developer should work on any one variant at a time.

Column layouts and components in UI Builder

Column layouts and components are the building blocks of your UI Builder page. Add column layouts and components to your page to build or customize your workspace or portal experience. For example, add a column layout and within a column place a button component that lets users submit requests.

You can add column layouts, columns, and components, as well as navigate between them, from the stage or the Content tree.

A column layout can contain one or more columns. Columns can contain components. Change the visual styling of column layouts, columns, and components to make it your own experience.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Types of page layout elements

Type	Description
Column layout	A flexible container with one to six columns. Add column layouts to provide structure and a framework for a page.
Column	Fill columns with components.
Component	Base elements of a page, such as buttons, lists, and forms.
Modal	Use a modal to create a page type in UI Builder that renders on top of the page and requires action.
Popover	A popover is a type of container that appears above a UI Builder page when triggered by an event. Use the popover component to display additional information or actions related to the page.

See Customize UI Builder pages using components for more information.

Create a UI Builder page

Create a UI Builder page to build a page experience from scratch or use a page template. Build a page one component at a time. If you use one of the pre-defined page templates, you start from a base structure and can customize it to meet your needs.

Name your UI Builder page. Set the path (or keep the default path that is automatically added based on your page name). A default path is added based on your page name. You can also create your own path, but the path must be unique. The **URL preview** shows the path of your page. Set the page URL type. URL types help you categorize and find important pages in the experience view.

The application scope protects applications by identifying and restricting access to application files and data. The application scope defaults to the scope that the user is currently in within the ServiceNow AI Platform®. For more information about application scope, see Learn about security and roles.

Use a UI Builder page template to create a page based on a pre-defined page template, and then customize the page to your needs. You can reference or copy a page template. For more information, see Create a page from a template.

Create a UI Builder page: Advanced settings

Add required parameters to your UI Builder page. A required parameter is a piece of data that your page requires, such as a sys_id, table, or query. Required parameters are useful for components as they can bind to the value of the required parameter. For example, you can add a table parameter to the URL field and then bind data to that table. When the table is referenced, it exposes the table data to any components on the page. Required parameters are visible in the URL when you add them to your page. In the following example, a required parameter called table was added. Notice it was appended to the URL.

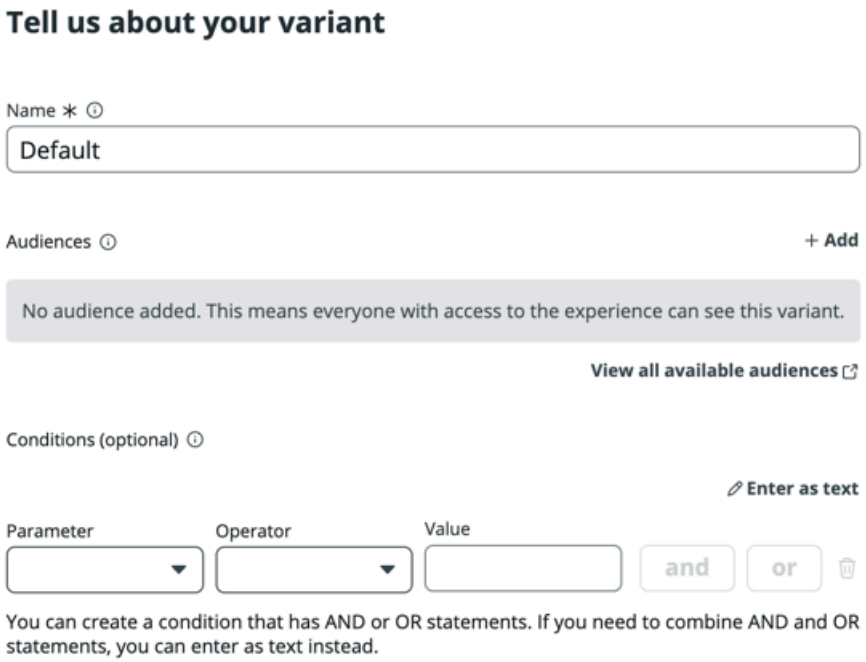
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Add optional parameters to your UI Builder page. Optional parameters are optional pieces of data that you can add to the URL of your page. Unlike required parameters, optional parameters are always name and value pairs that work no matter what order they're provided.



Set the audience and conditions settings for UI Builder page variants. When you create a page, UI Builder also creates a variant of the page for you by default. A page variant is a variation of your page at the same path that lets you target experiences for different audiences using user criteria. For example, a page for managers, and a variant of that page for the manager's direct reports. For more information about creating a variant, see Create a page variant.

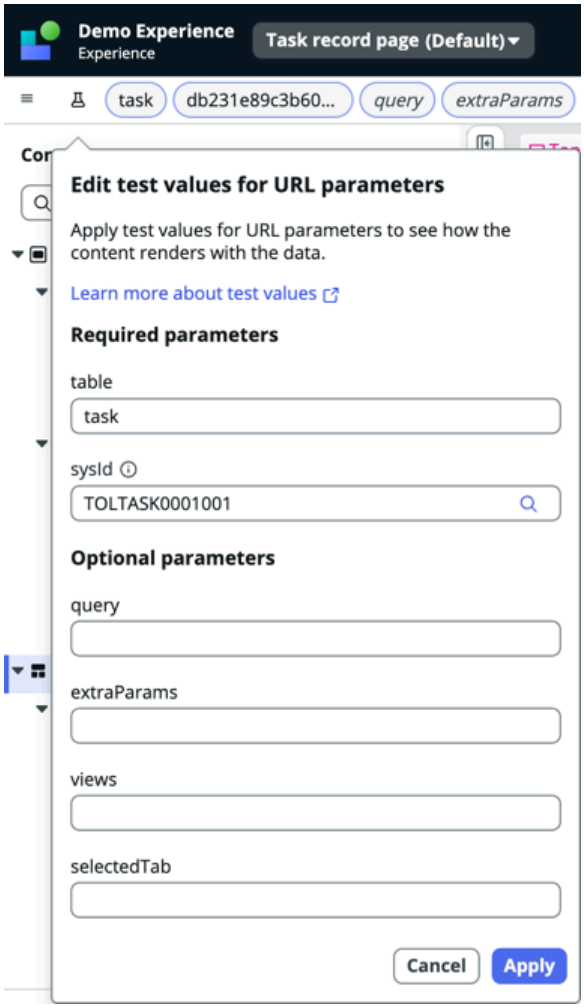


For more information about audiences, see Learn about audiences.

Test values in UI Builder

Add a test value to your UI Builder page as a way to bring test data into the page for testing purposes. For example, if you add a table as a required parameter, you could add a test value of incident and bind a data resource to it to bring in test data on the incident for that table.

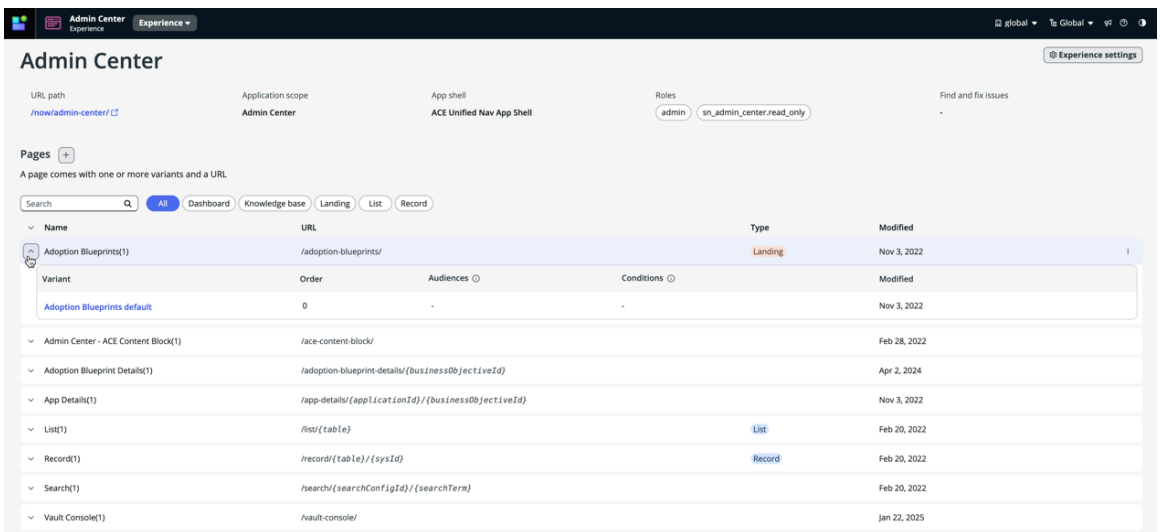
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



To get test values to show data, add a data resource, then configure the data resource to bind a record to the test value in the URL. For example, you could add `incident` as a test value. Then add a data resource named **Look up a single record**. In the **Table field**, dynamically bind the incident test value to a `context.props.table` table, as shown in the following image.

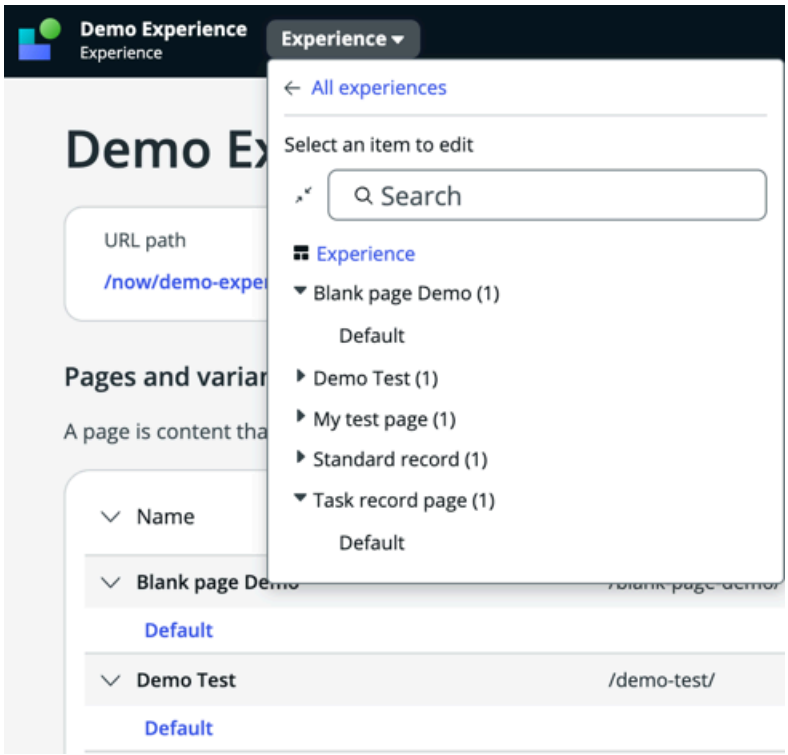
View an existing UI Builder page

You can work in any UI Builder page in your experience based on your role settings. Click on the name of the page you want to work in while in the experience view of your experience.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

While in the page editor, open a different page variant by selecting the drop-down in the header.



Create a UI Builder page variant

A variant lets you target different audiences with different content, using user criteria. For example, you can create a homepage for agents, and a variant of the page for managers that exists at the same URL path.

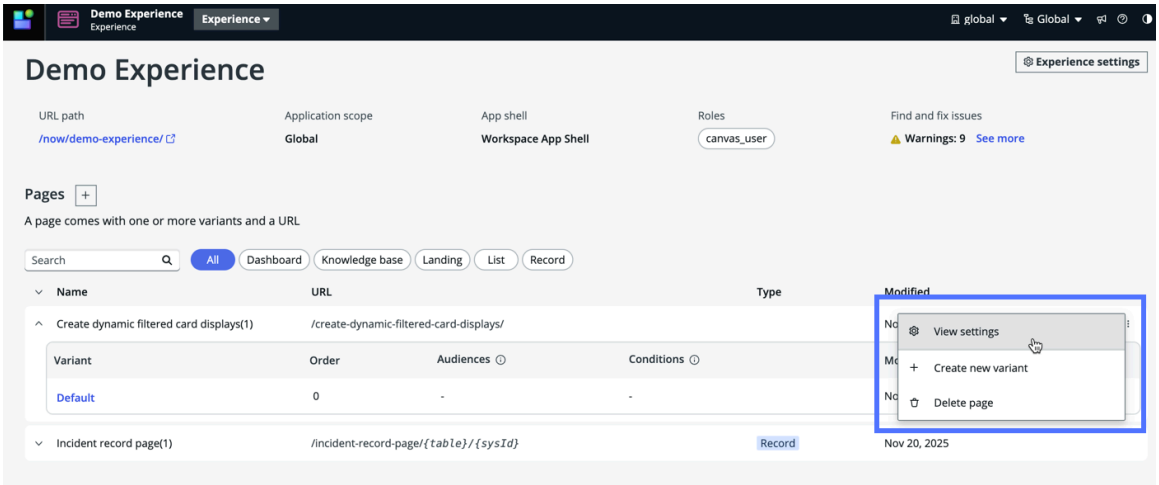
You set the audience for each UI Builder page variant. The audience determines who uses the page variant. For example, if you create a travel request page, create a variant of that page for managers to manage the employee travel requests. You set the audience for the manager page for anyone in the manager role. Employees cannot view that variant. For more information about audiences, see Learn about audiences.

See Create a page variant for more information.

Edit UI Builder page settings

Change the settings of your UI Builder page at any time by selecting the More actions icon then **View settings** in the UI Builder experience view. You can change the name, path, and parameters of your page after you create a page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

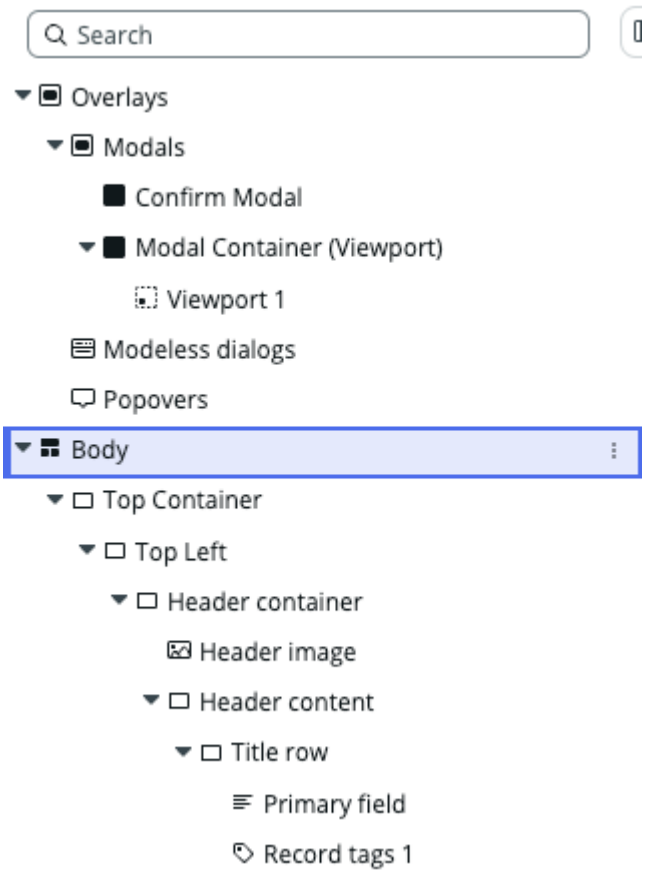


See Edit a page for more information.

UI Builder Content tree

The content tree in the page management panel is important. It not only shows every column layout, column, and component on your UI Builder page, it lets you easily find your components and work with them. The content tree is especially important when you have multiple components on your page. You select the component in your content tree to highlight the component on the page, making it easier to build your page.

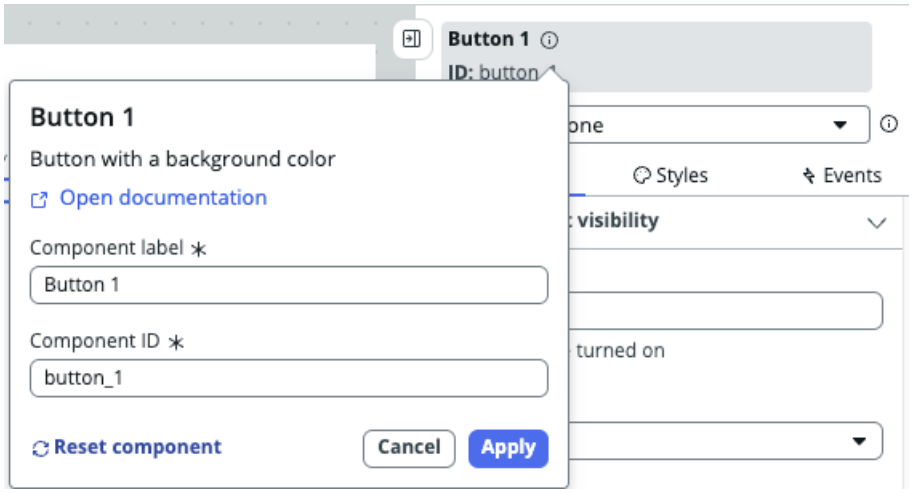
Content tree with components



When you click a component in the content tree, it highlights the component on your stage so you can configure, add styling, events, data, and so on. You can drag items in the content tree to reorganize your page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

It’s important to add a component label when you add a component to your page. The component label is used in the content tree to apply labels IDs to each component in the content tree. You can identify the components much easier in the content tree when they're labeled appropriately.

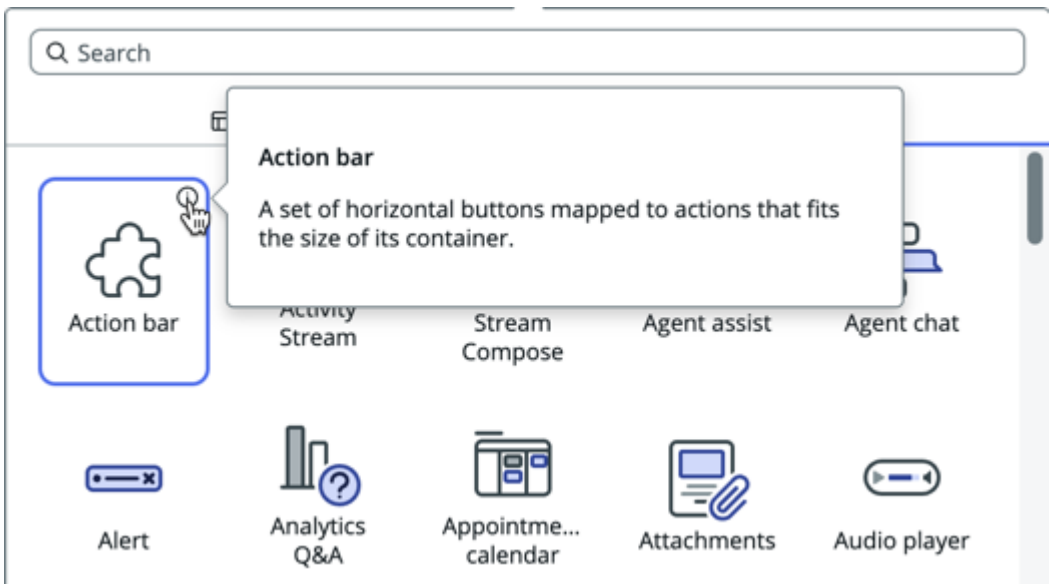


UI Builder Toolbox

Use the UI Builder toolbox to add layouts and components to your page. Access the UI Builder toolbox by selecting **+ Add content** or the **+** icon in the stage.

Search the toolbox to find layouts and components that you want to add to a UI Builder page.

View a brief description of a layout or component by selecting the information icon ⓘ to view the tooltip. The components tab displays different colors if a component can use a preset or is bundled with multiple components.

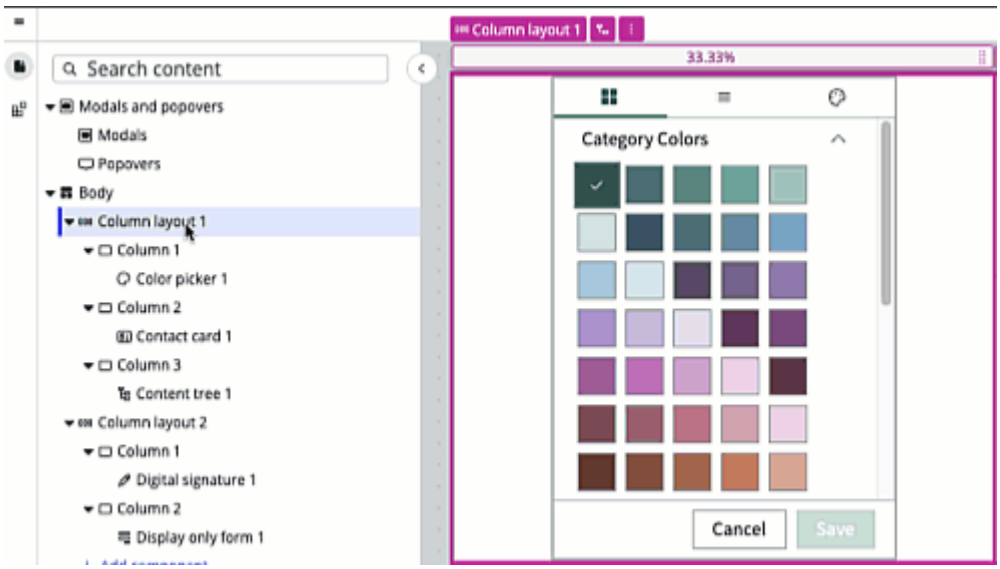


UI Builder Stage

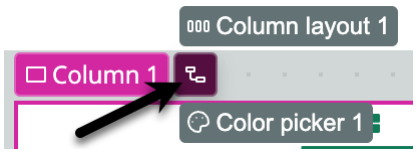
The stage is the largest area in the UI Builder editor and is used for building pages. Add your column layouts and components here by either selecting a + button, or dragging from the Component pane to the page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

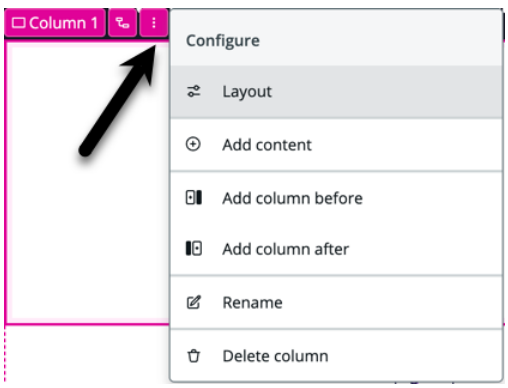
On the stage, the body, column layouts, and columns are outlined in magenta. Components are outlined in blue.



Use the Tree item icon to navigate to a different layer of content. For example, if a column is selected, easily navigate to the parent column layout or a component within the column.



Select the Menu icon on a column layout or component for options such as duplicate and delete.



Changes made on the stage are reflected in the Styles panel at right. Changes made in the Styles panel immediately update the elements on the stage.

Scale the size of the stage in UI Builder

Adjust the size of the stage in the UI Builder editor by selecting the Width drop-down.

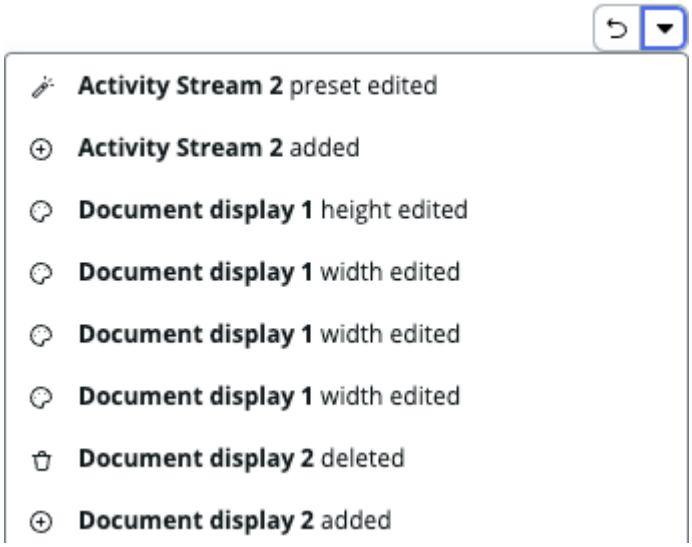
Undo and Redo in UI Builder

Undo and redo changes you've made to your UI Builder page. Reverse any action you make to a page layout, component, style or layout.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The undo and redo function can be found in the header toolbar. You can select the undo ( ) or redo ( ) button to reverse an action. You can also click the undo drop down to go back multiple steps. You cannot undo and redo changes made in the data shelf.

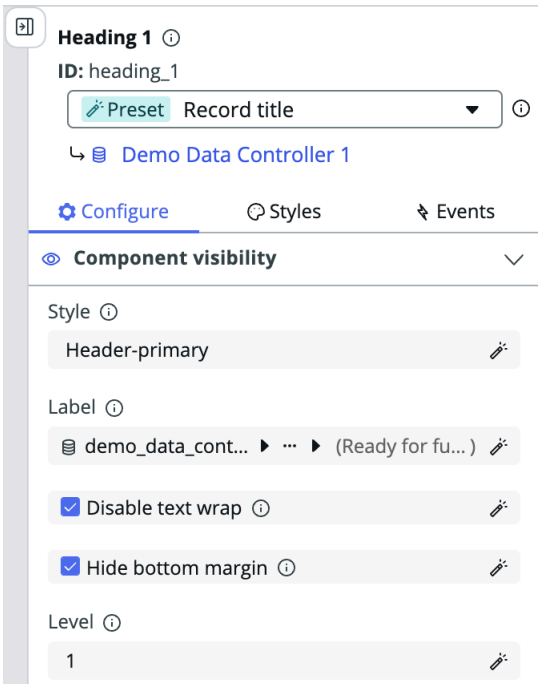
You can undo the previous 20 actions by selecting the undo drop-down.



UI Builder Configuration panel

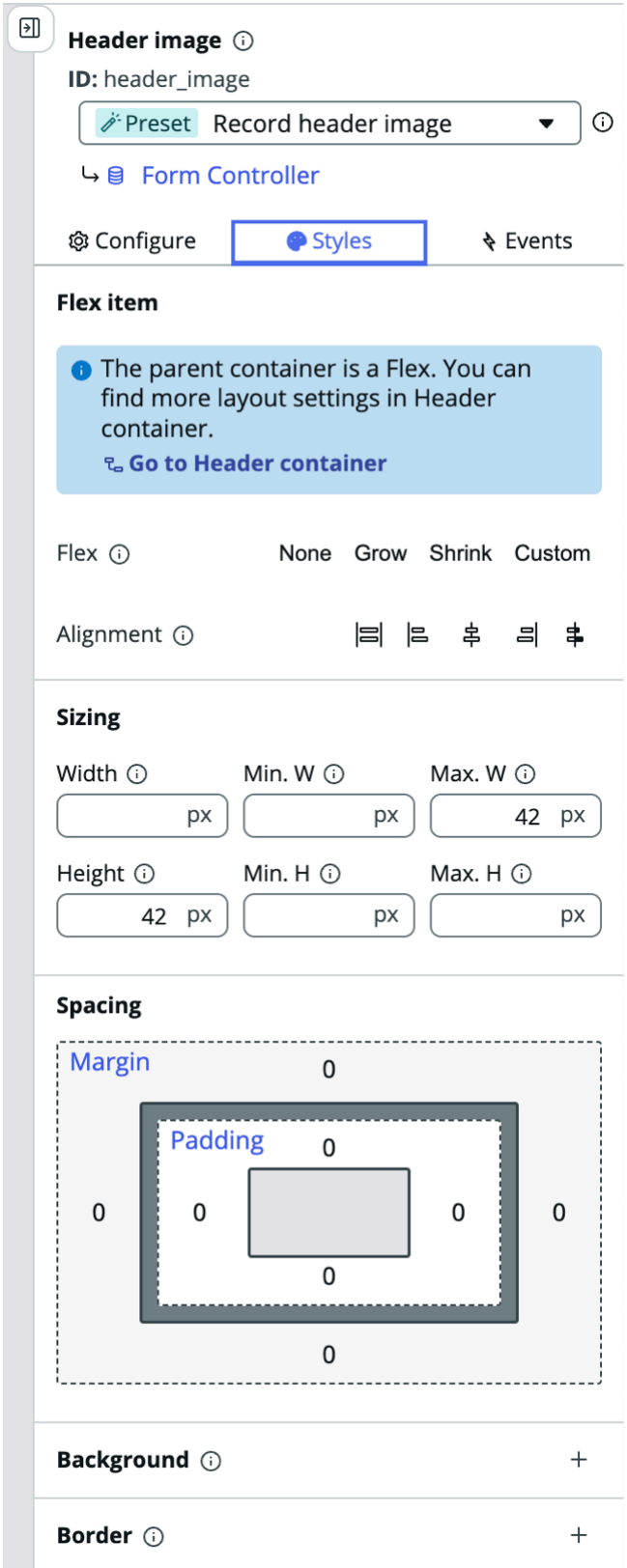
Use the configuration panel to work with components, including arranging, styling, and setting up event handlers.

- Select the **Config** tab to configure components. Each component has different configuration options that let you control the necessary parts of the component. For example, a button component is simple, and you can only configure its appearance, label, and a few properties. A list component is more complicated, and contains dozens of editable list parameters.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

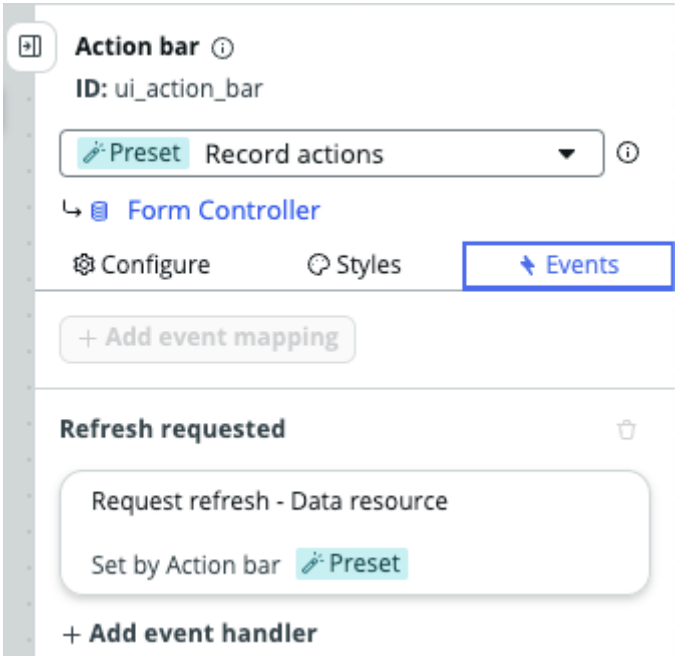
- Select the **Styles** tab in the configuration panel to add styling to a component. You can use standards-based Cascading Style Sheets (CSS) to change the visual style of a component. For example, add or change a background color, a border, or any other CSS style.



See Change the default appearance of components for more information.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

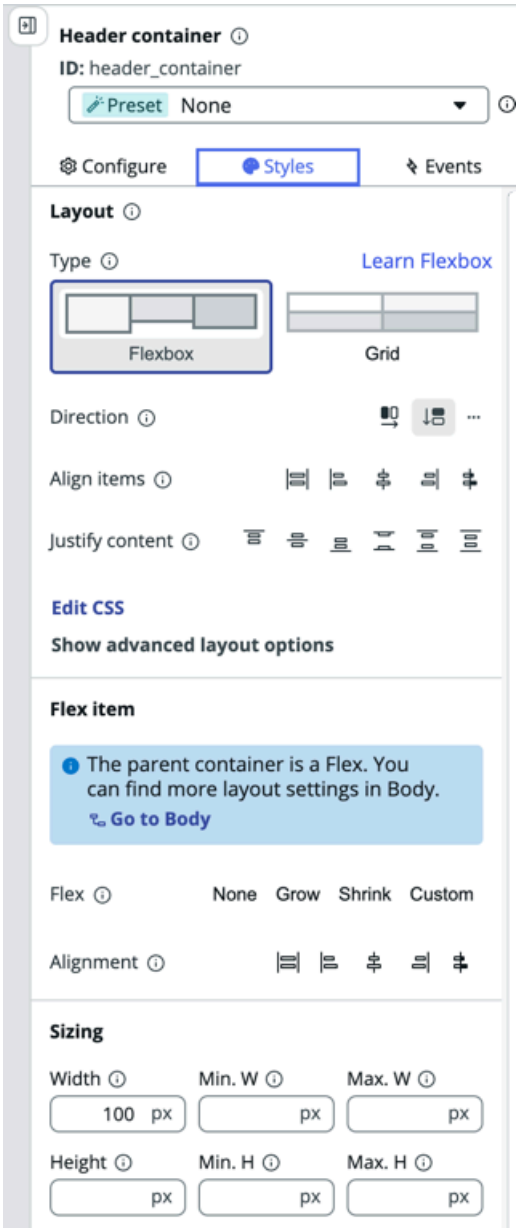
- Select the **Events** tab to configure events that add actions to your components, pages, data resources, and declarative actions. When you add components to your UI Builder page, they are not configured to perform any action. For example, a button component is static and does not do anything until you bind an event action to it, such as deleting a record.



See Manage actions in UI Builder pages for more information.

- Column layouts govern how components are arranged on a page. When you add a column layout to a UI Builder page, you can configure how you want the layout designed. After you add a component to a column in the layout, you can configure the elements in more advanced ways. For example, you can justify the content, align items, and set the height, width, margins, background, borders, and padding of your column layout, columns, and components. CSS grid is the most powerful layout system. CSS Grid is built on top of a two-dimensional grid that gives you power over how you create your pages.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



See Organize components in UI Builder pages for more information.

Open a UI Builder page to preview it

Preview a UI Builder page to see what the page looks like as a web page.

Select the Preview button to test the variant that you're currently working on, including modals, viewports, components, data resources, and dynamic data. Previewing a page variant is useful for seeing how it looks and functions while building an experience.

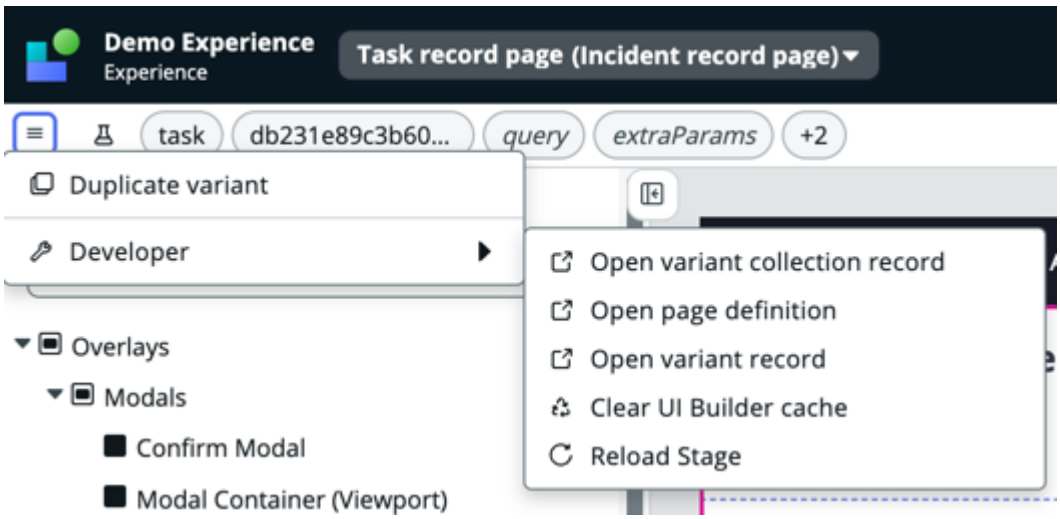
Another method of previewing a page is to request the URL path from the server. Use this method to test if a user sees the variant when they open the page. Select the drop-down arrow next to **Preview** and select **Open URL path** or select **Preview > Open URL path** Open URL path.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

See Learn how to view and test your UI Builder experience for more information.

Developer editing in UI Builder

From the Menu, edit the UI Builder page as a developer on the platform. This option lets you change the platform-level details of your page and takes you out of UI Builder. You must have the proper role as a developer.



Create a page in UI Builder

Create a page in UI Builder for a portal, workspace, or custom application so that you can build a web experience for your users.

Before you begin

Role required: ui_builder_admin

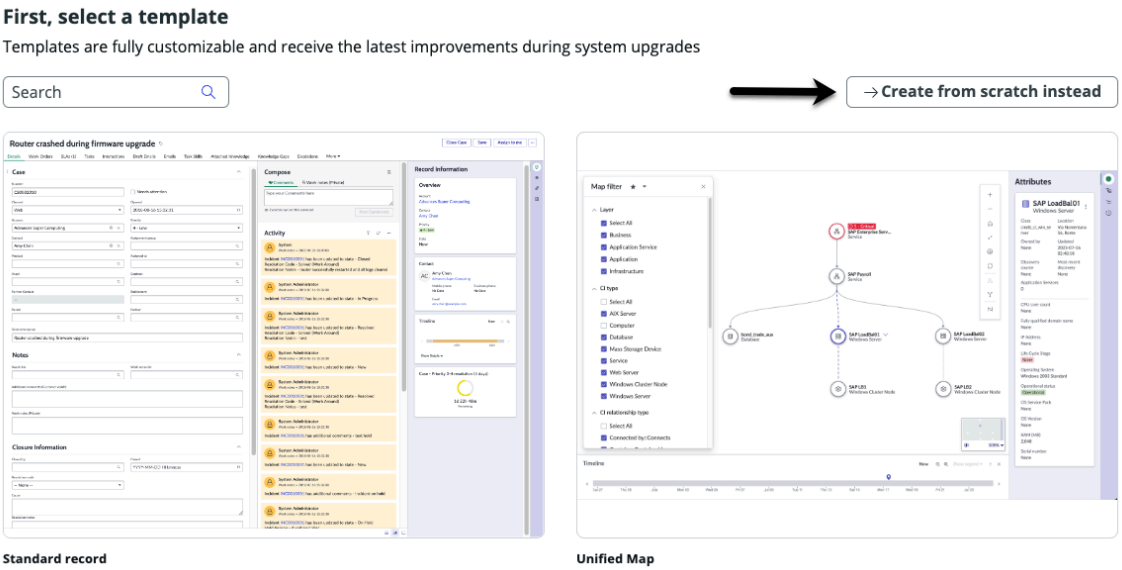
About this task

A page has a collection of components that make up a workspace, portal, or custom application user interface (UI).

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Select the **Create new page** in the center of the screen.
4. Select **Create from scratch instead**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



You can also create pages using page templates, see Create a page from a template for more information.

Note: As of Xanadu Store Release 1, responsive authoring is available when creating a new UI Builder page from scratch. Responsive authoring is not available for existing pages or pages created with a template, however, these types of pages will continue to use the existing, default reflow model. For more information, see Responsive authoring.

5. Enter a unique name for the page in the **Name** field.
6. Specify a path for your page in the **URL Path** field. UI Builder generates a default path based on the name that you provided in the previous step.

   A default path is added based on your page name. You can also create your own path. The path is required and must be unique. The path can include digits (0-9), letters (A-Z, a-z), and a few special characters (`" - "`, `" . "`, `" _ "`, `" ~ "`), with the words separated by a forward slash or hyphen. The **URL preview** shows the path of your page.

   Note: The application scope defaults to the scope that the user is currently in within the ServiceNow AI Platform®. For more information about the application scope, see Learn about security and roles.

7. Select a URL page type in the **Type** drop-down list.
   URL page types help you categorize and find important pages in the experience view. Select from **None**, **Dashboard**, **Knowledge base**, **List**, and **Record** page types. Pages assigned a type other than **None** are filtered more effectively in the experience view.

   Note: A landing page is assigned a **Landing** URL type by default.

8. Select **Continue**.
9. Add required parameters to your page URL.

   a. Click **+ Add**.

   b. Enter the required parameters for your page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

A required parameter is a piece of data that your page requires, such as a sys_id, table, or query. Required parameters are useful for components, because they can bind to the value of the required parameter.



For more information, see Manage UI Builder pages and page variants.

10. **Optional:** Add optional parameters to the URL of your page.

    a. Click **+ Add**.

    b. Enter the optional parameters for your page.
    Unlike required parameters, optional parameters are always name and value pairs that work no matter what order that they are provided in.

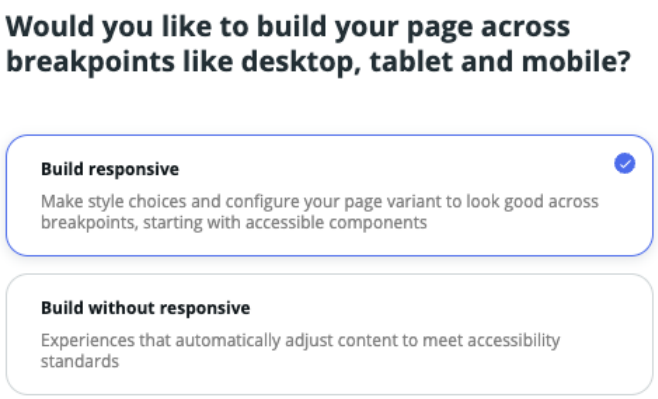


    For more information, see Manage UI Builder pages and page variants.

11. Select **Looks good**.
12. Enter a name for the page variant.
    The form automatically contains **Default** but you can edit the page variant name.
13. **Optional:** Add one or more audiences for this page.
    If an audience you need is not listed, you can choose the Open audiences in platform link to create one.
14. **Optional:** Declare conditions for when to display the page by either using the provided dropdown menus or writing an encoded query string.

    (Optional) For more information on writing encoded queries, see Encoded query strings.
15. Select **Continue**.
16. Select the **Build responsive** option (default) for greater control of how the page appears at different screen form factors or select **Build without responsive** for automatic adjustment.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



For more information, see Responsive authoring.

17. Select **Create** to create your blank page.
    The page opens in Editor view so you can start adding content such as layouts and components. For more information, see Customize UI Builder pages using components.

Create a page from a template

Use a template to create a page based on a pre-defined page template. A page template can help you create pages faster and keep pages consistent within an experience. Page templates may include components, data resources, and a layout.

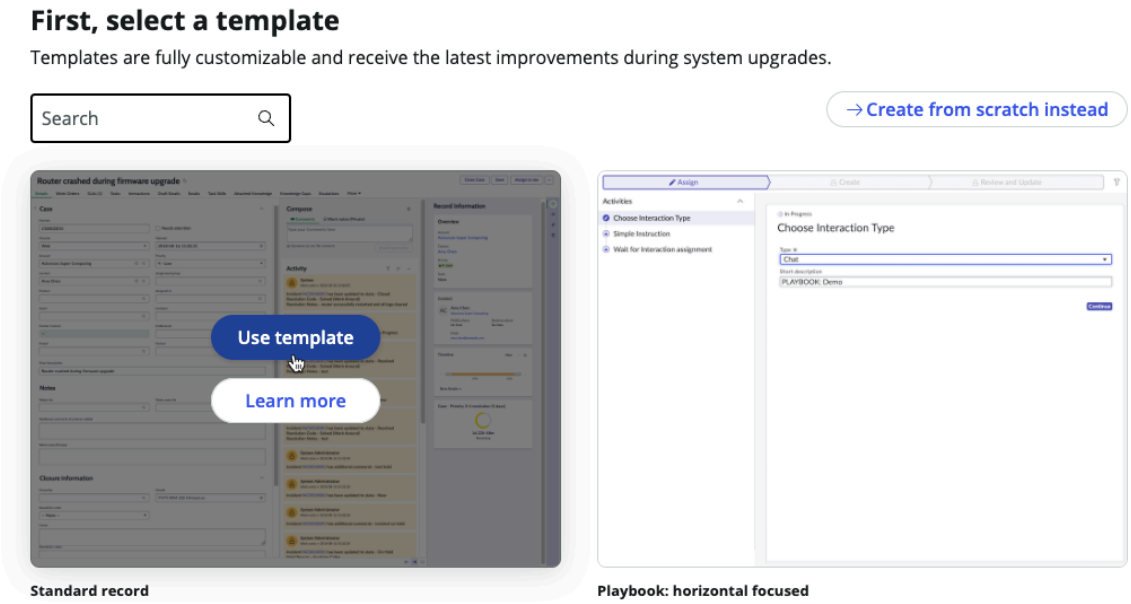
This video shows you how to perform the following procedure.

Before you begin
Role required: ui_builder_admin

About this task
Select a page template when creating a page in your experience. After creating a page from a template, you can customize the page to your needs. Page templates include controllers that can be used with component presets. See Bind data to UI Builder pages using controllers (advanced feature) for more information.

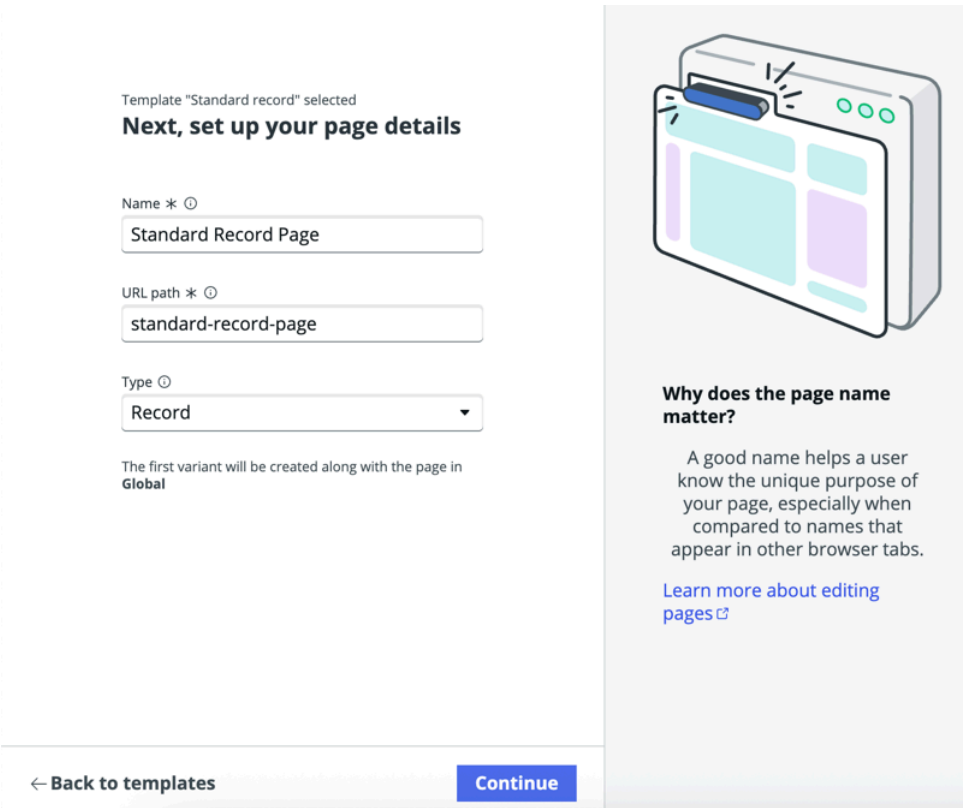
Procedure
1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Select the **Create new page** in the center of the screen.
4. Hover over the template you want to use and select **Use template**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



5. Enter a unique name for the page in the **Name** field.
6. Specify a path for your page in the **Path** field. UI Builder generates a default path based on the name that you provided in the previous step.

   A default path is added based on your page name. You can also create your own path. The path is required and must be unique. The path can include digits (0-9), letters (A-Z, a-z), and a few special characters (`" - "`, `" . "`, `" _ "`, `" ~ "`), with the words separated by a forward slash or hyphen. The **URL preview** shows the path of your page.



Note: The application scope defaults to the scope that the user is in within the ServiceNow AI Platform®. For more information about the application scope, see Learn about security and roles.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

7. Select a URL page type in the **Type** drop-down list. UI Builder automatically assigns a page type based on the template you select, but you can change this value if needed.
   URL page types help you categorize and find important pages in the Experience view. Pages with a type other than **None** are filtered more effectively in the Experience view.
8. Select **Continue**.
9. Review the page parameters included in the template.
   For more information about required and optional page parameters, see Adding Page Parameters in the ServiceNow Developer website.
10. Select **Looks good**.
11. Enter a name for the page variant.
    The first variant you create is named **Default** by default.
12. Add one or more audiences for this page.
    If an audience you need is not listed, you can choose the Open audiences in platform link to create one.
13. **Optional:** Declare conditions for when to display the page (this variant) by either using the provided dropdown menus or writing an encoded query string.

    (Optional) For more information on writing encoded queries, see Encoded query strings.
14. Select **Create** to create your page from template.
    The page you created displays in the **Page** section of your experience. Select **Editor** to start adding components to your page. For more information, see Customize UI Builder pages using components.

Create a page from a legacy template

Use legacy page templates to reuse a page definition, such as record or list page, for pages in your workspace or portal.

This video shows you how to perform the following procedure.

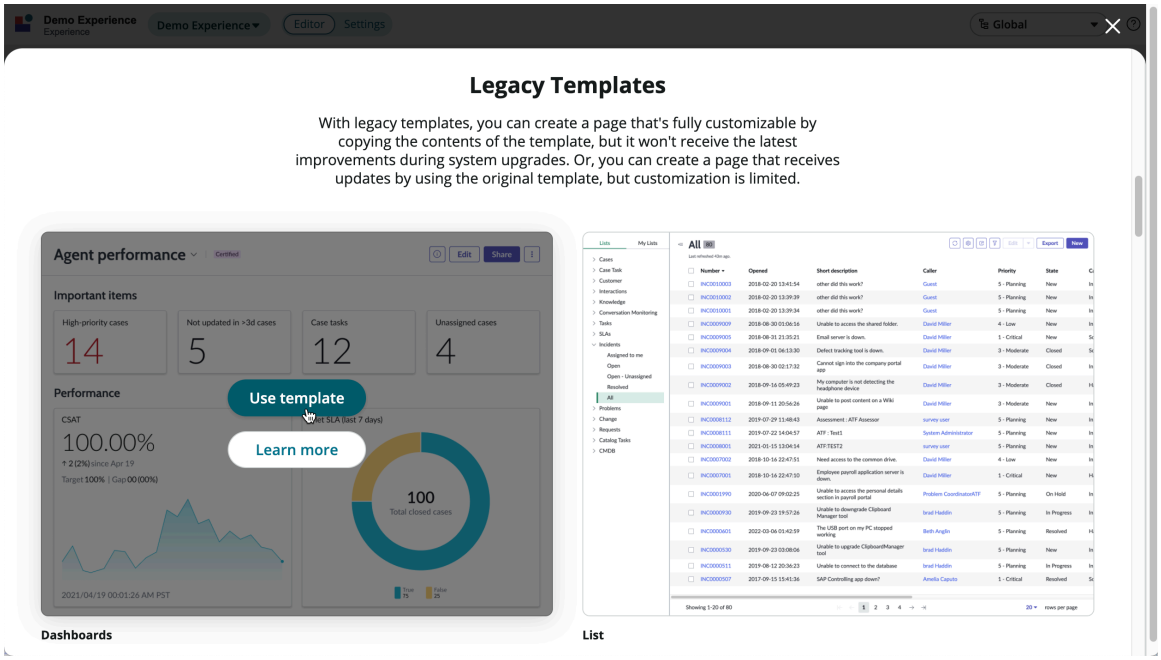
Before you begin

Role required: admin

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Select the **Create new page** in the center of the screen.
4. Scroll down and select **View legacy templates**.
5. Hover over the template you want to use and select **Use template**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



6. Enter a unique name for the page in the **Name** field.
7. Specify a path for your page in the **Path** field. UI Builder generates a default path based on the name that you gave in the earlier step.

   A default path is added based on your page name. You can also create your own path. The path is required and must be unique. The path can include digits (0-9), letters (A-Z, a-z), and a few special characters (`" - "`, `" . "`, `" _ "`, `" ~ "`), with the words separated by a forward slash or hyphen. The **URL preview** shows the path of your page.

   Note: The application scope defaults to the scope that the user is in within the ServiceNow AI Platform®. For more information about the application scope, see Learn about security and roles.

8. Select a URL page type in the **Type** drop-down list. UI Builder automatically assigns a page type based on the legacy template you select, but you can change this value if needed. URL page types help you categorize and find important pages in the Experience view. Pages with a type other than **None** are filtered more effectively in the Experience view.
9. Select how you want to use the base page template.
   Select **Use the original page (customization limited)** if you want to reference the template and its data. Select **Copy contents of the page (fully customized)** to copy the contents of the page template. When you reference a page template, your page automatically updates when you upgrade to a new release. If you copy the page template, your page will not update after upgrade.
10. Select **Continue**.
11. **Optional:** Add required parameters to your page URL.

    a. Click **+ Add**.

    b. Enter the required parameters for your page.
    A required parameter is a piece of data that your page requires, such as a sys_id, table, or query. Required parameters are useful for components, because they can bind to the value of the required parameter.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



For more information about required and optional page parameters, see Adding Page Parameters in the ServiceNow Developer website.

12. **Optional:** Add optional pieces of data that you want to add to the URL of your page.

    a. Click **+ Add**.

    b. Enter the optional parameters for your page.
    Unlike required parameters, optional parameters are always name and value pairs that work no matter what order that they are provided in.



    For more information, see Manage UI Builder pages and page variants.

13. Select **Looks good**.
14. Enter a name for the page variant.
    The form automatically adds **Default** to the page name.
15. Add one or more audiences for this page.
    If an audience you need is not listed, you can choose the Open audiences in platform link to create one.
16. **Optional:** Declare conditions for when to display the page by either using the provided dropdown menus or writing an encoded query string.

    (Optional) For more information on writing encoded queries, see Encoded query strings.

17. Select **Create** to create your blank page.
    The page you created displays in the **Page** section of your experience. Select **Editor** to start adding components to your page. For more information, see Customize UI Builder pages using components.

Edit a page

Edit a page to change the page name, path, type, and parameters.

Before you begin

Role required: ui_builder_admin

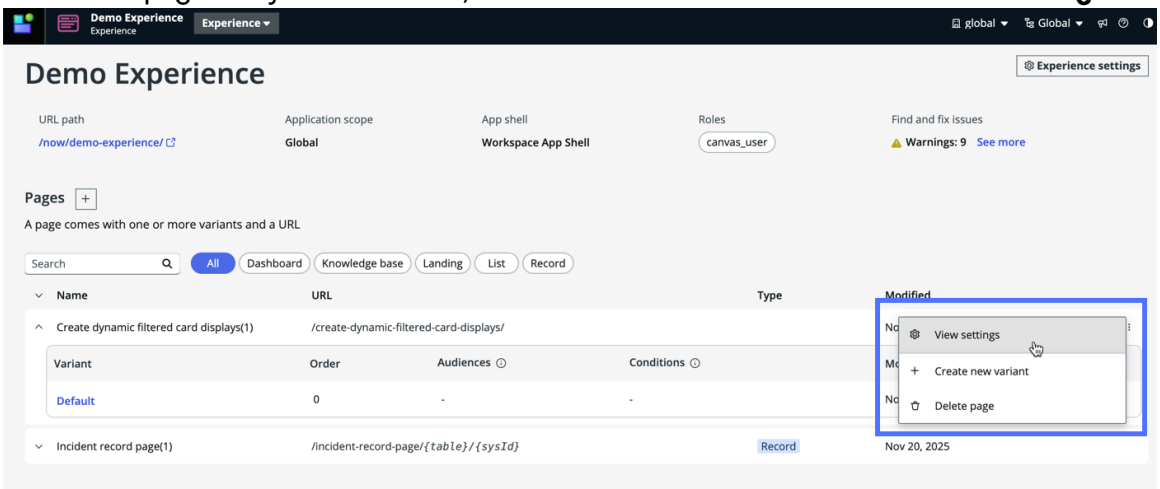
About this task

Edit page settings to change the name, path, and type of your page. This is useful if you decide this information should be changed after you create the page. If your page is in a different application scope, you can choose to edit it in the original scope.

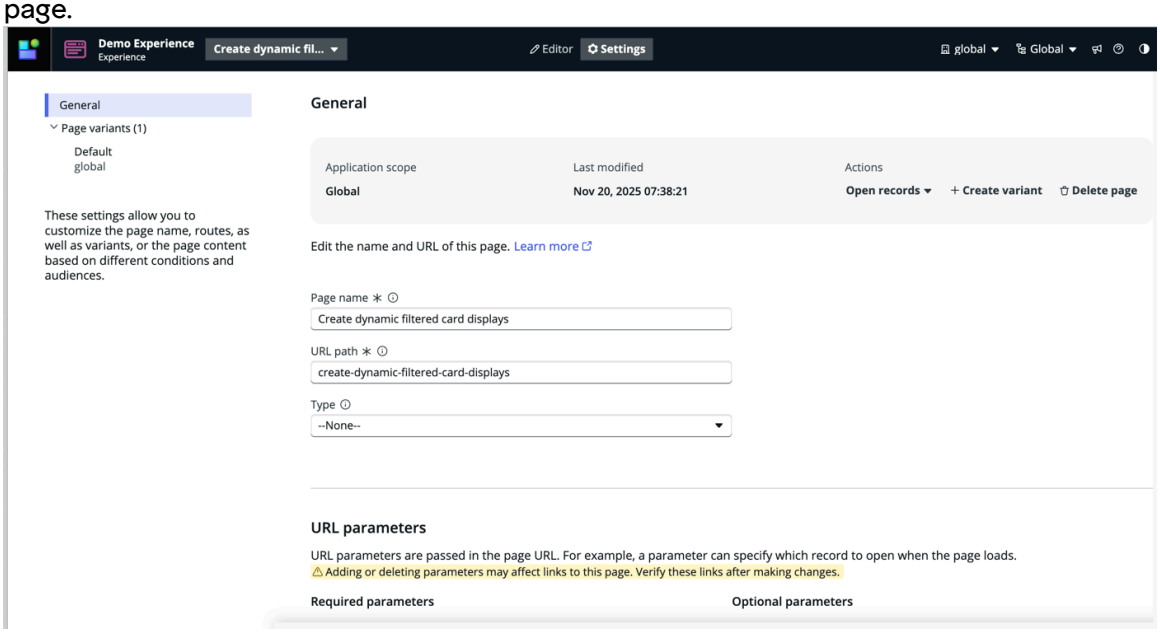
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open the experience with the page that you want to edit.
3. Next to the page that you want to edit, select the More actions icon ⋮ then **View settings**.



4. In the page settings window you can change the name, path, type, and parameters of your page.

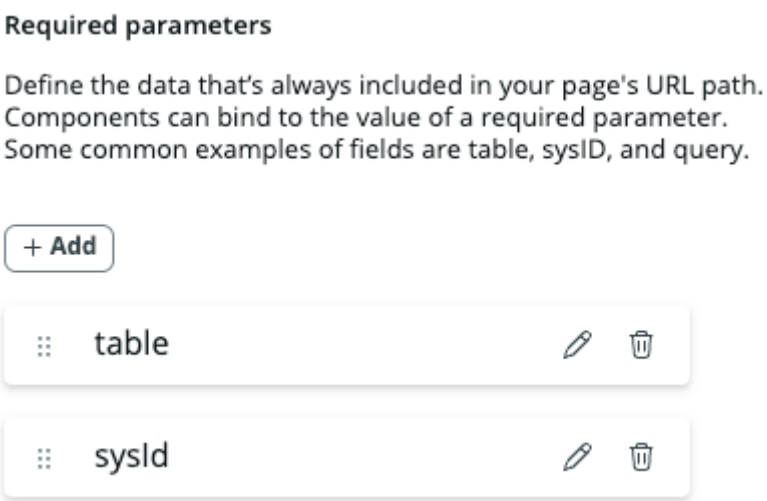


5. **Optional:** Add required parameters to your page URL.

   a. Click **+ Add**.

   b. Enter the required parameters for your page.
   A required parameter is a piece of data that your page requires, such as a sys_id, table, or query. Required parameters are useful for components, because they can bind to the value of the required parameter.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



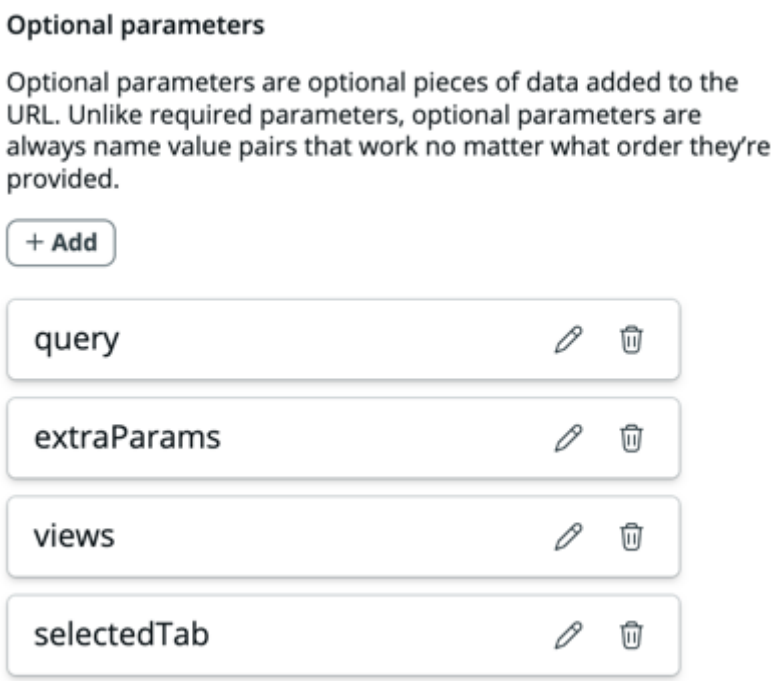
(Optional)

For more information, see Manage UI Builder pages and page variants.

6. **Optional:** Add optional pieces of data that you want to add to the URL of your page.

   a. Click **+ Add**.

   b. Enter the optional parameters for your page.
   Unlike required parameters, optional parameters are always name and value pairs that work no matter what order that they are provided in.



(Optional)

For more information, see Manage UI Builder pages and page variants.

7. Click **Save**.

Add an audience to your UI Builder page

Add one or more audiences to your page or page variant.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

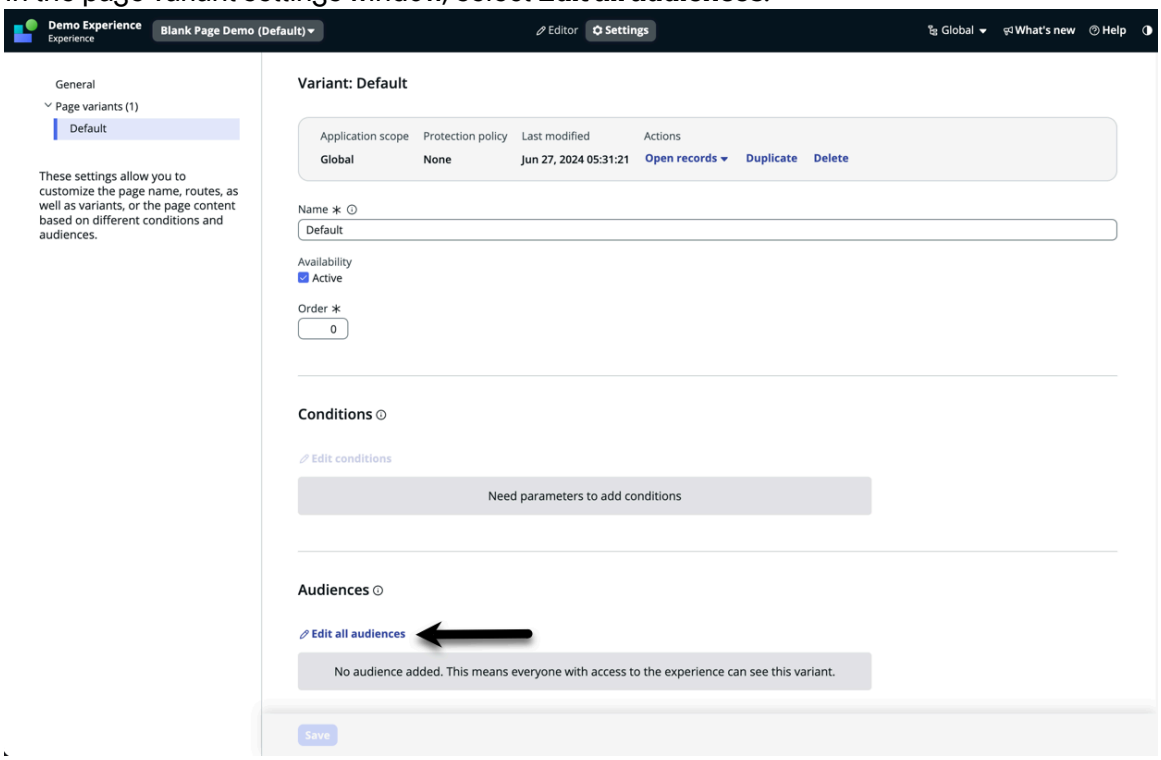
Before you begin

Role required: admin

The *glide.ux.user_criteria_enabled* property needs to be set to **true** to configure access for users based on role, department, group, location, or company. See Enable the user criteria property, for more information.

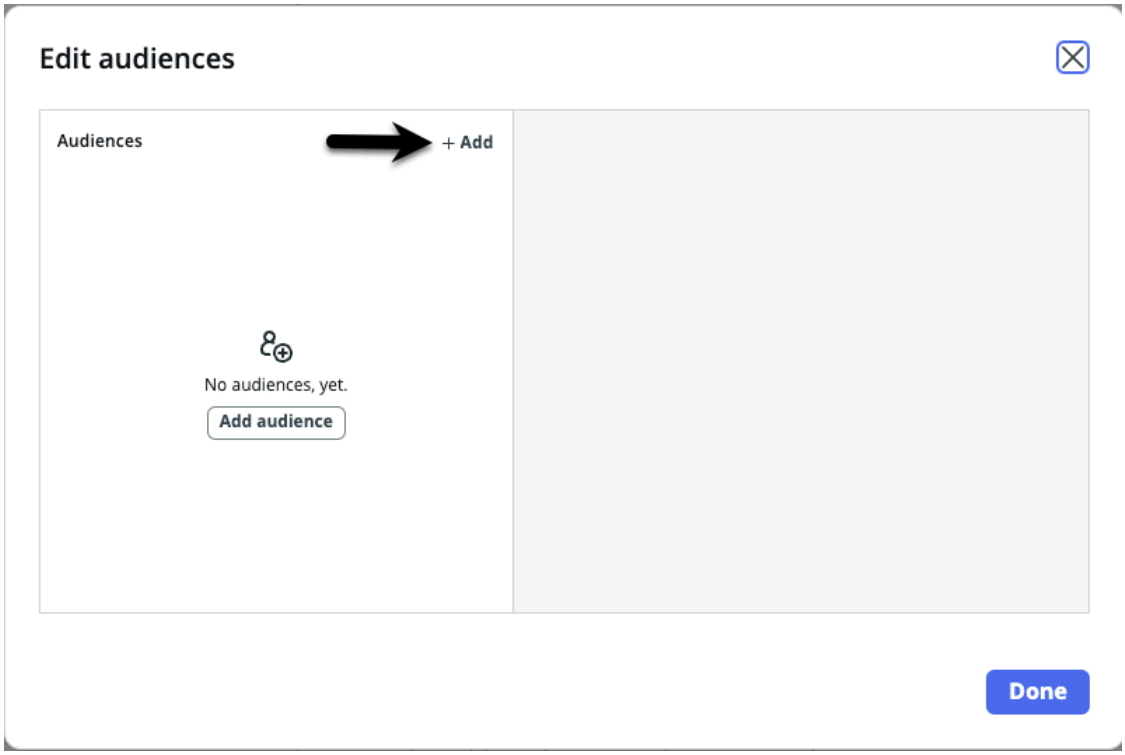
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open the experience with the page that you want to edit.
3. Expand the page to expose the associated variants.
4. Next to the page variant that you want to edit, select the Open settings icon.
5. In the page variant settings window, select **Edit all audiences**.

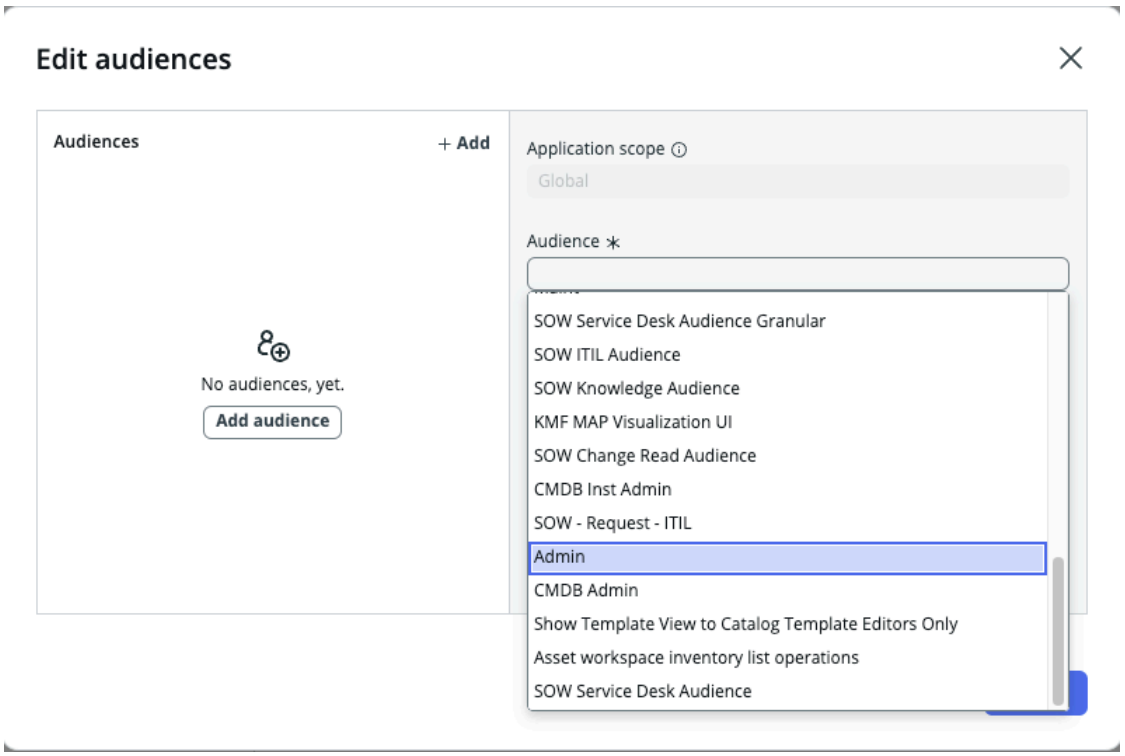


6. Click **+ Add**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



7. Select an audience in the audience field.



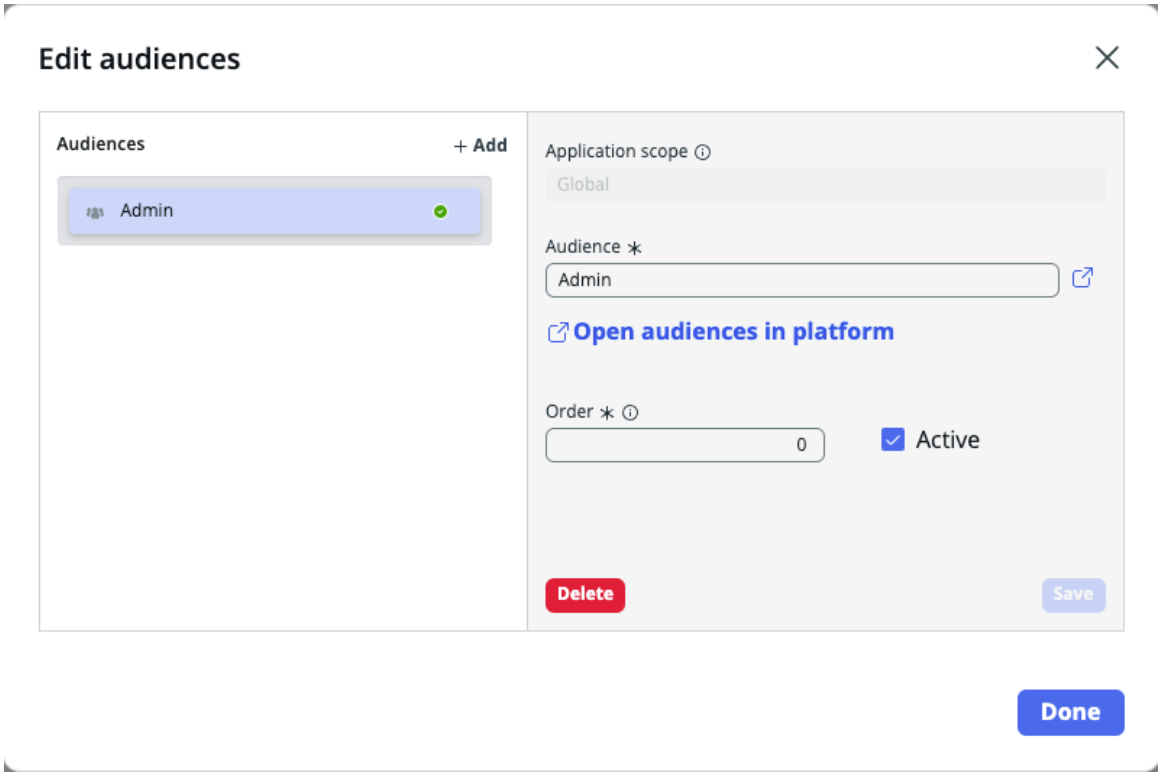
Note: If an audience you need is not listed, you can choose the **Open audiences in platform** link to create one.

8. Set the order of the audiences.

Note: To give higher priority to an audience, enter a lower number. If a user is part of multiple audiences, the audience with the highest priority is used.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

9. Check **Active** if you want the audience to be able view the page.
10. Select **Save**.
    The audience displays in the **Audiences** list.



11. Select **Done**.

Enable the user criteria property

Enable the user criteria property to configure access for users based on role, department, group, location, or company in UI Builder.

Before you begin

Role required: admin

Procedure

1. Navigate to **All** > **sys_properties.list**.
2. Select the *glide.ux.user_criteria_enabled* property.
3. In the **Value** field, specify **true**.
4. Click **Update**.

What to do next

Add audiences to your pages in UI Builder. For more information, see Add an audience to your UI Builder page.

Test values in a page

Add test values to your URL as a way to bring test data into a page.

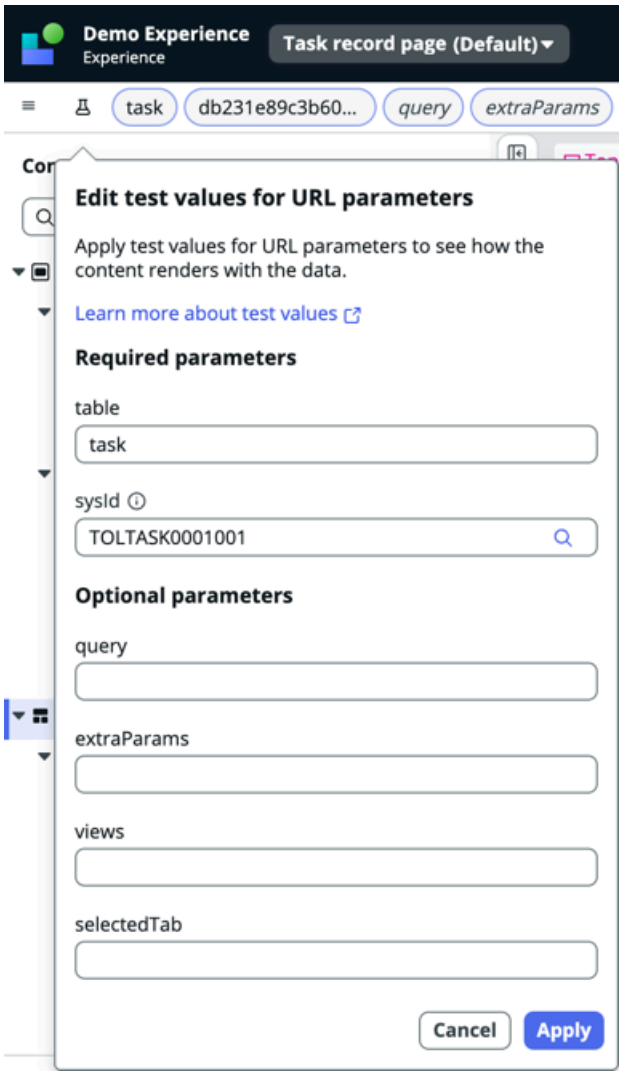
Use test values to give values for required and optional URL parameters as a way to test your page with actual data.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Test values are important because you can use them to simulate how your page behaves when the page gets its required or optional parameters from the URL. A UI Builder URL is not the same as the URL when you preview your page. So, UI Builder needs a way to simulate what happens to the page during different states of the preview URL.

For example, say that you're building a record page that displays a form for a single record. For the record page to load, the page must have a <table> and <sysId> in the URL so that it can get and display the proper record. In UI Builder, you must supply test values for the table and sys_id so that you can see how the page looks when you preview the page.

To get test values to show data, add a data resource, then configure the data resource to bind a record to the test value in the URL. For example, you could add `incident` as a test value.



Create a page variant

A page variant in UI Builder is a variation of a page that exists at the same path that targets different audiences using user criteria.

This video show you how to perform the following procedure.

Before you begin

Role required: ui_builder_admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

About this task

Learn how to create a page variant in UI Builder. A page variant is based on your page with content targeted for a different audience. For example, you can create a homepage for agents, and a variant of that page for managers of those agents. The variant exists at the same URL.

- When someone who is part of the application admins audience accesses the page URL, the user sees a version of the page that has application admin-specific details, such as application metrics and usage details. For the Special Occasions application the page would show how many users have birthdays and work anniversaries and how many do not. The page would also show how many issues have been reported with special occasions.

- When someone who is part of the department managers audience accesses the page URL, the user sees department-related information for the application, such as people in their department with upcoming birthdays and work anniversaries, and whether or not those people are okay with their special occasions being shared.

- When someone who is part of the application users audience accesses the page URL, the user sees a user-specific version of the page with their special occasions that allows them to configure whether or not they want others to see their special occasions.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open the experience with the page for which you want to create a variant.
3. Select the **+** icon in the **Pages** section.



4. Select **Add a variant to a page**.
5. Click **+ Add variant**.



6. Click **Build from scratch instead** or select **Use template** when pointing to the template that you want to use.
7. Enter a name for the variant page.

   The name can be similar to your page name, or what ever you want it to be.

   Note: The application scope defaults to the scope that the user is in within the ServiceNow AI Platform®. For more information about application scope, see Learn about security and roles.

8. Specify one or more audiences for your variant page by selecting **Add an audience**. The audience is important because it lets you assign your page variant to a specific role or group. If an audience you need is not listed, create one by selecting **View all available audiences**, and then selecting **New**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

9. Declare conditions for when to display the page by either using the provided dropdown menus or writing an encoded query string.

   To learn more about declaring conditions, see Control the conditions for a page variant. For more information on writing encoded queries, see Encoded query strings.

10. If creating a page from scratch, select **Continue** and then select the **Build responsive** option (default).
11. Click **Create** to create your page variant.

Edit page variant settings

Edit page variant settings to add additional criteria to determine when the page variant displays to users.

Before you begin

Role required: admin

About this task

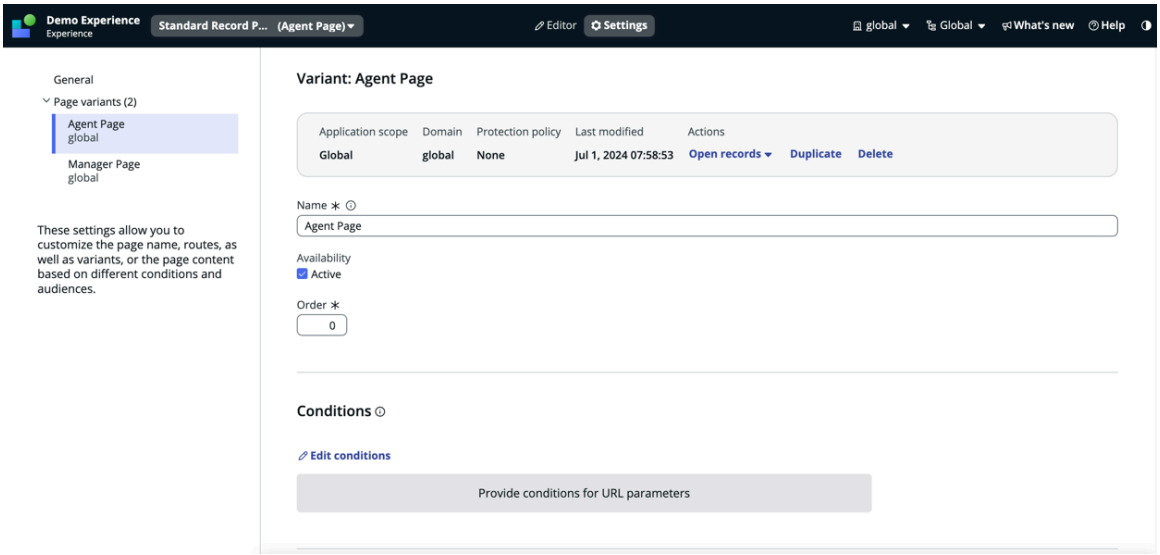
Customize page variant settings to set who sees the page variant. You can add and remove audiences as well as edit conditions to determine when a page variant is shown. Conditions can only evaluate the listed parameters. Conditions are based on setting an order and adding a query string that sets the criteria that must be met for the page variant to display. If you have multiple page variants that all have the same conditions, then the variants go by the order setting. The following task shows how to set the variant conditions and order.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open the experience with the page variant that you want to edit.
3. Expand the page to expose the associated variants.
4. Next to the page variant that you want to edit, select the Open settings icon.
5. On the form, fill in the fields.

Field	Description
Name	Enter a name for the page variant.
Availability	Select **Active** to make the page variant viewable to the selected audience.
Order	Add an order for the condition in the **Order** field. You may want certain conditions to have a higher priority than others. The lower the number, the higher the priority.
Conditions	Set the rules for when your pages are shown. For more information, see Control the conditions for a page variant.
Audiences	Add or remove audiences for the page variant. For more information, see Learn about audiences.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



6. Click **Save**.

Control the conditions for a page variant

Set the conditions and manage the criteria that determine when a page variant is displayed. UI Builder page variants enable you to create different versions of a page to tailor content for a specific audience.

Before you begin

Role required: ui_builder_admin

About this task

If you have multiple page variants that all have the same conditions, the variants go by the order setting.

Procedure

1. Create a page variant.
   For more information, see Create a page variant.
2. Set the conditions for a new variant.

   In this example, you configure a page variant to display when you open a record from the Task table.

   a. On the **First, select a template** screen, locate the **Standard record** template and select **Use template**.

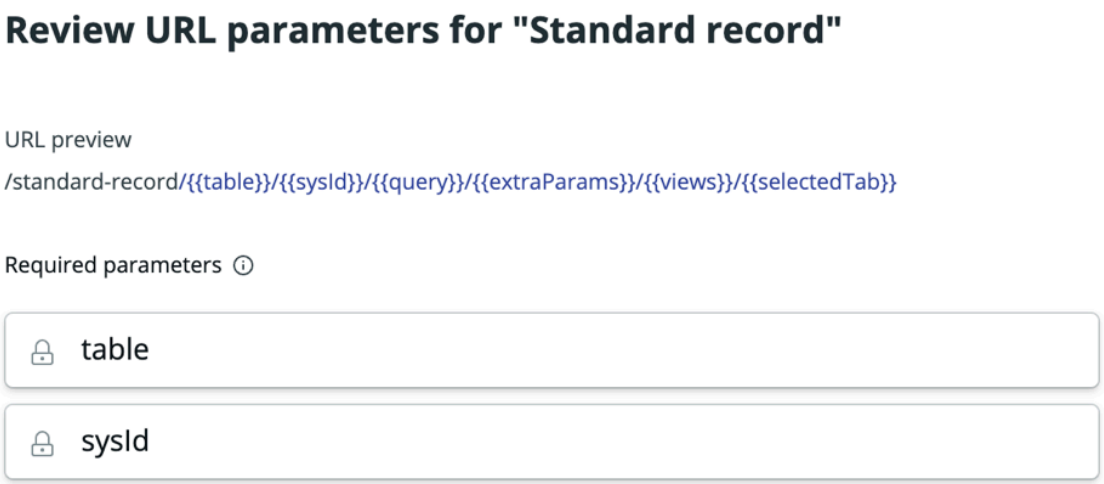
   > **Note:** Conditions can only evaluate parameters that have been added. In this example, the **Standard record** template includes the *table* and *sys_id* parameters.

   b. Enter a name for your page and select **Continue**.

   c. Review the URL parameters and select **Looks good**.

   Notice that *table* and *sysId* are added as required parameters, which appear in the URL preview.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



d. On the **Tell us about your variant** screen, enter a name for your variant.

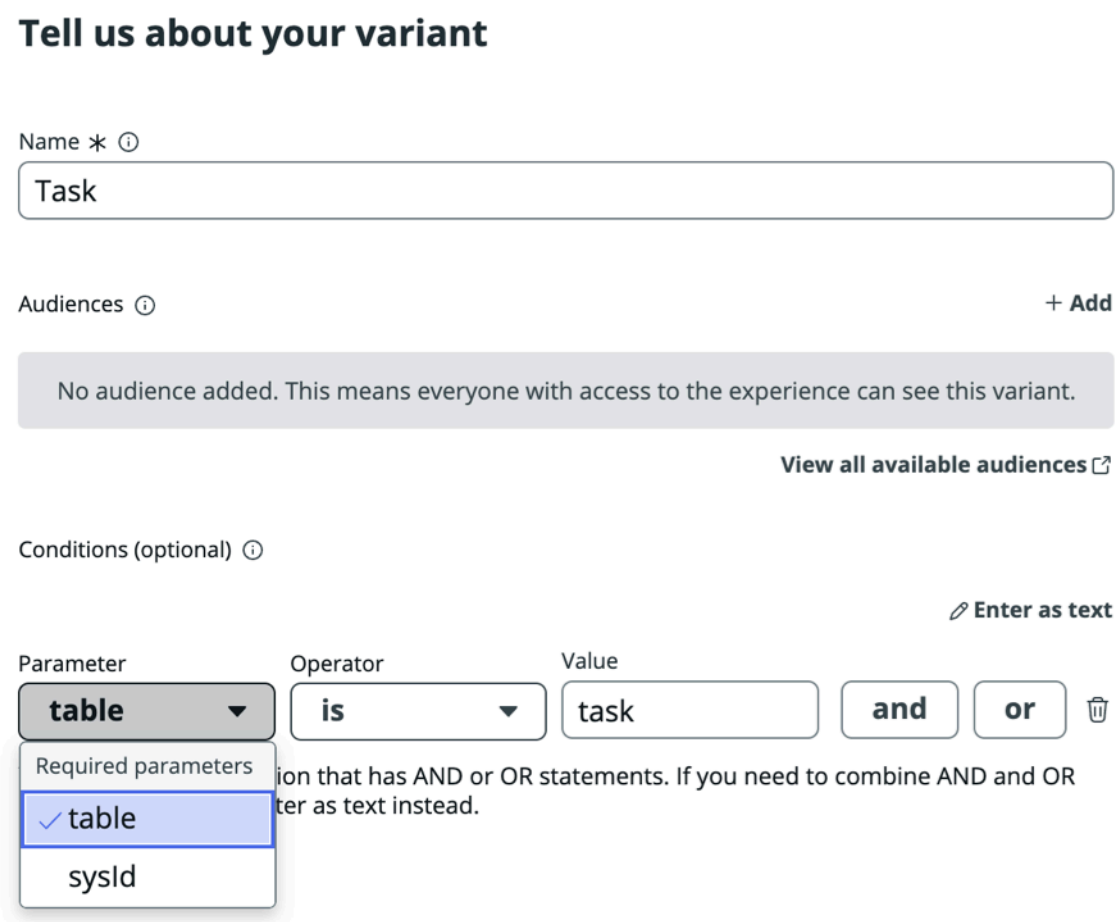
e. In the form section, fill out the fields.

Variant condition form section

Field	Description	Example
Parameter	The aspect or attribute that you want to check.	*table*, *sys_id* **Note:** Conditions can only evaluate the added parameters. However, conditions written for subpages inherit additional outputs from the parent page controllers, providing a wider range of parameters options.
Operator	The rule that compares the parameter to the value. It tells you how the comparison is made.	**is**, **is not**, **starts with**
Value	The number, text, or option input that you want to compare with the parameter.	`"Incident"`, `"12345"`, `"Active"`, `"Change Request"`

Note: The *table* and *sysId* parameters are available options for the **Parameter** field because they're required parameters as a part of the **Standard record** template.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



f. Select **Create**.

3. Review the conditions created for your variant.

a. In the Experience view, find the variant for which you want to view the conditions.

b. Select the **View** button to display the conditions set for that variant.

4. Edit the existing conditions for your variant.

a. From the Experience view, find the variant for which you want to edit the conditions.

b. Select **Settings**.

c. Under Conditions, select **Edit conditions**.

The condition you previously set appears. You can update these fields.

d. On the **Edit variant conditions** screen, next to the condition, select the **and**(and) or **or**(or) button to add another condition and specify the criteria.

For example, by adding the following **AND** condition (**[sysId] [is]** `[abcd1234]`), you configure the page variant to display when you access a record from the Task table with a sys_id of abcd1234.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

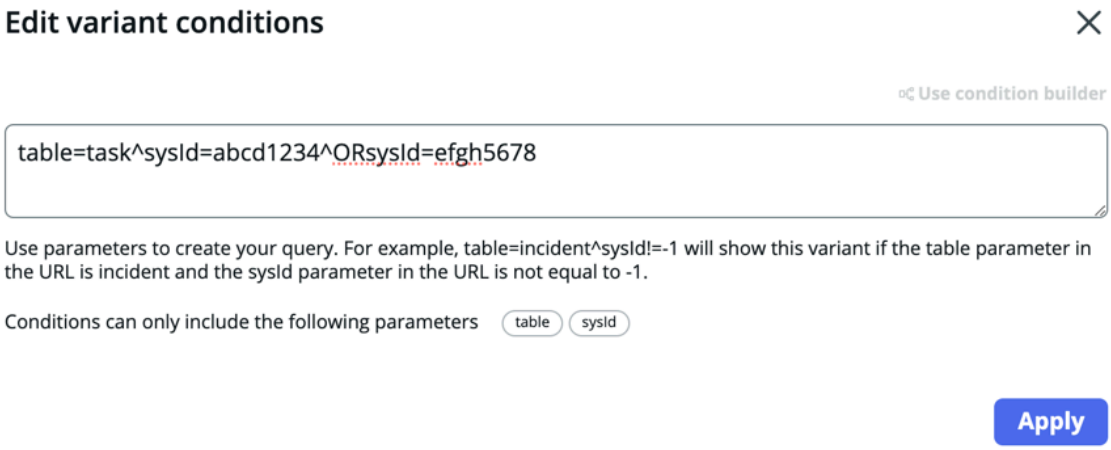
Note: You can add **AND** or **OR** conditions. To mix both types, you must write an encoded query.



e. **Optional:** Select **Enter as text** to write an encoded query that specifies the criteria.

(Optional) The **Edit variant conditions** screen displays the encoded query field, where the conditions you previously set are shown in encoded query form: `table=task^sysId=abcd1234`.

For example, by adding the following 'OR' condition (`^ORsysId=efgh5678`), you set the page variant to display when you access a record from the Task table with a sys_id of either abcd1234 or efgh5678. For more information on writing encoded queries, see Encoded query strings.



Use pages across experiences

Use pages created in one in experience in another without needing to rebuild the page.

Benefits of sharing pages across experiences

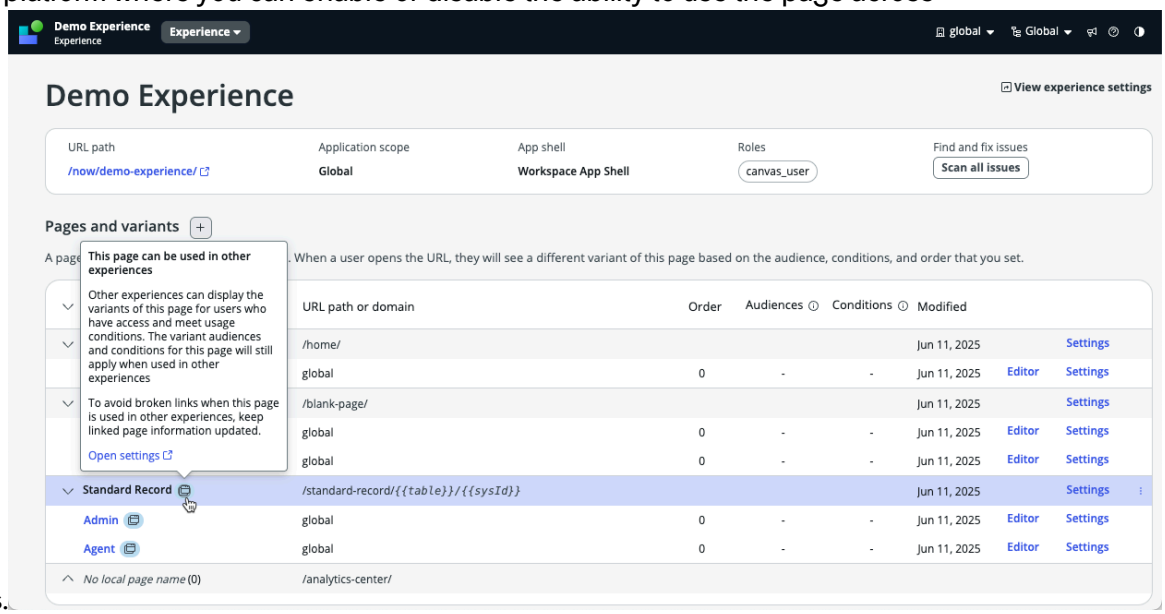
Pages shared across experiences enable administrators and page builders to reuse a single page in multiple experiences without creating duplicate copies. When a page is shared, you can access it from any experience where available, reducing context switching and providing a consistent experience.

A shared page exists as a single source, any updates to that page are automatically reflected everywhere it's used. This simplifies maintenance, reduces the risk of errors from managing multiple copies, and requires less time.

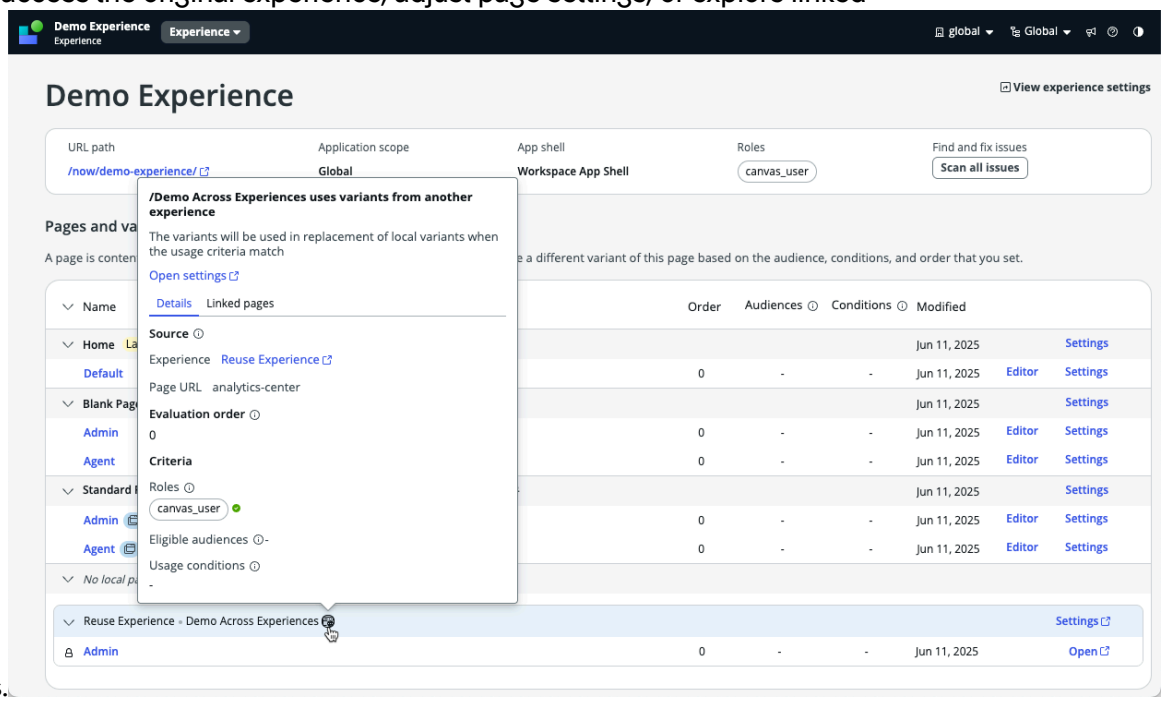
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Viewing shared pages in Experience View

Pages that can be used across experiences display a blue icon containing two pages in UI Builder's experience view. You can select the icon to go to the **UX App Route** settings page on the platform where you can enable or disable the ability to use the page across experiences.



Reusable pages added to your experience display the name of their source experience. Select the icon next to the page title to view details about the page, access the original experience, adjust page settings, or explore linked pages.



Limitations

When sharing a page, you must specify all the linked pages associated with it. **Linked pages** are pages whose routes are tied to the same source experience. This is required so that the experience receiving a shared page has references to all necessary outbound pages.

Note: Shared pages can only be configured in the ServiceNow® platform currently. However, you can view these pages while in the Experience View of UI Builder.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Enable pages to be used across experiences

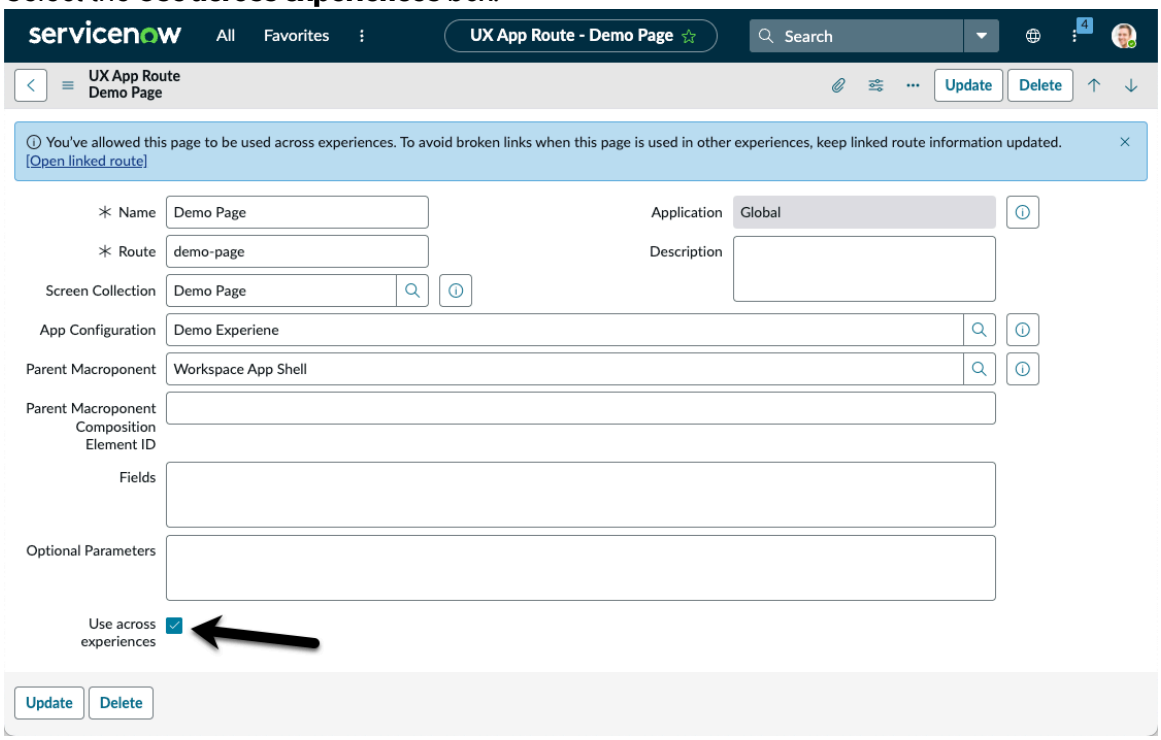
Select pages that you want to share across experiences in UI Builder.

Before you begin

Role required: ui_builder_admin

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Select **Experience settings**.
4. Select **Open app config**.
   The UX App Configuration tab opens.
5. Select the page that you want to use across experiences.
6. Select the **Use across experiences** box.

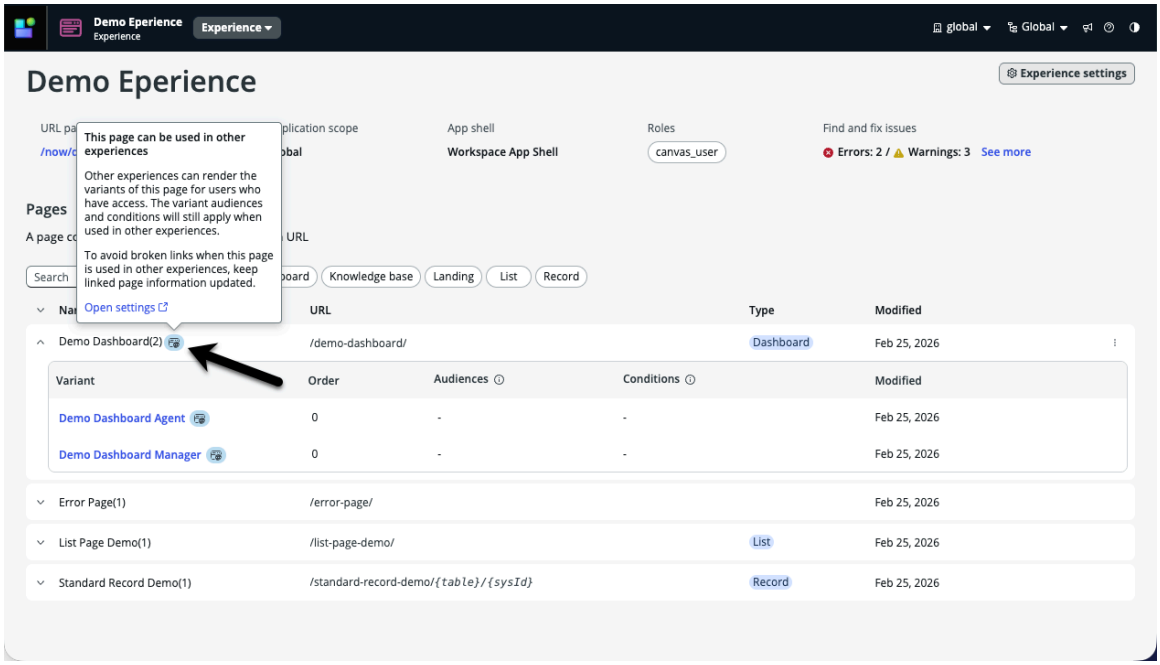


7. Select **Update**.

Result

The selected page displays the shared page icon, indicating it can be used across multiple experiences.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Add shared pages to an experience

Add a page from another experience so that users can access shared content without duplicating pages.

Before you begin

Role required: ui_builder_admin

About this task

Before adding a shared page to your experience, there must be at least one page configured to be used across experiences.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Select the **+** icon next to **Pages**.
4. Select the **Advanced** drop-down.
5. Select **Add a page from another experience**.
6. Select the page that you want to use from the list.
7. Select **Proceed to setup**.
8. Select the variants that you would like to use.

Page variant options	Description
All Variants	Add all variants of the selected page to your experience.
Select specific variants to use	Add specific variants to your experiences of the shared page to your experience.

9. Select **Next**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

10. Review the URL created for the shared page.
11. Select **Next**.
12. In the form, fill in the fields to configure when to display the shared page in your experience.

Create a page collection form

Field	Description
Order of evaluation	The priority number determines the order in which page content from multiple sources is evaluated, with lower numbers taking precedence.
Audiences	Select who can access the shared page.
Conditions	Specify conditions for using shared pages from the source, where the selected variant is determined by the audiences and conditions configured in the source.
Description	Add notes about the configuration.

13. Select **Complete**.
    The shared page displays in your experience along with a label identifying its source experience.

Responsive authoring

Use responsive authoring to create UI Builder pages that adjust smoothly to different form factors (sizes), such as desktop, tablet, and mobile.

Responsive authoring in UI Builder

Responsive authoring enables you to create pages that look good and function properly at any form factor, making it easier for people to interact with the content. For example, a page containing three columns when viewed on a laptop, can adjust to a single column for smaller screens.

UI Builder currently offers three default form factors:

- Desktop (1281 pixels to infinity)
- Tablet (1280 pixels and smaller)
- Mobile (500 pixels to zero)

In addition to these form factors, you can create up to three additional custom breakpoints (widths). For more information, see Create a breakpoint for responsive authoring.

There are different techniques for editing pages so they're usable at different form factors. Use any of the following options:

- Show or hide components
- Change component configuration
- Edit styles
- Rearrange the layout

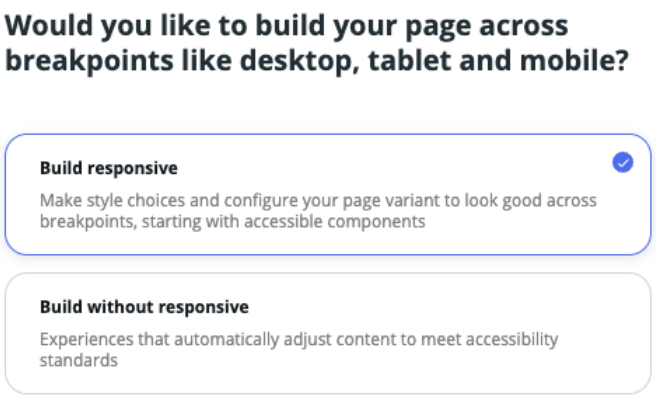
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Responsive authoring and reflow

As of Xanadu Store Release 1, responsive authoring is only available when creating UI Builder pages from scratch. Responsive authoring isn't available for existing pages or pages created with a template. These types of pages continue to use the existing, default reflow model to adjust pages for different screen widths.

Reflow transforms page layouts into a vertical, stacked view automatically without loss of content or functionality when users increase browser zoom to 400%. This adjustment helps users with low vision or who have trouble seeing web content in a browser due to monitor size, device type, poor lighting, or other situations.

All existing pages that were created with templates or from scratch use reflow automatically. Now, when creating pages from scratch in UI Builder, an extra step shows you that responsive authoring is selected by default. You can choose to use reflow instead by selecting **Build without responsive**, however, using the responsive authoring option gives you more control over how pages work and look at different form factors. For more information about creating pages in UI Builder, see Create a page in UI Builder.



For a list of UI Builder components that support reflow, see the **Support for reflow** section of the Next Experience Components release notes.

Application and cascading of changes

All changes made to a form factor are applied to all smaller form factors automatically. For example, any changes made to tablet apply to both tablet and mobile. However, if you then make changes to the mobile form factor, it overrides the cascaded settings from the tablet form factor. This cascading functionality enables you to make specific changes for each smaller form factor so the page looks and functions well.

Note: Changes don't cascade up to larger sizes. For example, changes made to tablet aren't applied to the desktop.

Responsive authoring and controllers

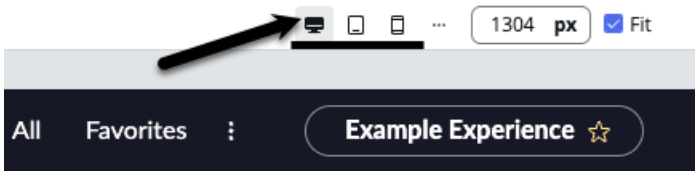
If you add a controller to a page, the controller properties are global and cannot be set per form factor. For more information about controllers, see Bind data to UI Builder pages using controllers (advanced feature).

Using responsive authoring as you create pages

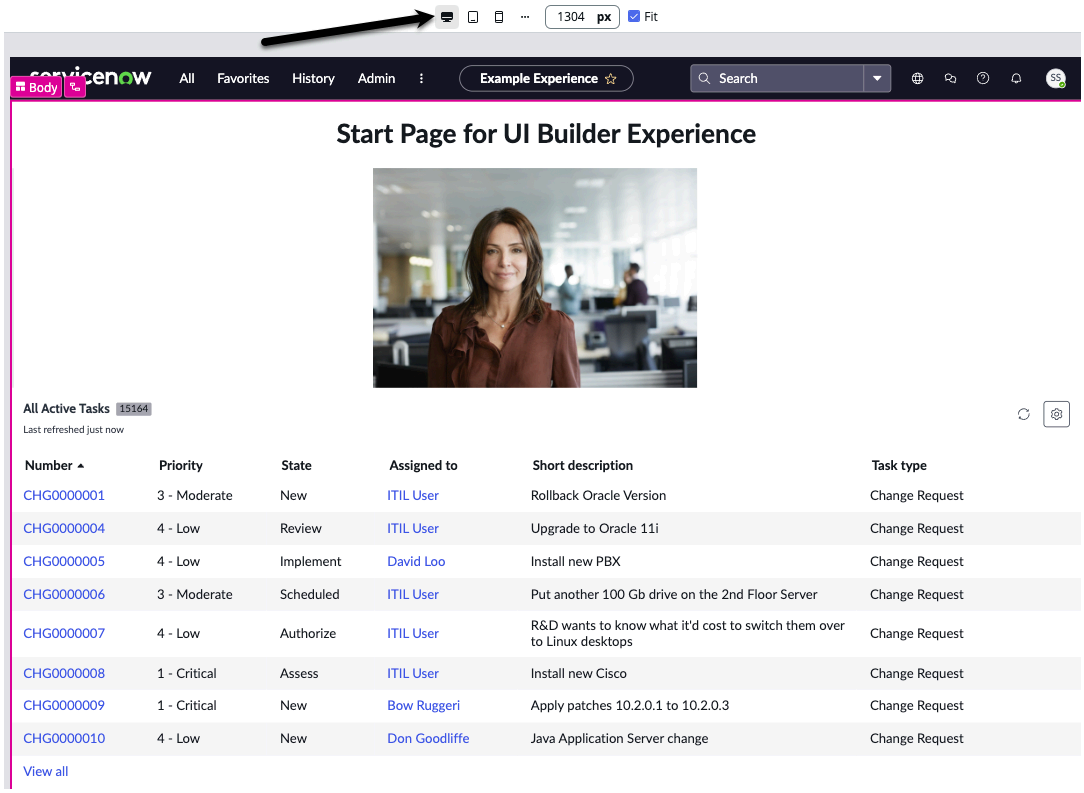
As you build pages, create tailored designs to control the look and feel for different form factors.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

By default on the stage, you build pages for the desktop (1281 pixels to infinity). At any time, select another icon to see the page on the stage at a different form factor, such as tablet or mobile. As needed, edit the page so it displays appropriately at the new form factor.

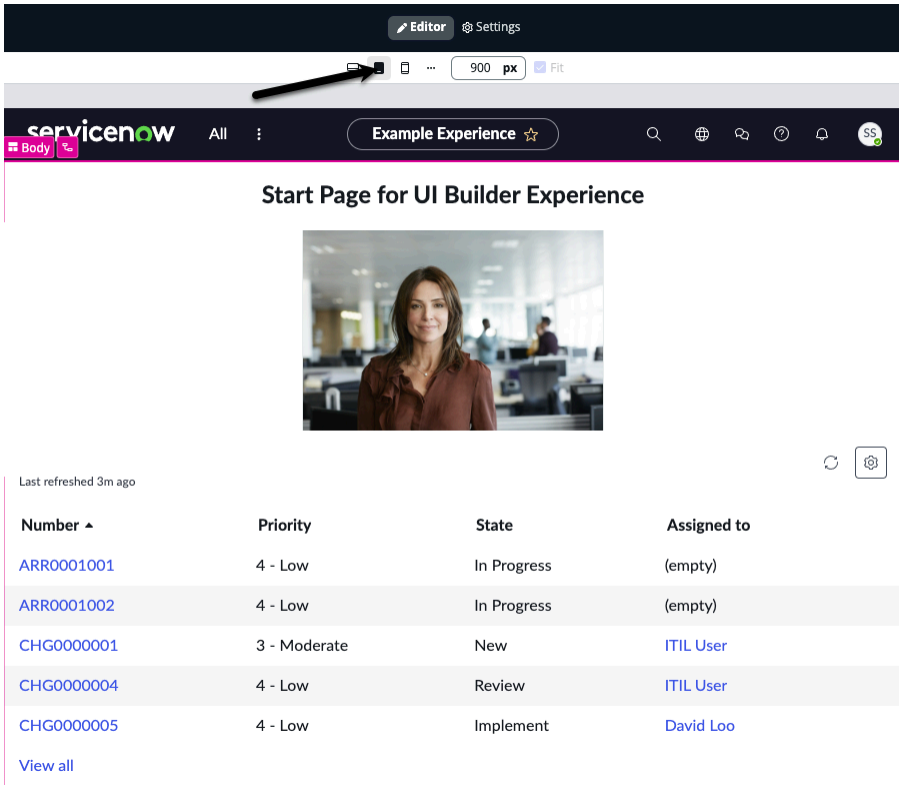


Here's what a sample page looks like at the desktop form factor.



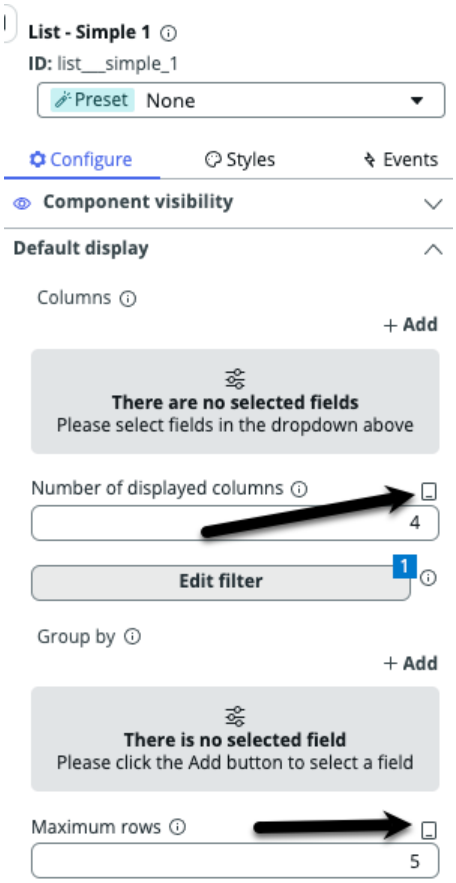
Here's the same page customized for the tablet form factor.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

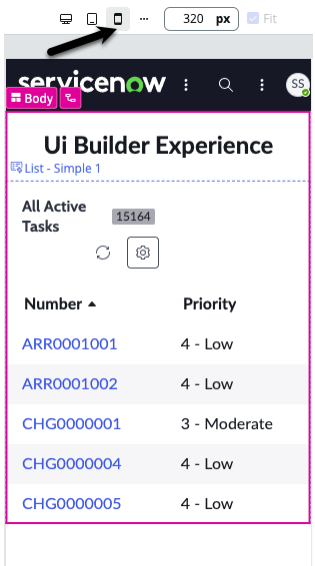


For tablet, the heading and image components have been reduced in size. The simple list component configuration was edited to show four columns instead of six and the maximum number of rows was set to 5. When editing a component for a form factor, the fields that have changed are marked with a cascading icon for that form factor (in this case, tablet). The icon enables you to determine the differences between the form factors for a given page easily.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

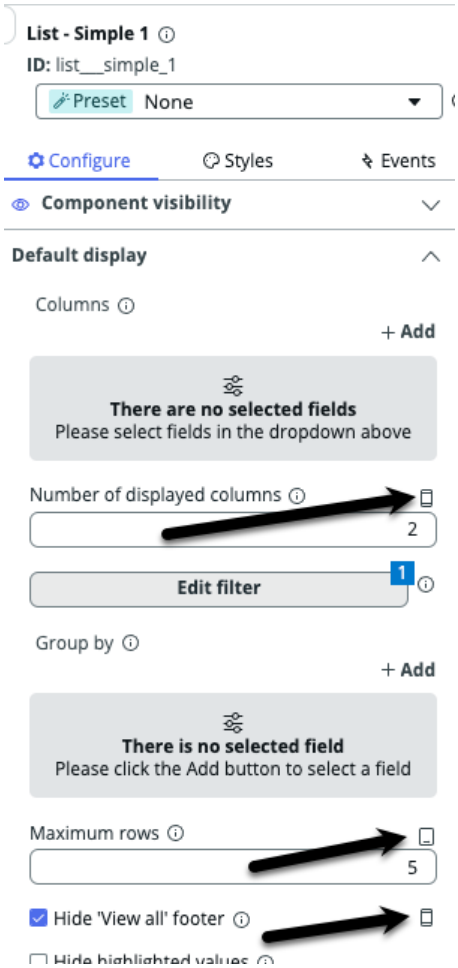


Here's the same page customized for the mobile form factor.

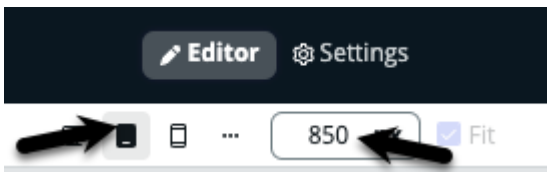


The number of words in the heading was reduced and the image component was hidden. The simple list component configuration was edited to show two columns instead of four and the option to hide the "view all" footer option was selected. The two fields customized specifically for mobile have a mobile cascading icon next to them. The **Maximum rows** field still retains the tablet cascading icon because that field is inheriting the tablet setting and wasn't customized for mobile.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



In addition to the default form factors in the editor view, at any time you can manually enter a number in the pixel field. The stage adjusts to the nearest form factor based on range. For example, entering 850 pixels adjusts the stage to the tablet form factor because its range is defined as 1280 to 501 pixels.



Note: Form factors and the ability to enter a pixel width are also available when previewing a page. For more information, see Learn how to view and test your UI Builder experience.

Adjusting component visibility in responsive authoring

When creating pages with responsive authoring for different form factors, learn how to show or hide components by completing steps to hide an image at mobile size.

Before you begin

Role required: ui_builder_admin

About this task

A method of increasing page usability is to hide components or show a different component across breakpoints. There are countless ways to use component visibility, but a few examples include:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Use multiple button components horizontally across the top of a page for desktop and tablet size, but at mobile size, hide the buttons and show the dropdown component.
- Use a smaller button size for mobile size, but check that the buttons are large enough for users to press with a thumb. Consider changing the placement of buttons for mobile as well.
- Use images in desktop and tablet pages, but hide images, especially larger ones, in mobile.

In this procedure, show buttons across the top for the desktop and tablet form factors, but hide the buttons and use the dropdown component for mobile.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create > Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create a page from scratch with responsive authoring.
   For more information about how to create a page, see Create a page in UI Builder.
4. At the top, check that the desktop form factor icon is selected.



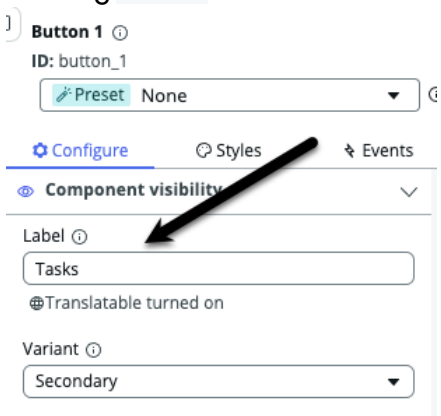
5. Add a column layout by selecting **+ Add content** under **Body** in the content tree.
6. On the **Layouts** tab, select **Single column**.
7. Add the first button component.

   a. Select the **+** icon in the center of the new column on the stage.

   b. On the **Components** tab, find and select the **Button** component.

   c. In the configuration panel, select **None - Configure the component manually**.

   d. In the configuration panel, on the **Configure** tab, replace the default text in the **Label** field by entering `Tasks`.



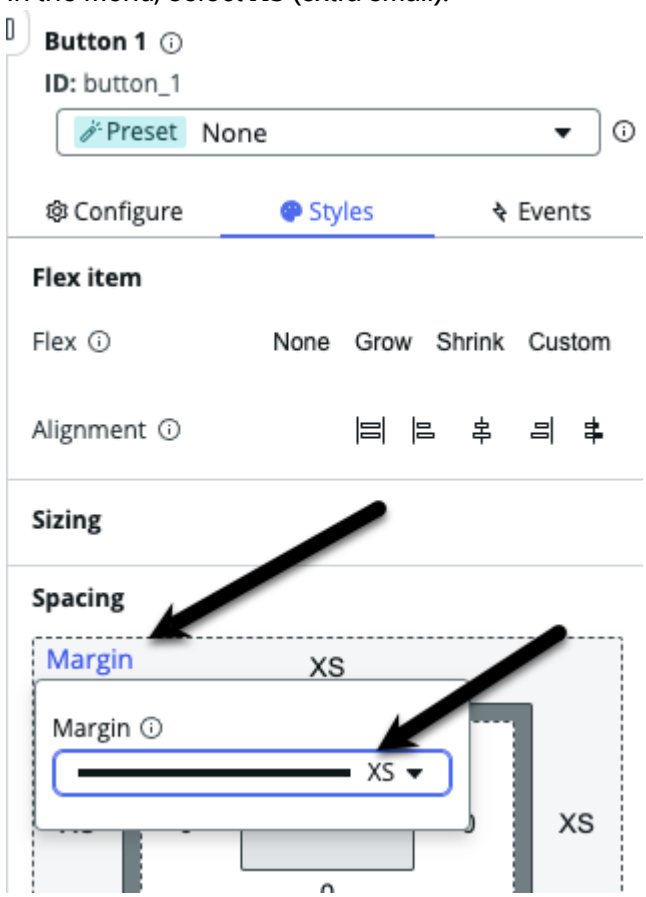
   e. Select **Save**.

8. Add two more buttons by repeating step 7 twice and in 7d name the buttons `Incidents` and `Changes`.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

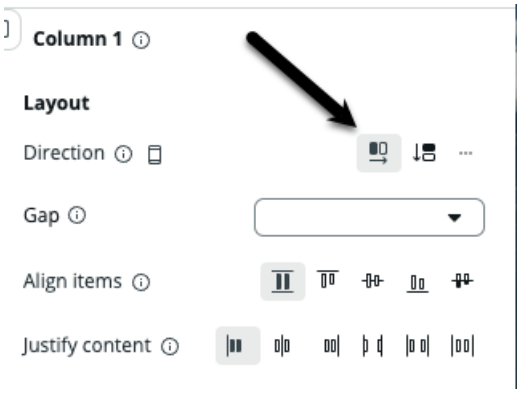
Note: To add the buttons, point to **Button 1** in the content tree, select the menu icon, and then select **Add after**.

9. Check that the content tree and page are accurate.
10. Add some space around each button.
    a. In the content tree, select **Button 1**.
    b. In the configuration panel, select the **Styles** tab.
    c. In **Spacing**, select **Margin**.
    d. In the menu, select **XS** (extra small).



    e. Select **Save**.
11. Add space around the other two buttons by repeating step 10 for each.
12. Change the buttons from the vertical direction to the horizontal direction.
    a. Select **Column 1** in the content tree.
    b. On the configuration panel, in **Direction**, select the row icon.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



c. Select **Save**.

The page looks good at desktop width.

13. Select the tablet form factor icon.

The spacing and position of the buttons also looks good at tablet width.

14. Select the mobile responsive authoring icon.

The buttons use almost all the horizontal space at the smaller size.

15. Add the dropdown component.

a. Select the desktop form factor at the top of the stage.

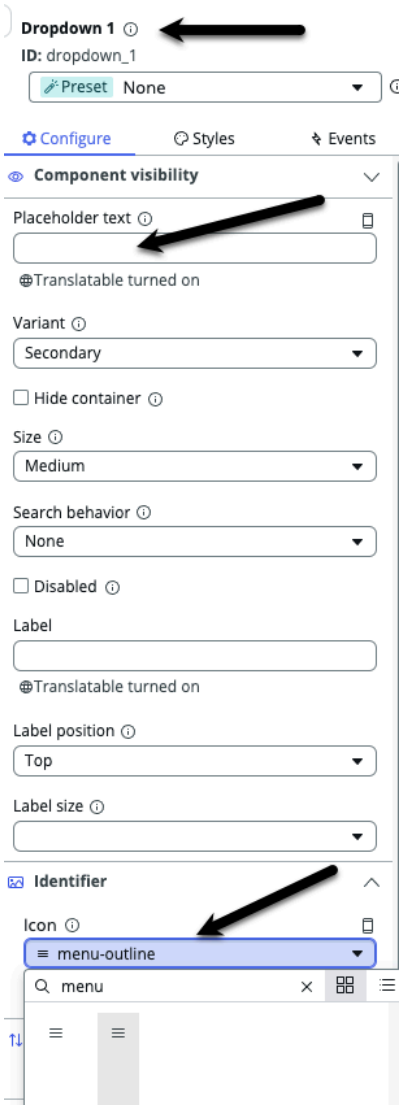
b. Point to **Button 3** in the content tree, select the menu icon, and then select **Add after**.

c. On the **Components** tab, find and select the **Dropdown** component.

d. In the configuration panel, select **None - Configure the component manually**.

e. In the configuration panel, on the **Configure** tab, remove the default text in **Placeholder text** and select **menu-outline** in **Icon**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



f. Select **Save**.

16. Hide the dropdown component at the desktop and tablet form factors.

a. In the content tree, select **Dropdown 1**.

b. In the configuration panel, on the **Configure** tab, select **Component visibility** to show the options.

c. Select **Hide component**.

d. Select **Save**.

17. At the mobile form factor, hide the button components.

a. Select the mobile form factor icon at the top of the stage.

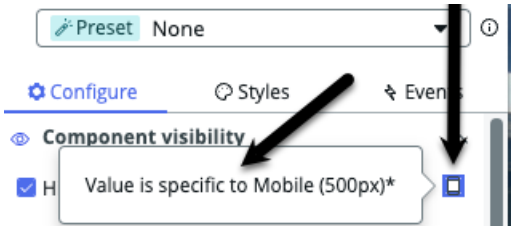
b. In the content tree, select **Button 1**.

c. In the configuration panel, on the **Configure** tab, select **Hide component**.

d. Repeat steps b-c for **Button 2** and **Button 3**.

e. Select **Save**.
Two locations show that the button components are hidden at mobile size. First, in the configuration panel, next to the **Hide component** option, select the mobile icon. A message is displayed to confirm that the value (hide component selected) is for the mobile size.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Second, in the content tree, the three button components are displayed with the hidden icon.

18. At the mobile form factor, show the dropdown component.
    a. In the content tree, select **Dropdown 1**.
    b. In the configuration panel, on the **Configure** tab, select **Hide component** to clear the field.
    c. Select **Save**.
19. Select **Preview**.
    The mobile form factor is displayed with the menu icon. If you select the arrow, the default options are displayed (List item 1, List item 2, and so on). You would configure the dropdown component to display **Tasks**, **Incidents**, and **Changes** just like the horizontal buttons available at the tablet and desktop form factors.
20. Select the desktop and tablet form factor icons at the top.
    The desktop and tablet form factors only show the horizontal buttons, not the menu icon.
21. Close the preview overlay by selecting the **X**.

Configuring components for responsive authoring

When creating pages with responsive authoring for different form factors, learn how adjusting some component configuration options can make pages look and work better at smaller sizes.

Before you begin

Role required: ui_builder_admin

About this task

A method of increasing page usability is to tailor the look and feel of components using different configurations across breakpoints. There are countless ways to use component configuration. In this procedure, edit several configuration options for the List component to make it more suitable for smaller form factors.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create > Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create a page from scratch with responsive authoring.
   For more information about how to create a page, see Create a page in UI Builder.
4. At the top, check that the desktop form factor icon is selected.

5. On the stage, select **+ Add content**.
6. On the **Layouts** tab, select **Single column**.
7. Add the list component by selecting the **+** icon in the center of the new column.
8. On the **Components** tab, find and select the **List** component.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

9. In the configuration panel, select **None - Configure the component manually**.
10. View the page at the desktop form factor.
11. Select the mobile form factor icon.

    At the mobile form factor, the list component is crowded and contains both vertical and horizontal scroll bars.

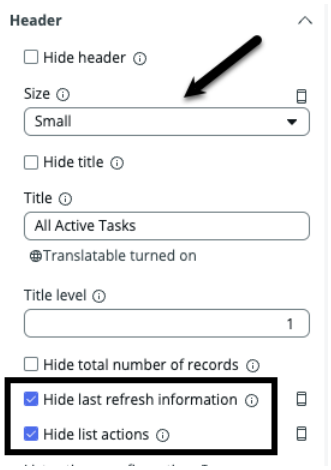
    Make some simple configuration changes to improve the usability and appearance for the mobile form factor.

12. Select **List 1** in the content tree.
13. In the configuration panel, on the **Configure** tab, find the **Number of displayed columns** field and enter 1.
    Only the **Number** column is displayed. The horizontal scroll bar was removed.
14. In the configuration panel, on the **Configure** tab, scroll down to make the following changes in the **Header** section:

    a. Change the **Size** to **Small**.

    b. Select the **Hide last refresh information** option.

    c. Select the **Hide list actions** option.

    As you made each change, the stage updated automatically. The header is a smaller size, the refresh time was removed, and the list actions icon was removed.

    d. Select **Save**.

15. On the stage, scroll to the bottom of the mobile view.
16. In the configuration panel, on the **Configure** tab, scroll down and make the following changes in the **Pagination** section (if necessary, select **Pagination** to expand the field):

    a. Select the **Hide range** option.

    b. Select the **Hide row count** option.

    c. Select the **Hide rows per page selector**.

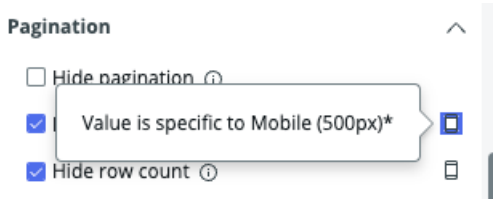
    d. Select **Save**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The range, row count, and rows per page selector were removed. (The row count is still available at the top of the mobile screen).

In the configuration panel, the options and fields you edited are marked with the mobile cascade icon.

17. Select the mobile cascade icon next to any option or field to confirm that the value is for the mobile form factor.



18. Select the tablet and the desktop form factor icons at the top to confirm that no changes were made to the larger form factors.

Adjust styles for responsive authoring

When creating pages with responsive authoring for different form factors, learn how to change style options to increase the usability of the page at smaller sizes.

Before you begin

Role required: ui_builder_admin

About this task

A method of increasing page usability is to tailor components using different styles across breakpoints. There are numerous options for editing styles. In this procedure, edit the margin around the stylized text component and reduce the text size so it uses less space on smaller form factors.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create > Experience**. See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create a page from scratch with responsive authoring.
   For more information about how to create a page, see Create a page in UI Builder.
4. At the top, check that the desktop responsive authoring icon is selected.

5. On the stage, select **+ Add content**.
6. On the **Layouts** tab, select **Single column**.
7. Add the stylized text component.

   a. Select the **+** icon in the center of the column on the stage.

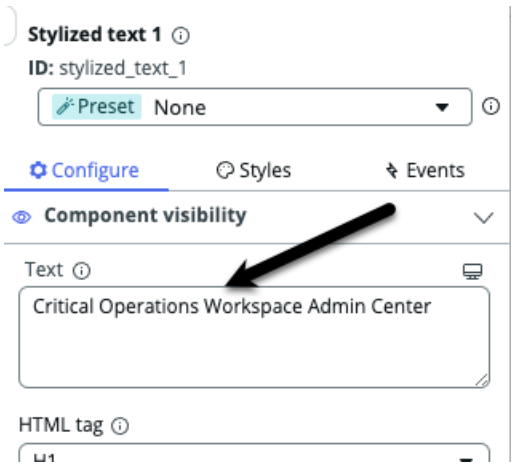
   b. On the **Components** tab, find and select the **Stylized text** component.

   c. In the configuration panel, select **None - Configure the component manually**.

   d. In the configuration panel, on the **Configure** tab, double-click (or use the keyboard shortcut) in the **Text** field to select the default text.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

e. Replace the default text by entering `Critical Operations Workspace Admin Center`.



f. Select **Save**.

8. Add some space around the stylized text component and center it.

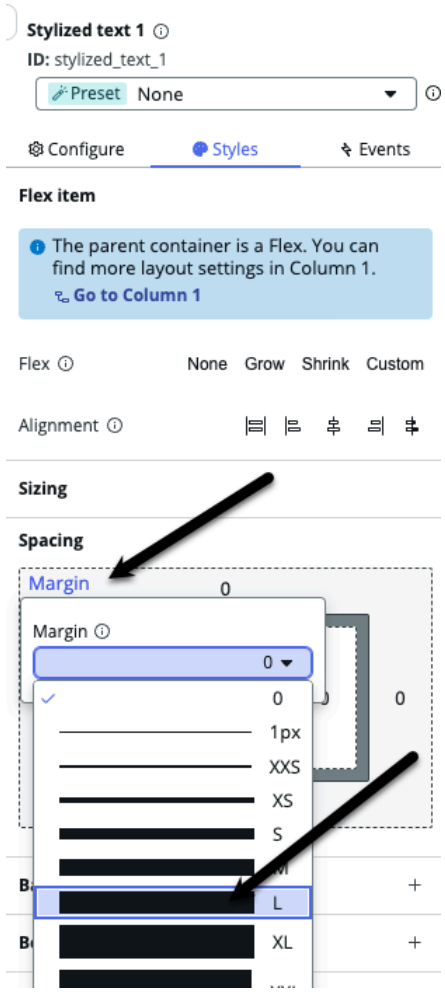
a. In the content tree, select **Stylized text 1**.

b. On the configuration panel, select the **Styles** tab.

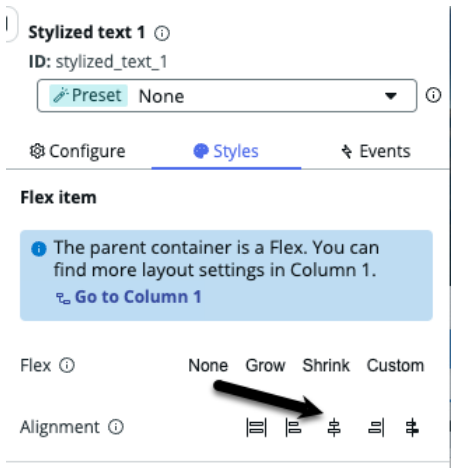
c. In **Spacing**, select **Margin**.

d. Select **Large** in the list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



e. In **Alignment**, select the center icon.



f. Select **Save**.

The stage updates automatically, showing the additional white space around the stylized text component and centering it.

9. Select the tablet form factor icon.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The spacing and heading size of the stylized text component looks good at tablet size.

10. Select the mobile form factor icon.

The stylized text component uses a great deal of vertical space at the smaller size.

11. Remove the extra margin space and make the heading smaller just for the mobile form factor.
    a. In the content tree, select **Stylized text 1**.
    b. On the configuration panel, select the **Styles** tab.
    c. In **Spacing**, select **Margin**.
    d. Select **0** in the list.
    e. Select the **Configure** tab.
    f. In **HTML tab**, select **H3**.
    g. Select **Save**.
    The stage updates automatically showing that the stylized text component uses less space.
12. Select the tablet and the desktop form factor icons at the top to confirm that no changes were made to the larger form factors.

Adjust layout for responsive authoring

When creating pages with responsive authoring for different form factors, learn how to adjust the layout to improve the look and feel of the page at smaller sizes.

Before you begin

Role required: ui_builder_admin

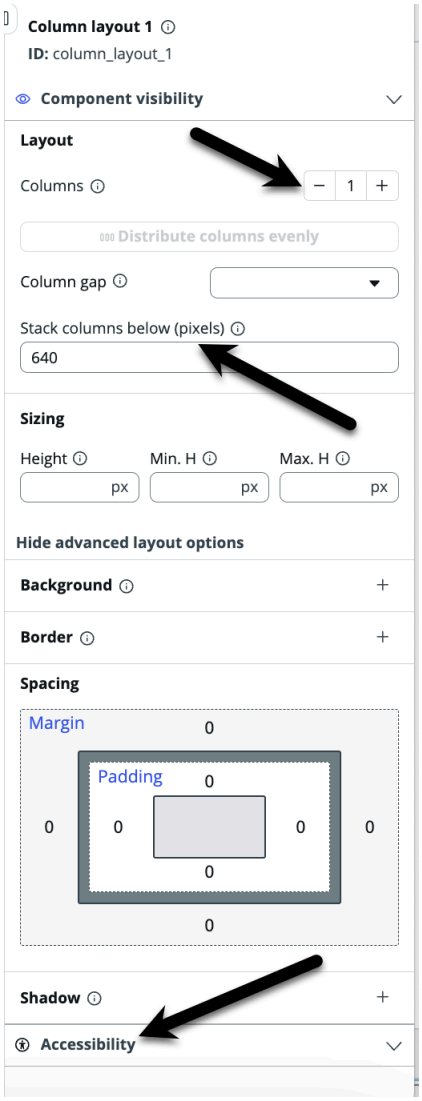
About this task

A method of increasing page usability is to tailor components by adjusting the layout across breakpoints. There are numerous options for adjusting the layout. In this procedure, adjust the layout of the tabs component for the mobile form factor.

Most column layout options on the property pane can be edited for different form factors. However, the following options have global values across form factors:

- Number of columns
- **Stack columns below (pixels)** option
- All options under **Accessibility**

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

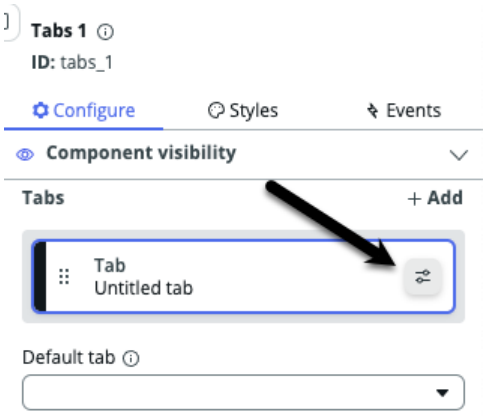


Procedure

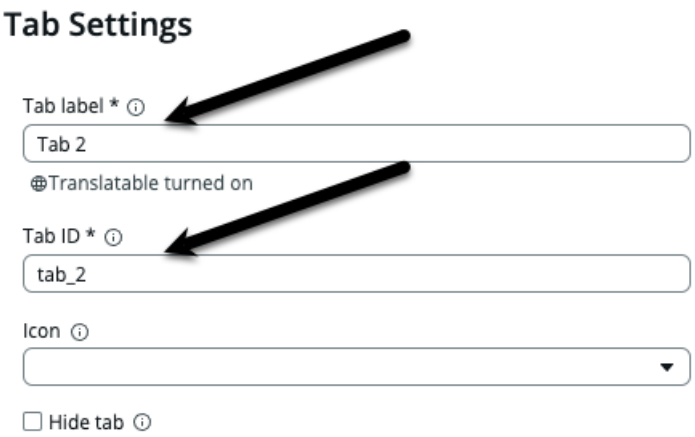
1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create > Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create a page from scratch with responsive authoring.
   For more information about how to create a page, see Create a page in UI Builder.
4. At the top, check that the desktop form factor icon is selected.

5. On the stage, select **+ Add content**.
6. On the **Layouts** tab, select **Single column**.
7. Add the tabs component.
   a. Select the **+** icon in the center of the column on the stage.
   b. On the **Components** tab, find and select the **Tabs** component.
   c. Select **Save**.
8. In the configuration panel, on the **Configure** tab, edit the name of the first tab by selecting the edit icon.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



9. In **Tab label**, enter `Tab 1`.
10. In **Tab ID**, enter `tab_1`
11. Select **Save**.
12. Add a second tab.
    a. On the **Configure** tab, select **+ Add** next to **Tabs**.
    b. Select **Start from an empty container** and then select **Next**.
    c. In **Tab label**, enter `Tab 2`.
    d. Check that **Tab ID** has been automatically filled in with **tab_2**.



    e. Select **Create**.
13. Add a third tab by repeating step 12 and in 12c name the tab `Tab 3`.
14. Check that your content tree and page are accurate.
15. Select the tablet form factor icon.

    The spacing and position of the tabs looks good at tablet width.
16. Select the mobile form factor icon.

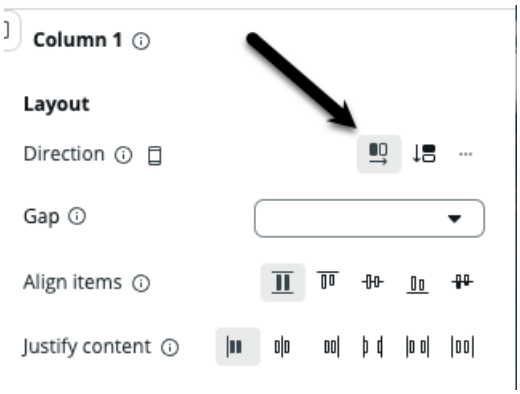
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The tabs take more space at the smaller size.

17. Change the layout of the tabs just for the mobile size.

    a. In the content tree, select **Column 1** under **Column layout 1**.

    b. On the configuration panel, in **Direction**, select the row icon.



    c. Select **Save**.

    The stage updates automatically showing that tab 1 is visible and that there's a new dropdown menu.

18. Select **Preview**.
19. Select **More**.
    The other two tabs are available.
20. Select the desktop and tablet form factor icons to see that the More drop-down doesn't appear.
21. Select the **X** to close the **Preview** overlay.

Create a breakpoint for responsive authoring

Learn how to create custom breakpoints for responsive authoring to control the look and feel of a page at different form factors.

Before you begin

Role required: ui_builder_admin

About this task

UI Builder currently offers three default form factors:

- Desktop (1281 pixels and larger)
- Tablet (1280 pixels and smaller)
- Mobile (500 pixels and smaller)

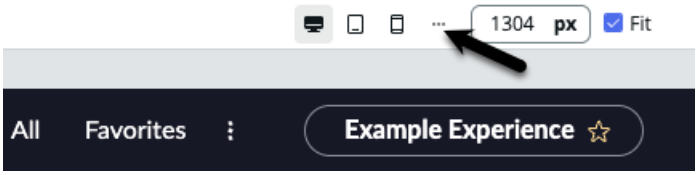
Add up to three additional breakpoints. For example, there's a default form factor of 500 pixels and smaller for mobile devices, but you may need a breakpoint of 360 pixels.

Note: Custom breakpoints represent a range. If you create a breakpoint at 800 pixels, all changes apply to devices with a width at and lower than the new size. For example, a page opened on a screen 750 pixels wide uses the 800 pixel settings (properties and styles) unless there's another custom breakpoint at 750 pixels.

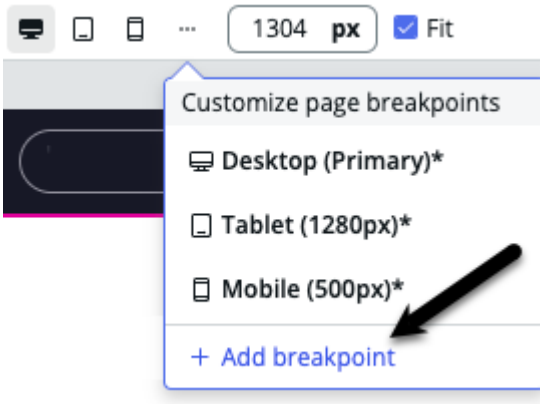
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

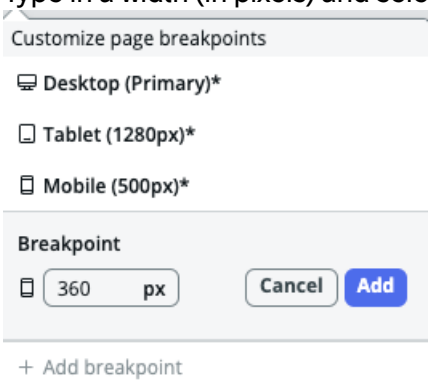
1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create > Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Open an existing page created from scratch with responsive authoring or create a page from scratch with responsive authoring.
   For more information about how to create a page, see Create a page in UI Builder.
4. Select the menu icon next to the three form factor icons.



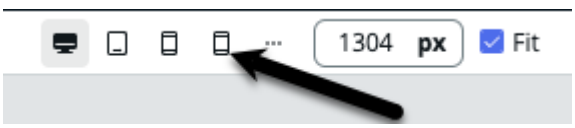
5. Select **+ Add breakpoint**.



6. Type in a width (in pixels) and select **Add**.



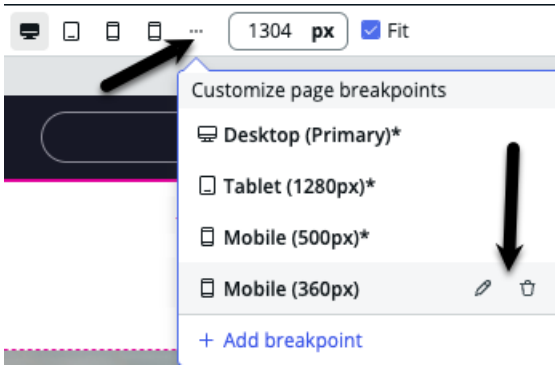
   The new breakpoint is saved and appears as a form factor icon.



   As you build the page, select any of the form factor icons to view and edit the page at that size.

7. To edit or delete a custom breakpoint, select the menu icon, point to the breakpoint name, and select the edit or delete icon.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Note: The default form factors (desktop, tablet, and mobile) can't be edited or removed.

Learn how to view and test your UI Builder experience

Preview your experience in UI Builder to see how it looks and functions while building an experience. Previewing helps ensure that your experience works as expected, that the data resources are available, and that the layouts are set up correctly.

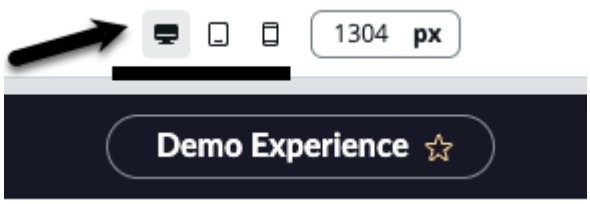
Test your page

You can test your page as the developer or see it as it would appear to an end user. Previewing doesn't require saving first.

Note: Previewing inactive variants and page collections (sub pages) isn't currently supported.

You can test various aspects of your page.

- Test the variant that you're currently working on, including modals, viewports, and dynamic data by selecting the **Preview** button, which opens an overlay.
- Test with different data by changing the test values and selecting **Apply**. For more information, see Test values in a page.
- See what the variant looks like at different screen widths by selecting a form factor icon. For example, view the variant at tablet or mobile size. For more information about creating pages for different form factors, see Responsive authoring.



Note: If you edit the test values or the form factor when previewing, the new test values or form factor are retained when you view the page variant in the editor. For example, if you changed the form factor to mobile while previewing, the page variant is displayed in the editor at the mobile form factor. You can always change to another size by selecting a form factor icon.

When you have finished testing, select the **X** in the upper right to close the overlay.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Test your page as an end user

Test which variant users see by opening the URL path. Select the arrow next to the **Preview** button and in the drop-down list select **Open URL path**. UI Builder checks the audience, conditions, and condition order, and then opens the appropriate variant in a new browser tab. When you have finished testing, close the tab.

Resolve a missing page definition in UI Builder

Fix UI Builder page render errors by linking a page definition in the variant record.

Before you begin
Role required: ui_builder_admin

About this task
Pages in UI Builder are backed by a screen record (`sys_ux_screen`) that references a page definition. The page definition is a macroponent record that UI Builder renders. If the reference is removed or was not set, for example, after a partial import, a manual database edit, or data corruption, UI Builder cannot render the page.

Procedure
1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open the experience containing the page displaying the error.
3. Select the page displaying the error.
4. Select **Open variant record**.
5. Right-click the page with the error.
6. Select **Copy sys_id**.
7. Locate the macroponent by navigating to **All** > **sys_ux_macroponent.list**.
8. Filter the list by the application scope associated with the affected experience.
9. Find the macroponent record that matches the page you want to restore.
10. Copy the macroponent's name from the list.
11. Return to the experience containing the page displaying the error.
12. Select the page displaying the error.
13. Select **Open variant record**.
14. Select the **UX Screens** tab.
15. Open the screen record by selecting the variant screen name.
16. Enter the macroponent from the previous step in the **Page Definition** field.
17. Select **Update**.

What to do next
Open the experience containing the page with the error to verify that the error has been resolved.

Organize components in UI Builder pages

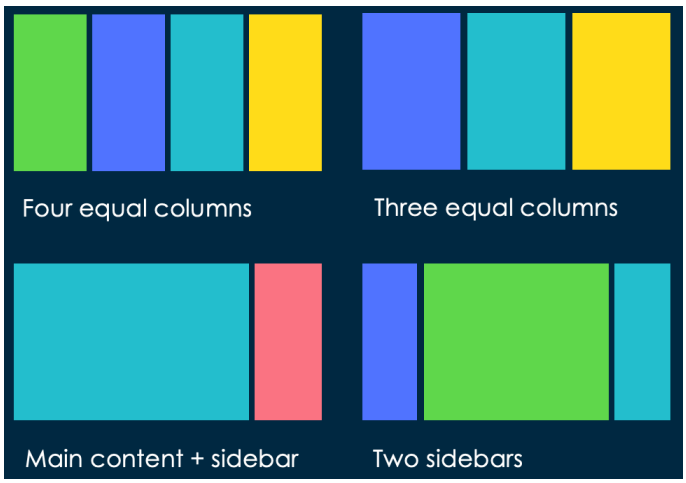
Column layouts organize components on a UI Builder page. Learn how column layouts are used in UI Builder.

Column layouts enable you to create useful and visually appealing pages in an experience. A column layout is a type of flexible container that can hold components. Column layouts can

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

have columns of equal or different widths. You can configure the column layout as a whole and the individual columns within the layout. For example, set a different border width and color for a column layout and for the individual columns within the layout. Column layouts can contain empty columns on a UI Builder page to create white space. Use multiple column layouts on a page. For example, a two-column layout could contain a header and image, then a three-column layout below it could hold more components like a list and form.

Think about the layout of your page before you begin building it. Analyze what you want to achieve and how complex the page layout must be. These questions determine what type of column layout or layouts you should use.



Column layouts in UI Builder use standards-based Cascading Style Sheets (CSS) rules and web layout technologies such as Flexbox, CSS grid, and absolute positioning. So whatever you can do in CSS you can do here. Visit Mozilla to learn more about CSS.

Note: If you have existing UI Builder pages created in an earlier release (Utah or Tokyo), you can upgrade to Vancouver or Washington DC and add column layouts to the pages.

Column layouts

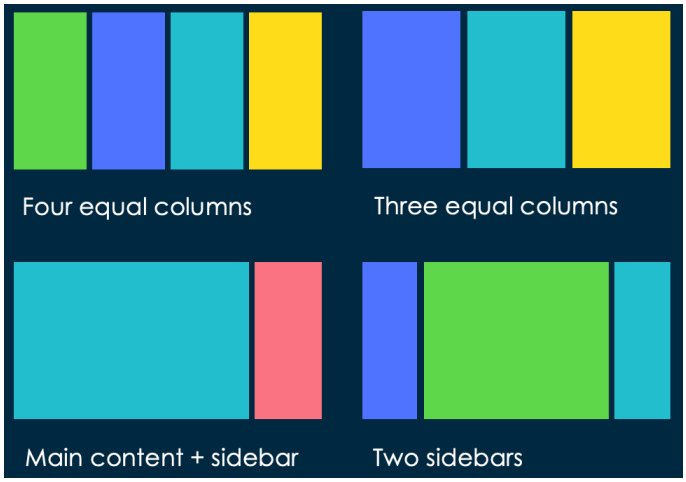
Column layouts are used to design and organize UI Builder pages.

This video shows you how to perform the following procedure.

Column layouts enable you to create useful and visually appealing pages in an experience. A column layout is a type of flexible container that can hold components. Column layouts are simple layouts that are preconfigured, enabling you to design an efficient and attractive page quickly. It's useful to define and set the structure of the page with column layouts before adding components.

Column layouts can have columns of equal or different widths.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



You can configure the column layout as a whole and the individual columns within the layout. For example, set a different border width and color for a column layout and for the individual columns within the layout.

Note: Currently, hiding/showing columns in a column layout based on conditions isn't supported.

Add a column layout

Add a column layout in UI Builder to build structure and organize components on an experience page.

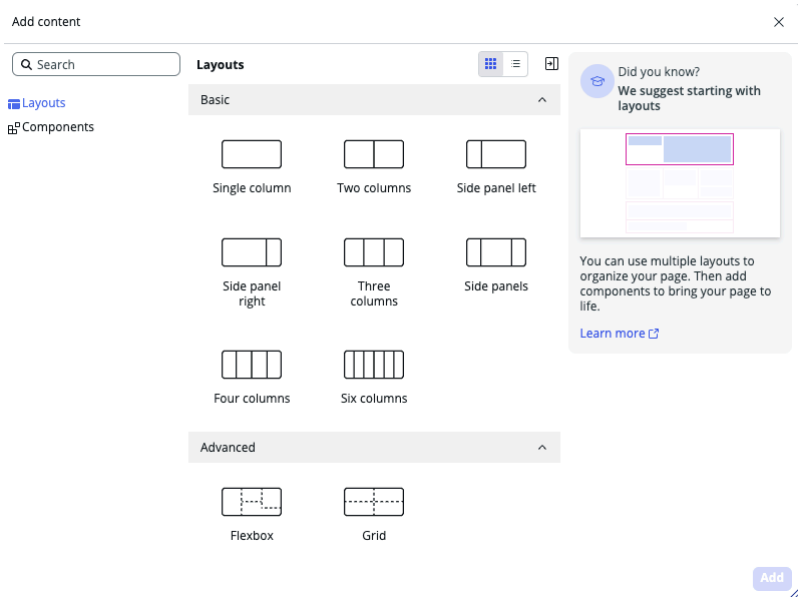
Before you begin

Role required: ui_builder_admin

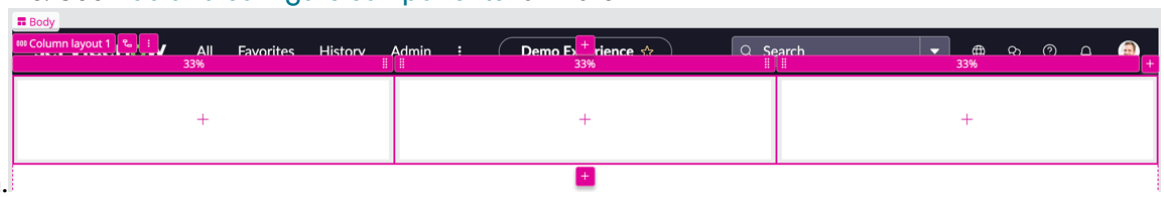
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create a page in UI Builder or open a page.
4. Select **+ Add content** in the content tree.
5. Select one of the **Basic** options on the **Layouts** tab.

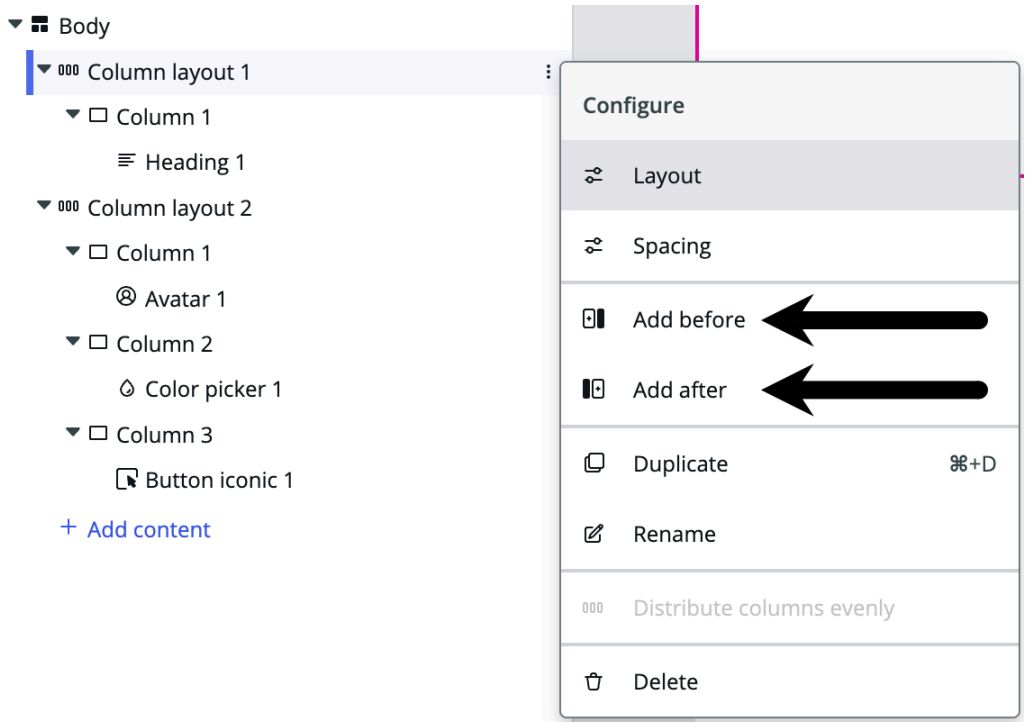
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



The column layout appears on the stage so you can add components to the columns. See Add and configure components for more information.

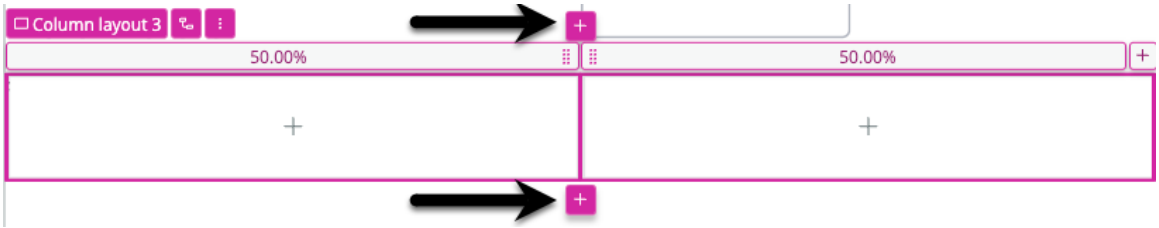


6. Add additional column layouts above or below.
   - In the content tree, select and hold (or right-click) on a column layout name or select the Menu icon for a column layout. Then, select **Add before** or **Add after** from the list and select a layout.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- On the stage, select a column layout, and then select the plus sign at the top or bottom.



Configure column layouts

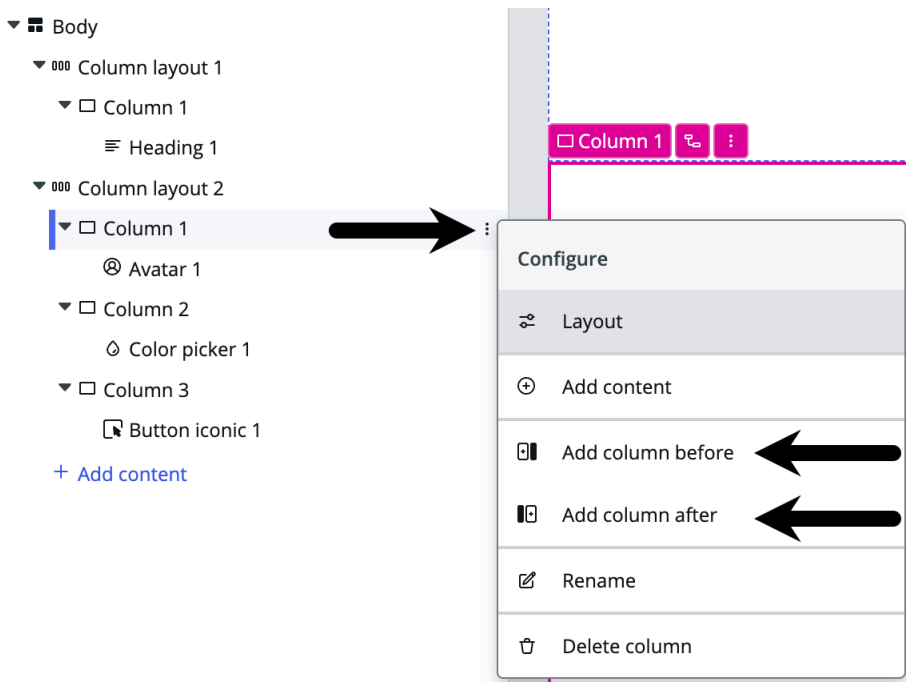
Configure column layouts in UI Builder to add the appropriate number of columns and the look you want on an experience page.

Before you begin

Role required: ui_builder_admin

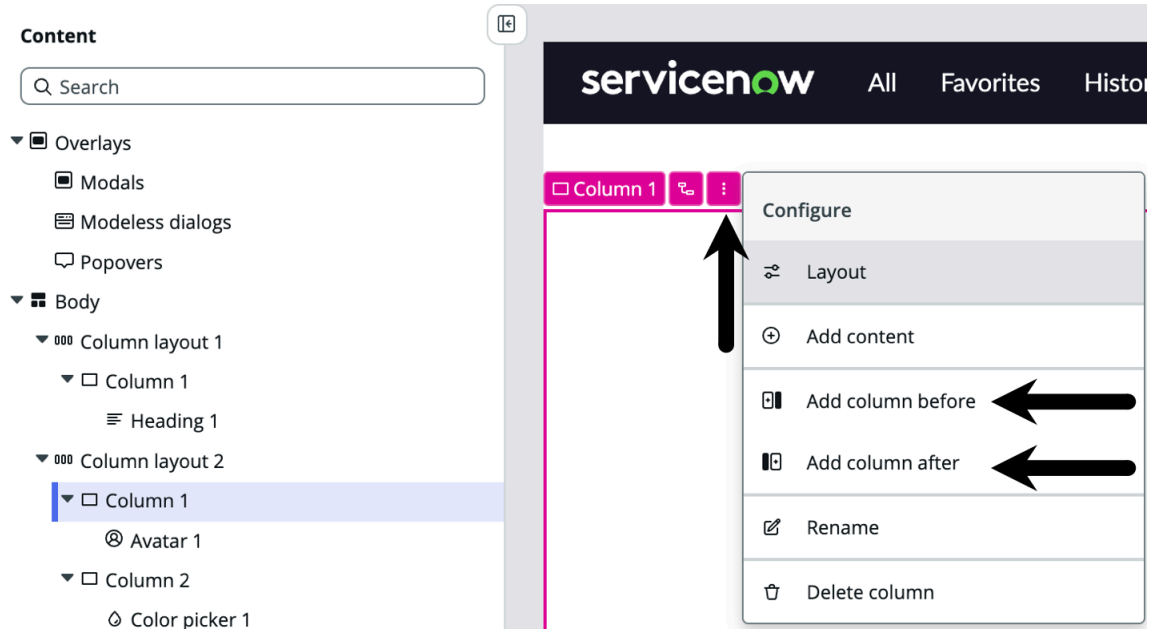
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create a page in UI Builder or open a page.
4. If the page doesn't already contain a column layout, Add a column layout.
5. Add (up to a maximum of six) columns using the content tree, the stage, or the configuration panel.
   - In the content tree, select and hold (or right-click) on a column name or select the Menu icon for a column and then select **Add column before** or **Add column after** from the list.

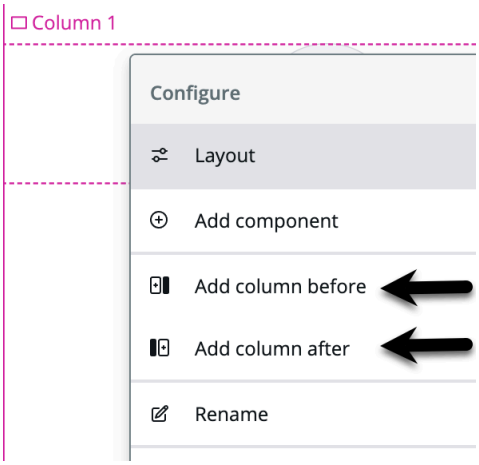


   - On the stage, select a single column, select the Menu icon next to the column name, and then select **Add column before** or **Add column after** from the list.

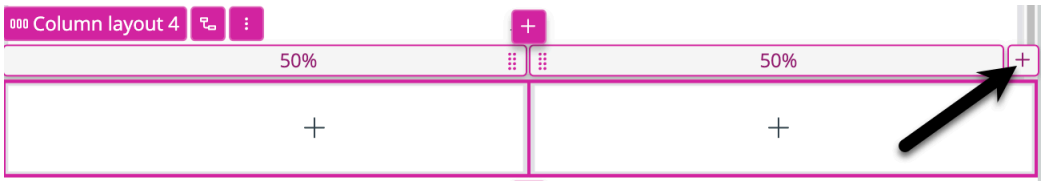
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



- On the stage, select and hold (or right-click) on a column and then select **Add column before** or **Add column after** from the list.

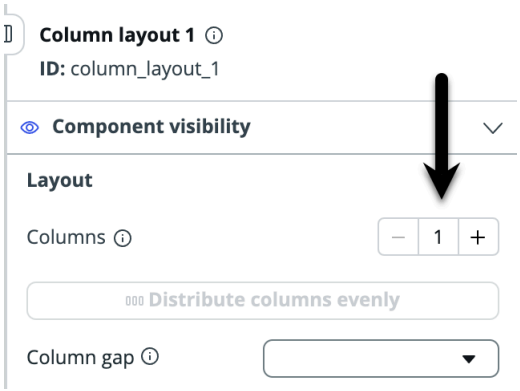


- In the content tree, select a column layout and then on the stage select the + plus icon to add columns to the right side of the column layout.



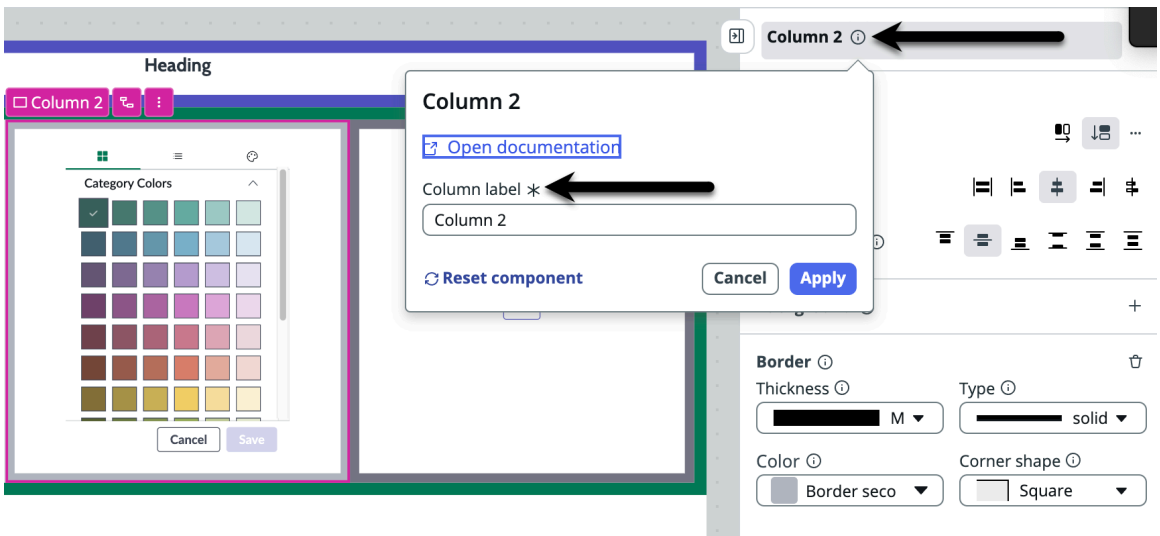
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- In the content tree, select a column layout and then in the configuration panel use the + plus icon next to **Columns** to add columns to the right side of the column layout.



Note: Columns can be nested inside one another, but this level of complexity isn't often required on pages.

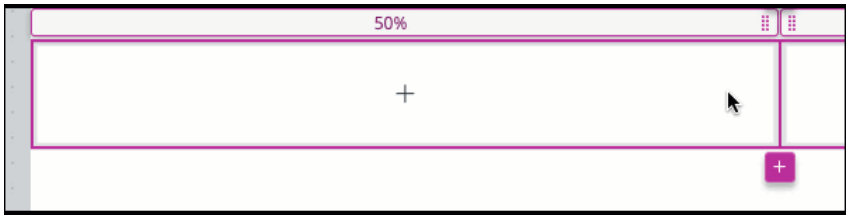
6. Rename a column by editing the column label from the content tree, the stage, or the configuration panel.
   - In the content tree, select and hold (or right-click) on a column name or select the Menu icon for a column, select **Rename**.
   - On the stage, select the Menu icon next to a column name, select **Rename**.
   - In the configuration panel, select the information icon next to the column name.



   Renaming columns can help you distinguish between different columns on the stage and in the content tree. If you have a complex page with many column layouts and columns, it is useful to rename columns.

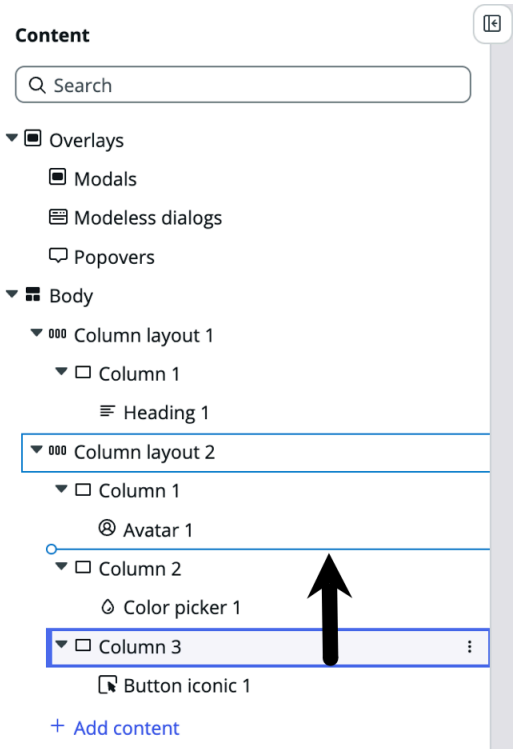
7. To change the width of a column in a layout, select the column layout (in the content tree or on the stage), then drag a column divider on the stage left or right to make the column narrower or wider.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



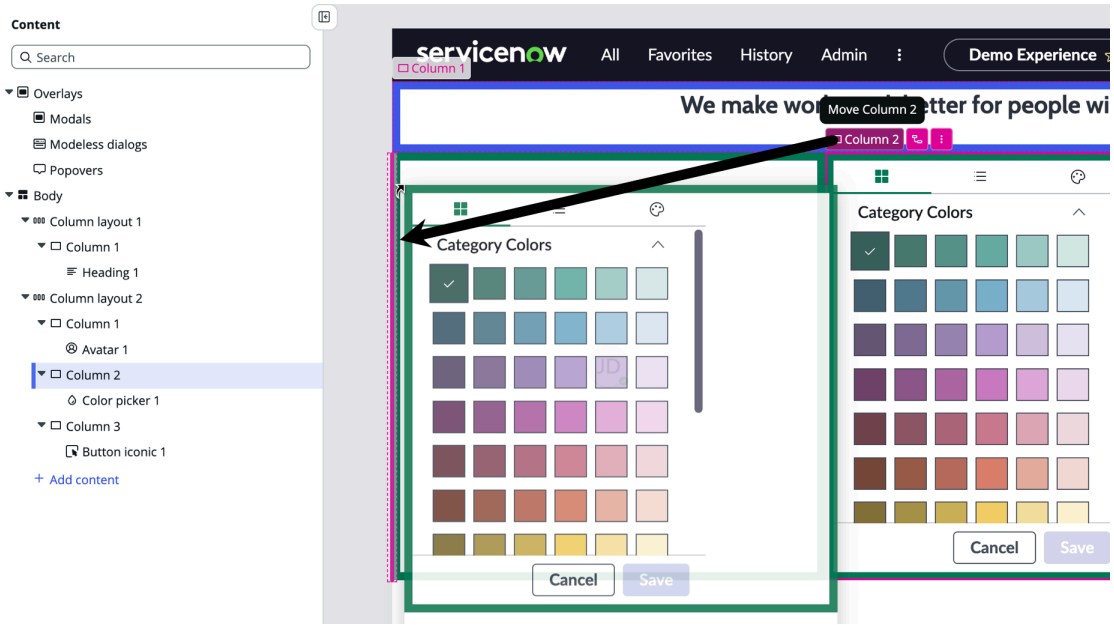
The width percentages at the top of the columns change automatically as you drag the column divider. The percentages add up to 100%. When resizing a column, be aware of how much space the component or components within the column need. For example, list and form components usually don't display well in narrow columns.

8. Distribute columns in a layout evenly across the horizontal using the content tree, the stage, or the configuration panel.
   - In the content tree, select and hold (or right-click) on a column layout name or select the Menu icon for a column layout, and then select **Distribute columns evenly** from the list.
   - On the stage, select a column layout, select the Menu icon next to the column layout name, and then select **Distribute columns evenly** from the list.
   - Select a column layout (in the content tree or on the stage) and in the configuration panel select **Distribute columns evenly**.
9. Reorder columns (including their contents) within a column layout using the content tree or the stage.
   - In the content tree, select the column and drag it to a different position (a blue horizontal line shows where the column can be dropped).



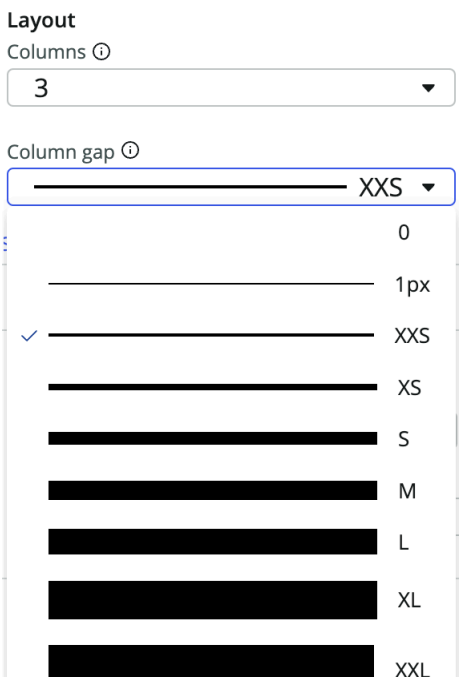
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- On the stage, select a column header, drag the column to a different position, and drop the column into the drop zone defined by a vertical magenta line.



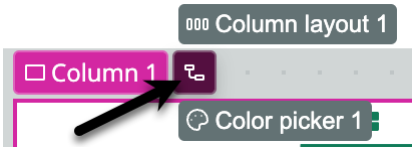
If you reorder the column in the content tree the stage updates, and if you reorder on the stage the content tree updates.

10. Specify the amount of space between columns from the content tree, the stage, or the configuration panel.
    - In the content tree, select and hold (or right-click) on a column layout name or select the Menu icon for a column layout, select **Layout** from the list, and select a size option in **Column gap**.
    - On the stage, select a column layout, select the Menu icon next to the column layout name, select **Layout** from the list, and select a size option in **Column gap**.
    - Select a column layout (in the content tree or on the stage) and in the configuration panel select a size option in **Column gap**.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

11. Select the Tree icon to navigate to a different layer of content, for example, from a column to the parent column layout or a child component.



12. Delete columns from the content tree or the stage.
    - In the content tree, select and hold (or right-click) on a column name or select the Menu icon for a column, and then select **Delete column** from the list.
    - On the stage, select a single column, select the Menu icon next to the column name, and then select **Delete column** from the list.
    - In the content tree, select a column layout and then in the configuration panel use the - minus icon next to **Columns** to remove columns from the right side of the column layout.

    For information about editing column spacing, see Set the gap between components in columns.

Set the gap between components in columns

In UI Builder, if a column contains multiple components, set the gap between the components.

Before you begin

Role required: ui_builder_admin

Procedure

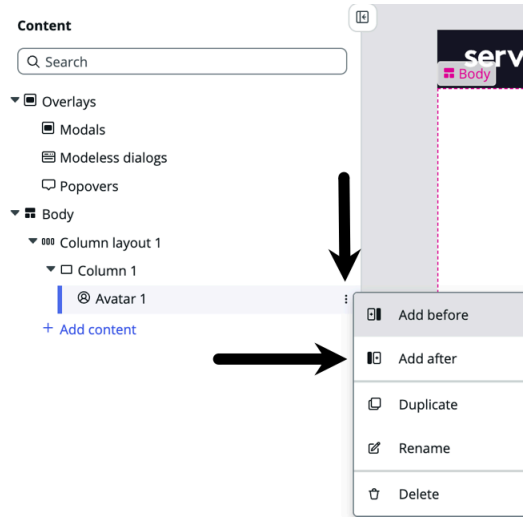
1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create a page in UI Builder or open a page.
4. If the page doesn't already contain a column layout, Add a column layout.
5. Within a single column, add two components

    a. In the content tree, identify a column and select **+ Add content** under the column name.

    b. In the toolbox window, select a component, for example, **Avatar**.

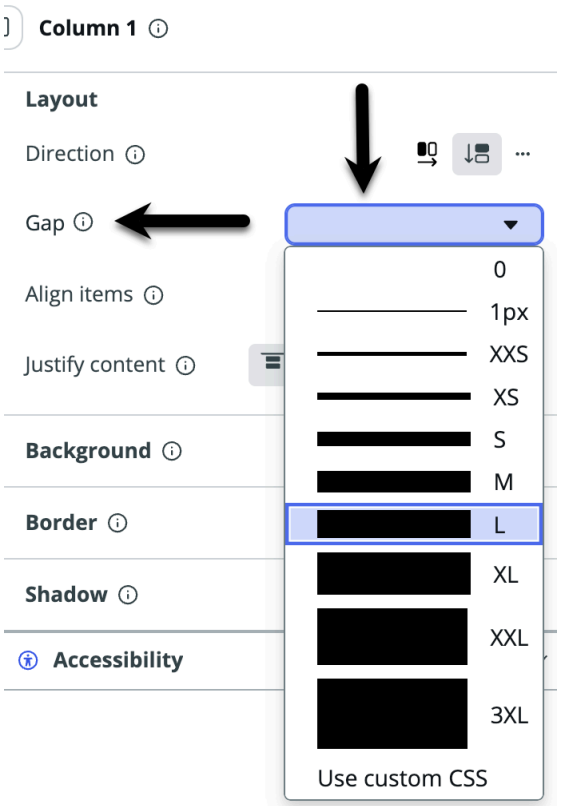
    c. In the content tree, move your mouse over the name of the component you just added, select the open menu (three vertical dots) icon, and select **Add after**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



d. In the toolbox window, select another component, for example, **Button**.

6. Select **Save**.
7. In the content tree, select the column to which you added the two components.
8. In the configuration panel, select a size option in **Gap**.



9. To specify an exact gap size in pixels, move your mouse over the **Gap** field, select the **Use custom value** (pencil) icon, and type in a number, for example, 300px.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



10. To change between stacking the components and placing them next to each other in the column, select the **Row** or **Column** icon in **Direction**.



To reverse the order and placement of the components in the column, select the **Advanced** (three horizontal dots) icon in **Direction**, and then select the **Reverse** option.



Set advanced column layout options

Set advanced column layout options in UI Builder including stack width and column layout height.

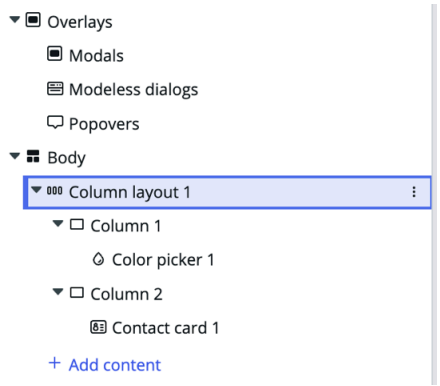
Before you begin

Role required: ui_builder_admin

Procedure

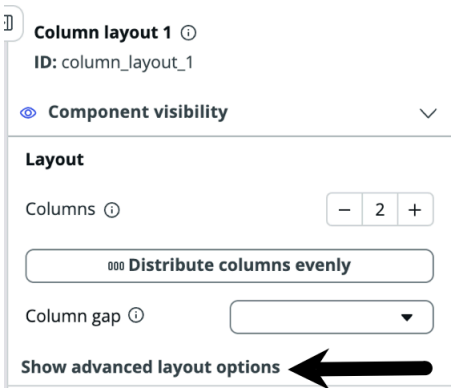
1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create a page in UI Builder or open a page.
4. If the page doesn't already contain a column layout, Add a column layout.
5. Select a column layout containing two or more columns in the content tree.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



6. To set a stack width, go to **Configuration panel > Layout** and select **Show advanced layout options**.

   Note: Setting a stack width enables you to control how a page with column layouts looks in a narrow browser window, tablet, or mobile device.

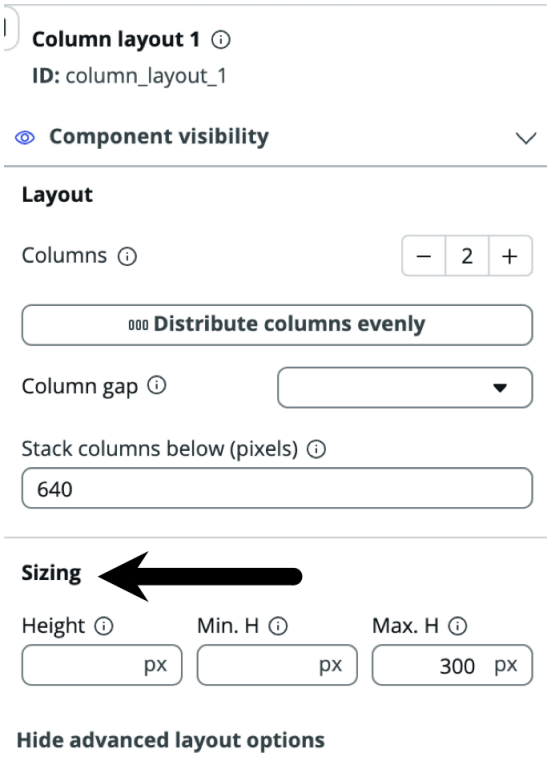


7. Type a width in **Stack columns below**.
8. Select **Save**.
9. To test the updated stack width, select the arrow next to **Preview** and select **Open URL path**.
10. Reduce the browser window width to less than the stack width that you specified to check that the columns on the right move below the first column.
11. Close the browser tab that opened with the URL path.
12. To control column layout height, select a column layout with two or more columns in the content tree, go to **Configuration panel > Layout**, and select **Show advanced layout options**.

    Note: Setting height is useful if your column layout contains a component that can be tall, like the List component.

13. Under **Sizing**, type numbers into one or more of the options (**Height**, **Min. H**, and **Max. H**), and set the sizing unit (for example, **px**, **%**, or **em**).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



14. Select **Save**.
15. To test the updated sizing, select the arrow next to **Preview** and select **Open URL path**.
16. Reduce and expand the browser window height to test the specified values.
17. Close the browser tab that opened with the URL path.

Upgrading layouts in UI Builder

Upgrade layouts created in Quebec and Rome to the new layout system.

The UI Builder layout system was enhanced and updated in the San Diego release. This new layout system features a simplified content tree and a new styling panel for layout configuration and component styling. All newly created page variants use the new layout system. You have the option to upgrade to the new layout system for existing and custom pages.

The new layout system includes the following:

- A simplified content tree
- Improved styling panel for layout configuration and component styling
- Low-code UI for configuring CSS Flexbox and Grid layouts.
- Enhanced UI for component styling
- Updated layouts do not require slots

Things to look for after upgrading layouts

Variants and pages are upgraded individually.

A page/variant is made up of a layout, components, and subcomponents, that are wrapped in a container within a layout with data resources, event bindings, and composition. Only the layout of your page is upgraded when upgrading to the new layout system, so there is no impact to your data, components, or component styling.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

All slots in the old layout system are migrated during the upgrade process to the new layout. The upgrade process changes how the component containers are structured on your page.

While component styling is not impacted, there may not be a one-to-one style migration of how components were positioned. Complex pages can have visual misalignments that occur through the upgrade process as styles are merged from slots to containers. Some issues must be resolved manually.

Upgrade your layout to the new layout system

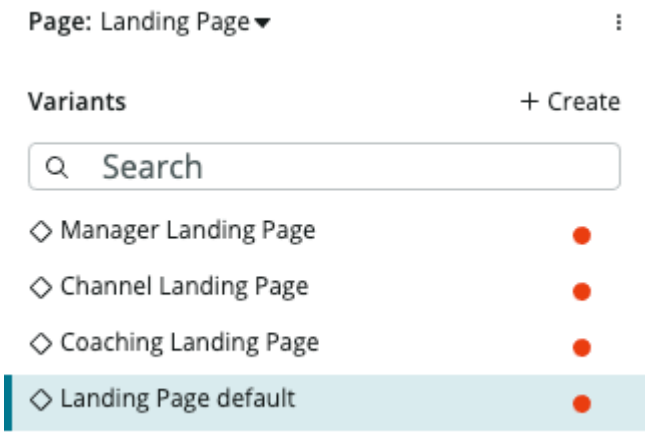
Upgrade your UI Builder pages to the new layout system.

Before you begin

Role required: ui_builder_admin

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience with the page you want to upgrade.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Select the page that you want to upgrade.

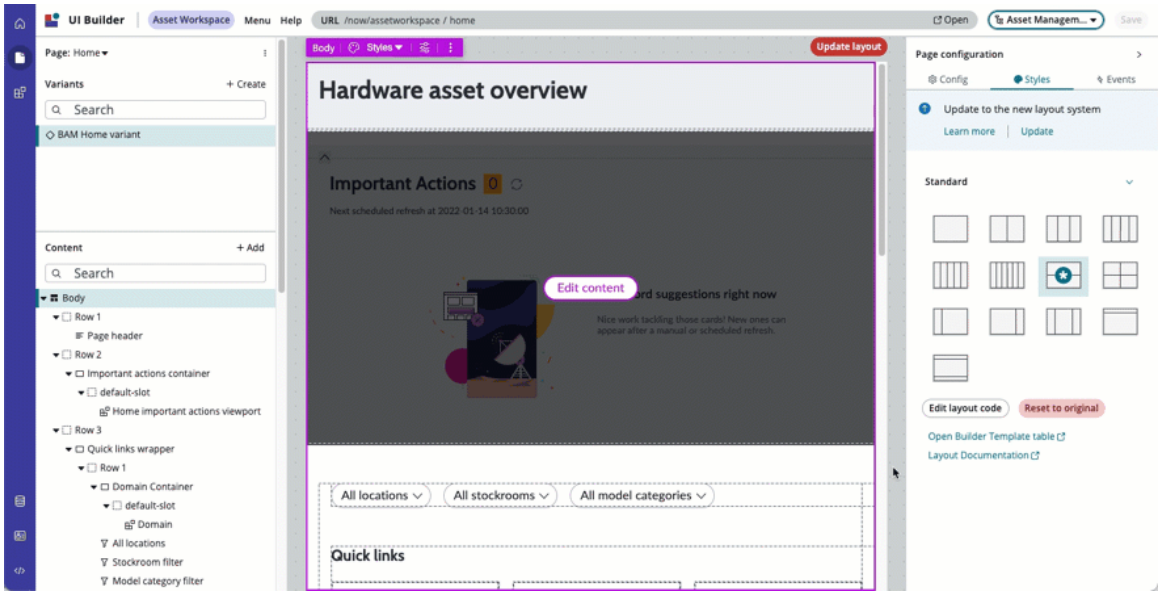


   You can find page variants that can be upgraded to the new layout system tagged with a red dot.

   Note: Legacy pages created using the pre-Quebec row/column layout system are not upgradable and must be rebuilt with the new layout system.

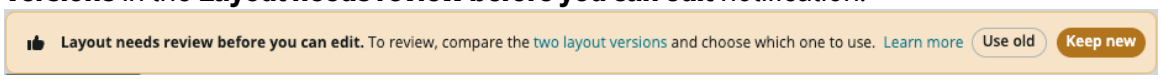
4. Select the **Update layout** button.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



UI Builder begins updating your page layout to the new layout system. A notification appears when the process is complete.

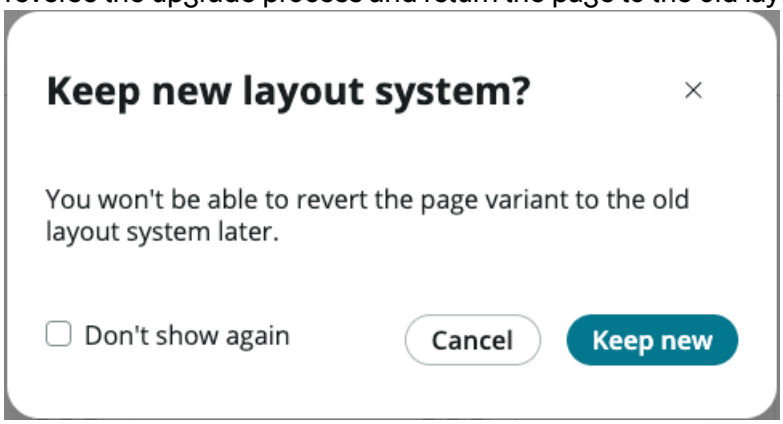
5. You can compare your old layout to the upgraded layout by selecting **compare the two layout versions** in the **Layout needs review before you can edit** notification.



Two browser tabs open, one displaying your current layout and one displaying the upgraded layout.

6. Select **Keep new** if you would like to update the page to the new layout system, or select **Use old** if you would like to keep the page layout as it was.
A modal appears asking you to confirm your selection.

7. Select **Keep new** to complete the upgrade to the new layout system, or select **Use old** to reverse the upgrade process and return the page to the old layout system.



8. The page reloads with the changes applied.

9. View and test your page by selecting the **Preview** button (

).

Using Flexbox layouts in UI Builder

Create a Flexbox layout in UI Builder to build powerful pages so that you can customize with cascading style sheets (CSS) and can improve your performance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

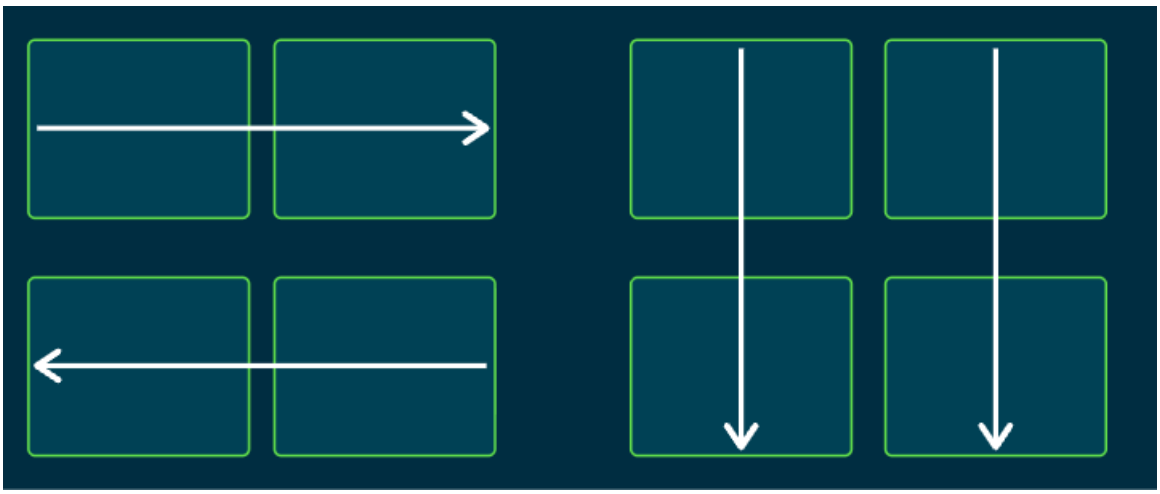
You can easily build custom pages with Flexbox layouts in UI Builder. Customizing the layout of your page lets you take full advantage of Flexbox so that you can achieve your overall page design. For more information, see Organize components in UI Builder pages.

Flexbox is a one-dimensional layout system in CSS. Flexbox is inherently flexible, which is useful for when you don't know the size of your content. If you plan to build complex pages, you can customize the layout of your page in the configuration panel styles tab or with CSS. Visit Mozilla to learn more about Flexbox.

You can change the direction of your content in the layout in the following ways:

- Row: Rows are organized left-to-right or right-to-left, depending on the direction of your browser's default language. Left/right is the case for an English browser.
- Row-reverse: Rows are organized in the reverse direction of your browser's default language, such as right-to-left or left-to-right.
- Column: Up/down or down/up
- Column-reverse: Down/up

Flexbox row and column directions



- Justify content: Defines the alignment along the main axis. Choices are as follows:
  - Flex-start: Items are at the start of the flex direction, similar to left-justified content. This is the default setting.
  - Flex-end: Items are at the end of the flex direction, similar to right-justified content.
  - Space-between: Items are distributed evenly.
  - Space-around: Items are distributed evenly with equal space around them.
  - Space-evenly: The spacing between items is equal.
- Align items: Defines how you want your flex content displayed along the cross axis. Choices are as follows:
  - Stretch: Stretch your content to fill the container. This is the default setting
  - Flex-start: Place your content at the start of the cross axis.
  - Flex-end: Place your content at the end of the cross axis.
  - Center: Center your content in the cross axis.
  - Base system: Align your content the same as your baseline alignment.
- Height: Set the height automatically or manually.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Width: Set the height of your flexbox items automatically or manually.
- Margin: Set your minimal distance between flexbox items.
- Padding: Set the padding for each side of your flexbox items.

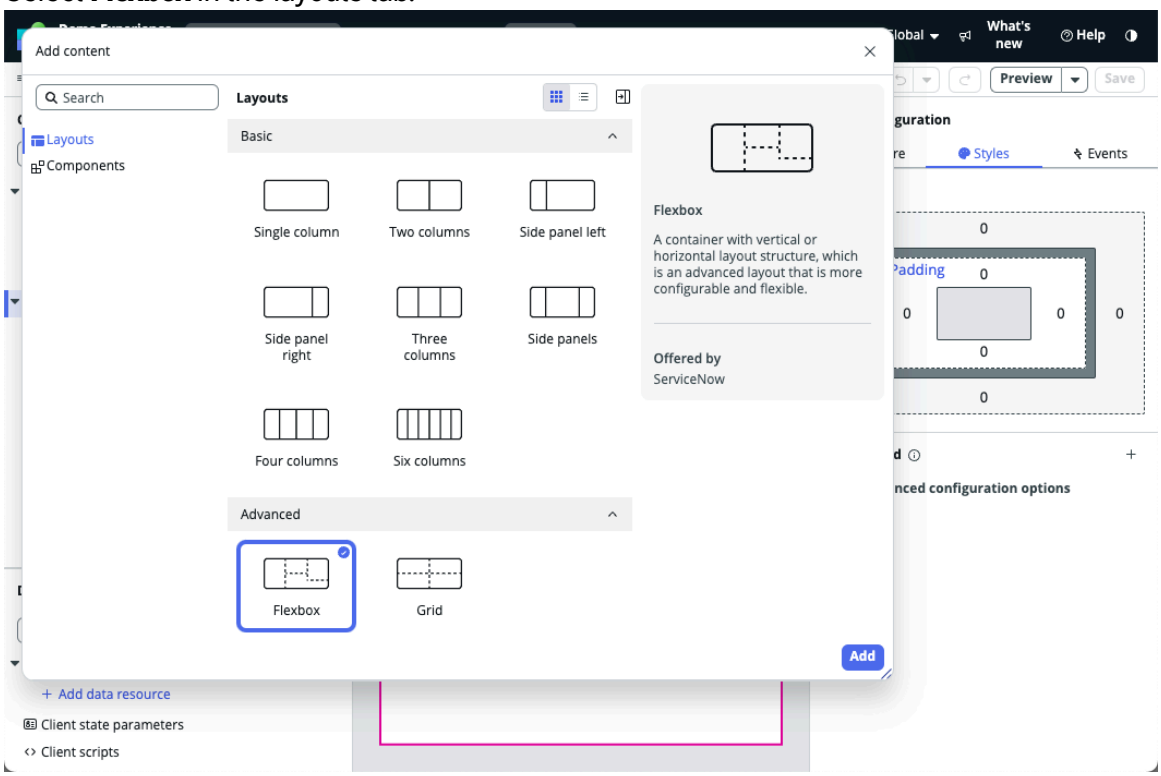
Create a Flexbox layout with the new layout system

Create a Flexbox layout in UI Builder to build powerful pages in a low-code environment.

Before you begin
Role required: admin

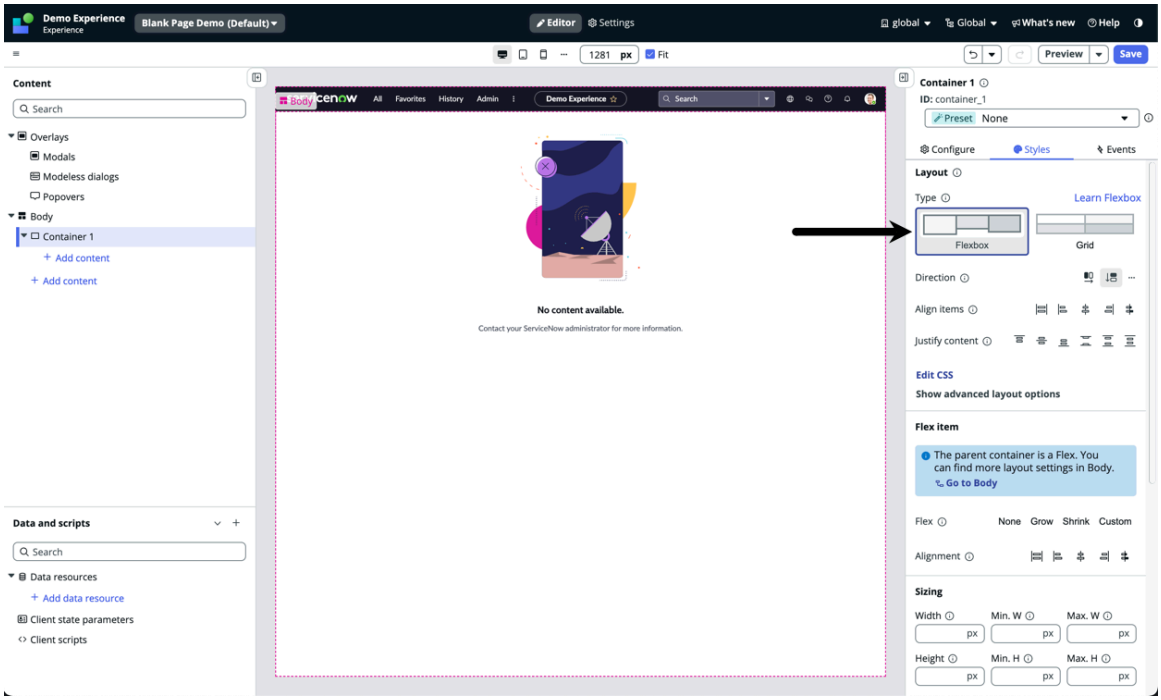
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create a page in UI Builder.
4. Select the **+ Add content** button in the UI Builder stage.
5. Select **Flexbox** in the layouts tab.



6. Select **Add**.
7. Select the **Styles** tab in the configuration panel.
8. In the **Layout** section, you should see **Flexbox** highlighted.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



9. In the **Layout** section of the styles tab, you can change the following:

   Direction

   Define how you want components to stack, horizontally or vertically.

   Align Items

   Define how components within containers align as a group.

   Justify content

   Define how to pack or space apart the components within the container.

   Note: Select **Show advanced layout options** to see the next two options.

   Gap

   Set the size of the gap between rows and columns.

   Wrap child elements to multiple lines

   Set whether components are forced onto one line or overflow onto multiple lines.

   Visit Mozilla to learn more about Flexbox layout configurations.

10. **Optional:** You can edit the CSS code by selecting **Show advanced configuration options** at the bottom of the **Styles** tab, and then selecting the **View and edit CSS** option.
11. Select **Save**.
    Your page refreshes and displays the changes made to the layout.
12. Add components to your page.
    See Add and configure components for more information.
13. View and test your page by selecting Preview.

Create a Flexbox layout with the old layout system

Create a Flexbox layout in UI Builder to build powerful pages so that you can customize with cascading style sheets (CSS) and can improve your performance.

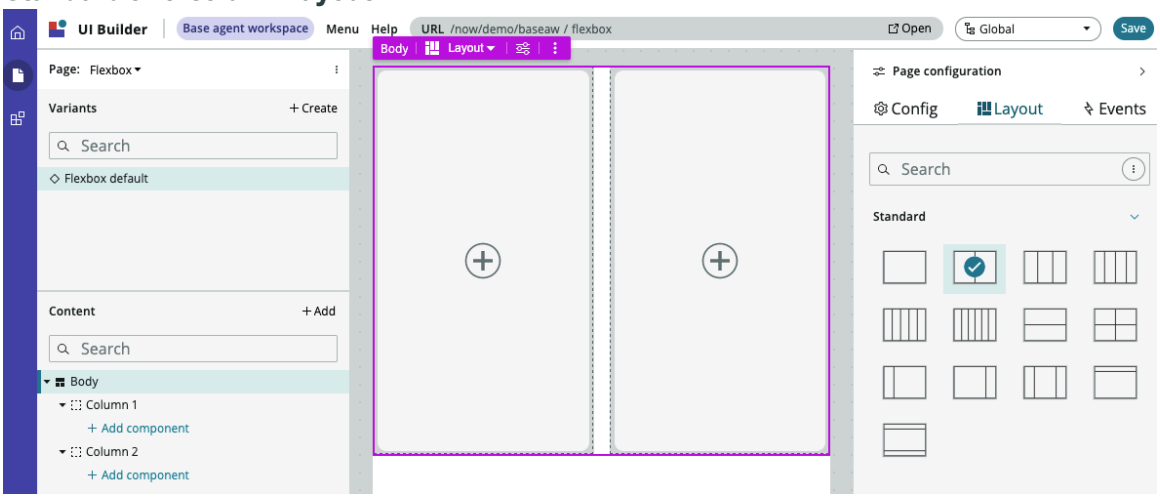
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin
Role required: admin

About this task
In the following procedure, you learn how to use Flexbox to change the CSS code to customize the layout of your page.

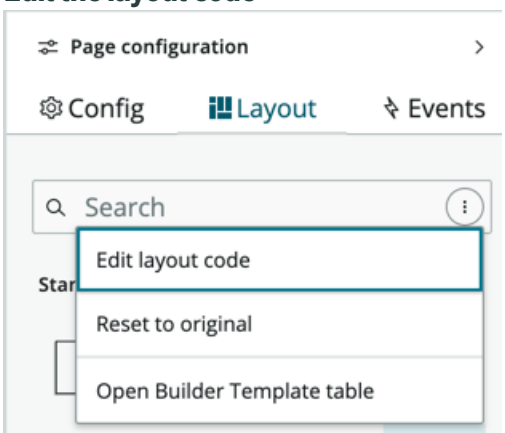
Procedure
1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create a page in UI Builder or open a page.
4. Click the **Layout** tab and choose the two-column layout.

Standard two-column-layout



5. Click **Edit layout code**.

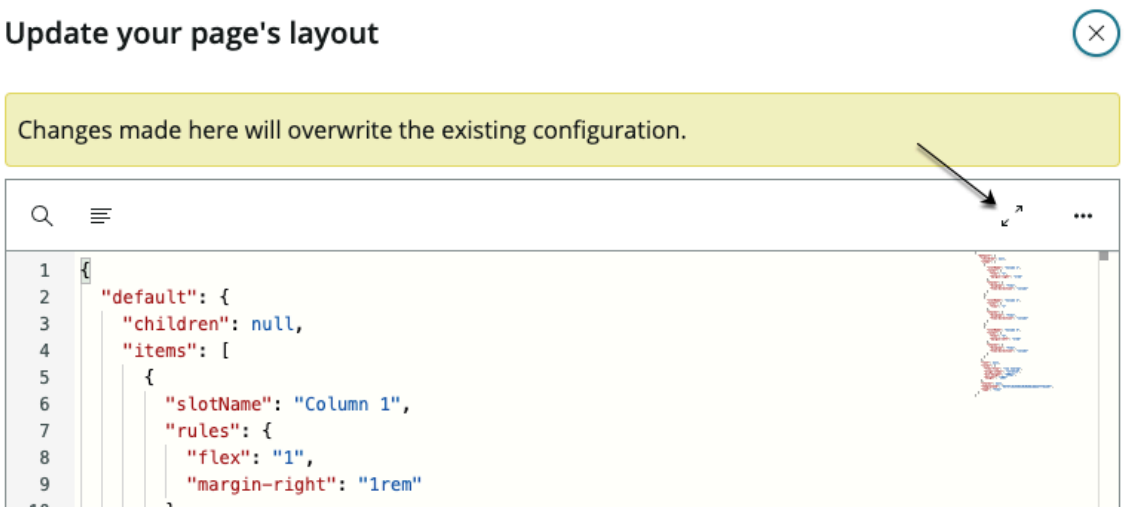
Edit the layout code



6. Expand the code editor so that you can easily view the CSS code.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Expand the code editor



7. Add another slot to the layout by copying the CSS code for an existing slot.

   Copying the code is easier than typing it into the layout.

Copy the CSS code for a slot



8. Below the code that you copied, place a comma and then paste the following CSS code:
   a. Change the *"slotName"* property to `"Column 3",`.
   b. Change the *"flex"* property to `"2"`.
   c. Add `"margin-left": "1rem"`.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

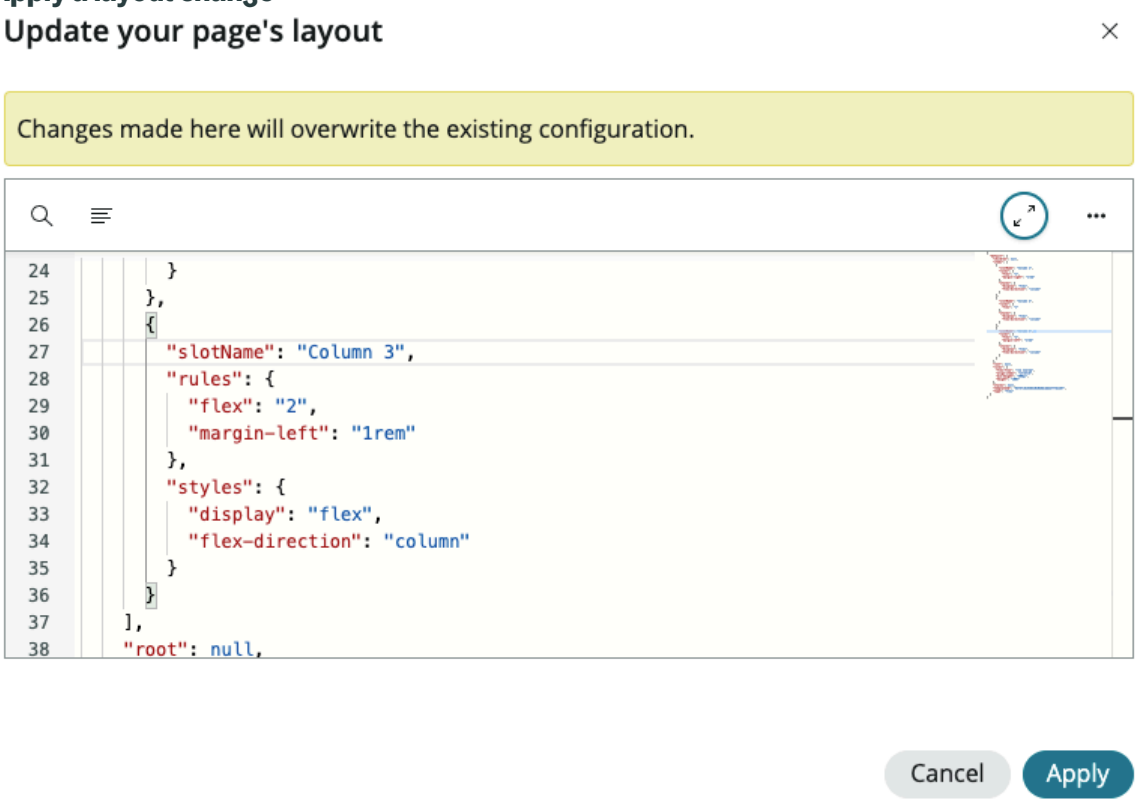
Paste the CSS code

```
        "flex": "1"
      },
      "styles": {
        "display": "flex",
        "flex-direction": "column"
      }
    },
    {
      "slotName": "Column 3",
      "rules": {
        "flex": "2",
        "margin-left": "1rem"
      },
      "styles": {
        "display": "flex",
        "flex-direction": "column"
      }
    }
  ],
  "root": null,
  "rules": {
    "flex-flow": "row nowrap",
    "align-items": "stretch",
    "min-height": "400px",
    "height": "100%"
```

Changing the *"flex"* property increases the size of the column. By using *"margin-left"*, you add space between the previous columns.

9. Collapse the expanded view, and then click **Apply**.

Apply a layout change

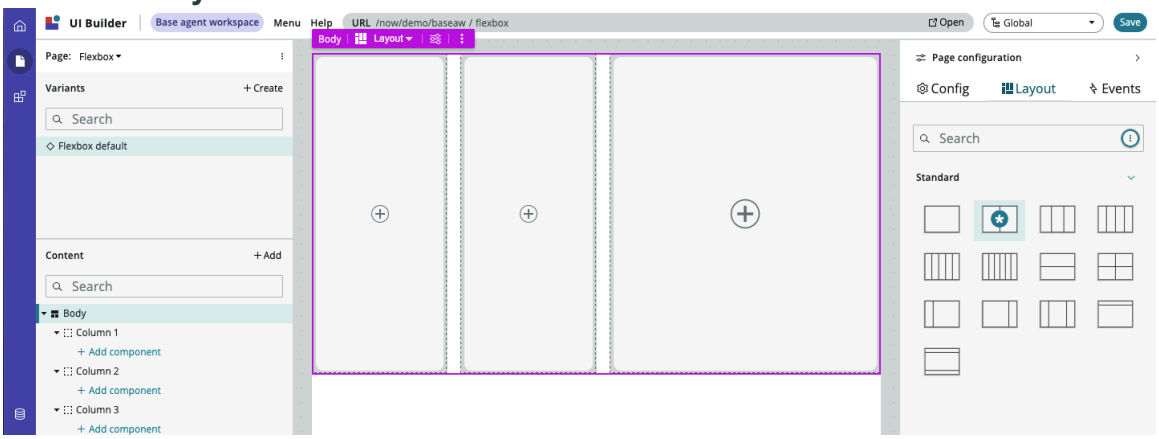


© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Result

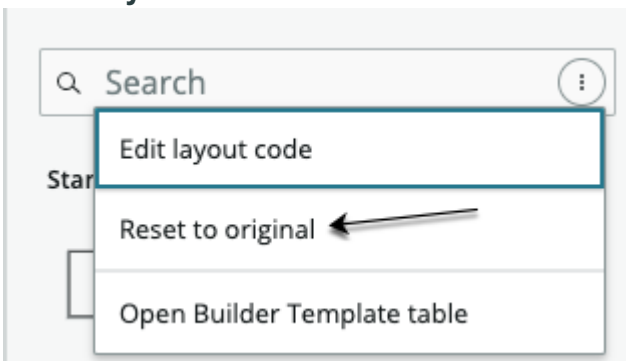
The new Flexbox layout that you created shows the new slot that you added to the original two-column layout.

New custom layout



Click **Reset to original** to reset layout changes back to the original.

Reset a layout



Related topics

Create a CSS Grid layout with the old layout system

Using CSS Grid layouts to build a page

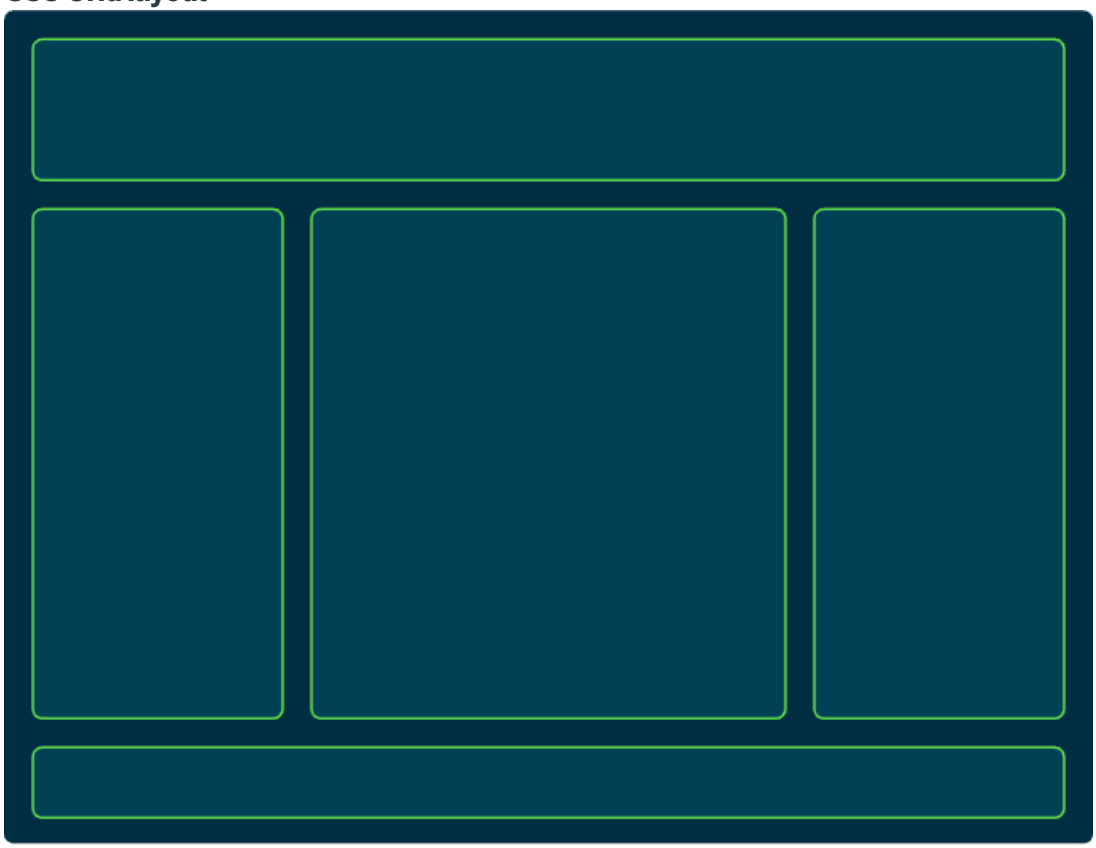
Create a CSS Grid layout in UI Builder to build powerful pages so that you can customize with cascading style sheets (CSS) and can improve your performance.

CSS Grid is the most powerful layout system in CSS. CSS Grid is built on top of a two-dimensional grid. CSS Grid gives you control over how you create your pages. Use UI Builder to implement a CSS Grid layout using low-code layout configuration properties. For advanced layouts, you can view and edit CSS code to customize your layouts. Visit Mozilla to learn more about CSS Grid.

For example, you can have a grid with three columns and three rows to make a grid nine cells. You can place components inside these cells or make the component span multiple cells.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

CSS Grid layout



Customizing the layout of your page lets you take full advantage of CSS Grid so that you can achieve your overall page design. For more information, see Organize components in UI Builder pages.

To find out more about CSS layouts within your UI Builder instance, you can find them in the [sys_uib_template] table.

Create a CSS Grid layout with the new layout system

Create a CSS Grid layout in UI Builder to build powerful pages in a low-code environment.

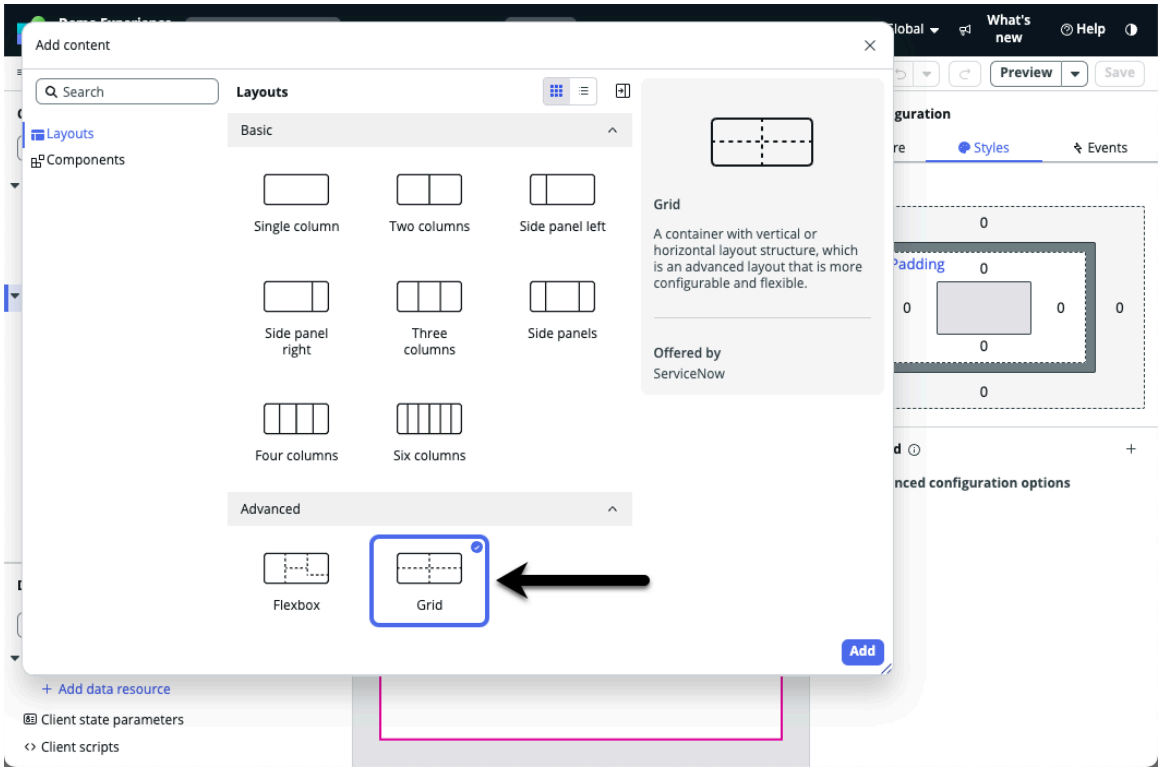
Before you begin

Role required: admin

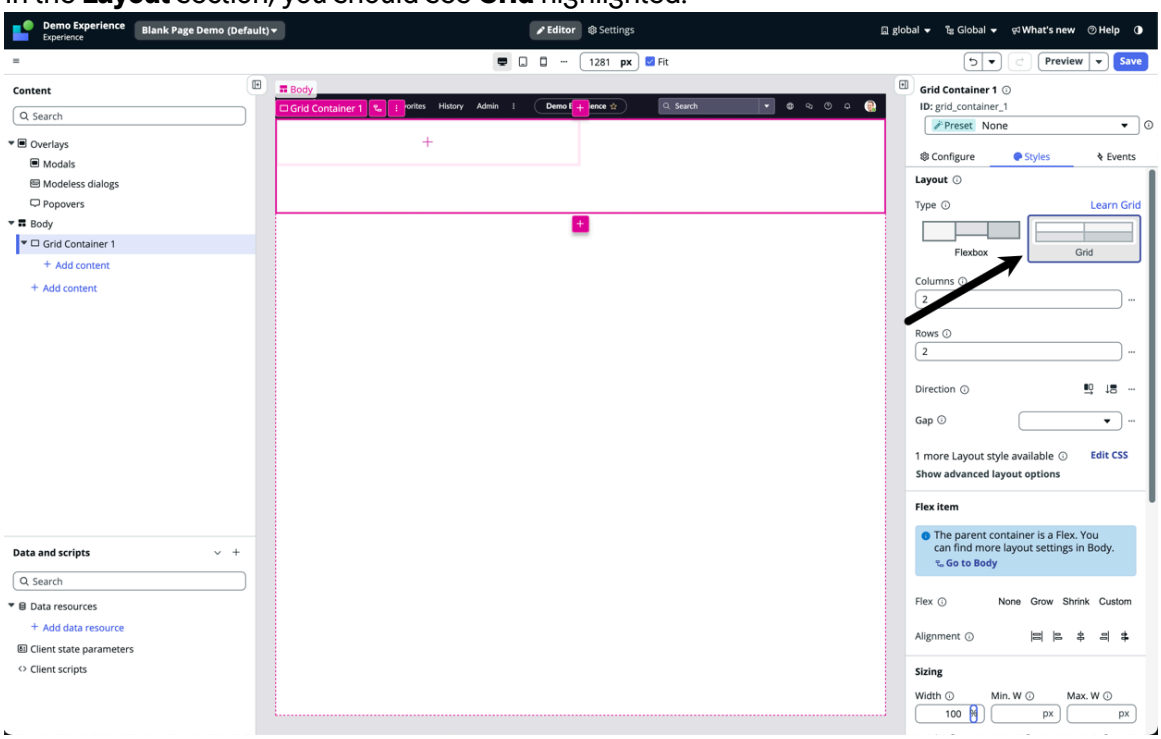
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create a page in UI Builder.
4. Select the **+ Add content** button in the UI Builder stage.
5. Select **Grid** in the layouts tab.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



6. Select the **Styles** tab in the configuration panel.
7. In the **Layout** section, you should see **Grid** highlighted.



8. In the **Layout** section of the styles tab, you can change the following:

Columns

Defines the number, order, and width of grid columns.

Rows

Defines the number, order, and width of grid rows.

Direction

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Sets whether newly added components are stacked horizontally or vertically.

Gap

Sets the size of the gap between rows and columns.

Note: Select **Show advanced layout options** to see the next four options.

Align items

Defines how components within the container align as a group.

Justify items

Defines how the browser distributes space between and around content items along the inline axis of a grid container.

Align content

Sets the distribution of space between and around content items along a grid's block axis.

Justify content

Defines how to pack or space apart the components within the container.

Visit Mozilla to learn more about CSS Grid layout configurations.

9. **Optional:** You can edit the CSS code by selecting **Show advanced configuration options** at the bottom of the **Styles** tab, and then selecting the **View and edit CSS** option.
10. Select **Save**.
    Your page refreshes and displays the changes made to the layout.
11. Add components to your page.
    See Add and configure components for more information.
12. View and test your page by selecting Preview.

Create a CSS Grid layout with the old layout system

Create a CSS Grid layout in UI Builder to build powerful pages so that you can customize with cascading style sheets (CSS) and can improve your performance.

Before you begin

Role required: admin

About this task

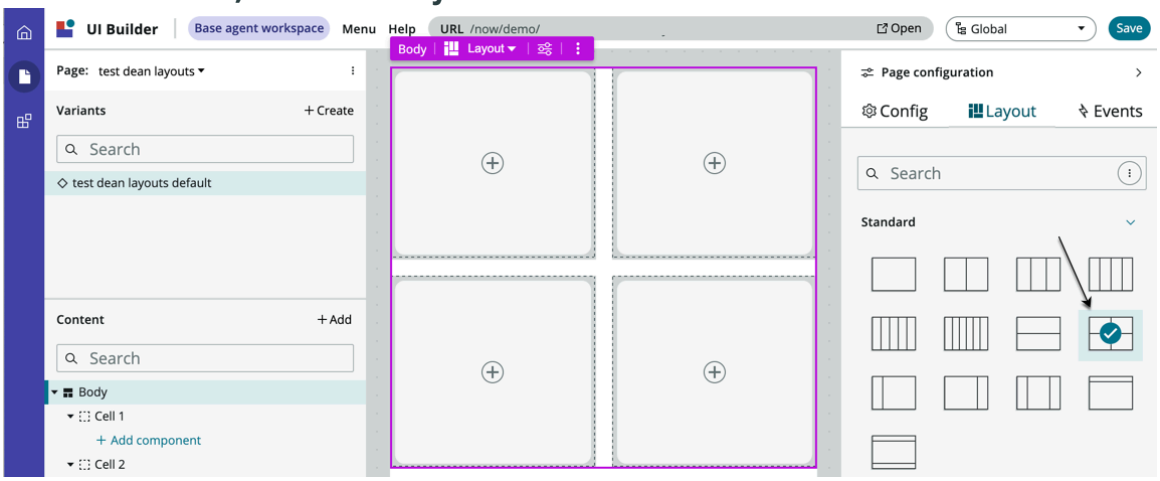
In the following procedure, you learn how to use CSS Grid to modify your CSS to customize the layout of your page.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create a page in UI Builder or open a page.
4. Click the **Layout** tab and choose the two-row by two-column layout.

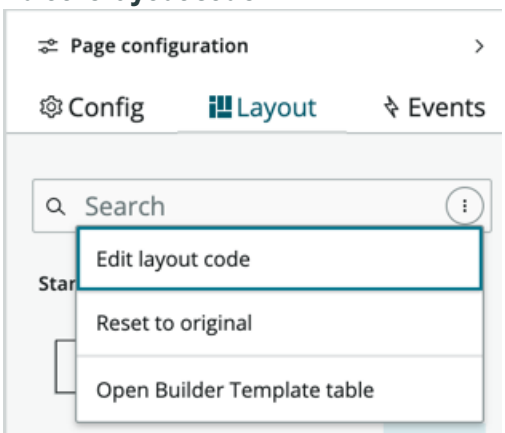
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Standard two-row, two-column layout



5. Click **Edit layout code**.

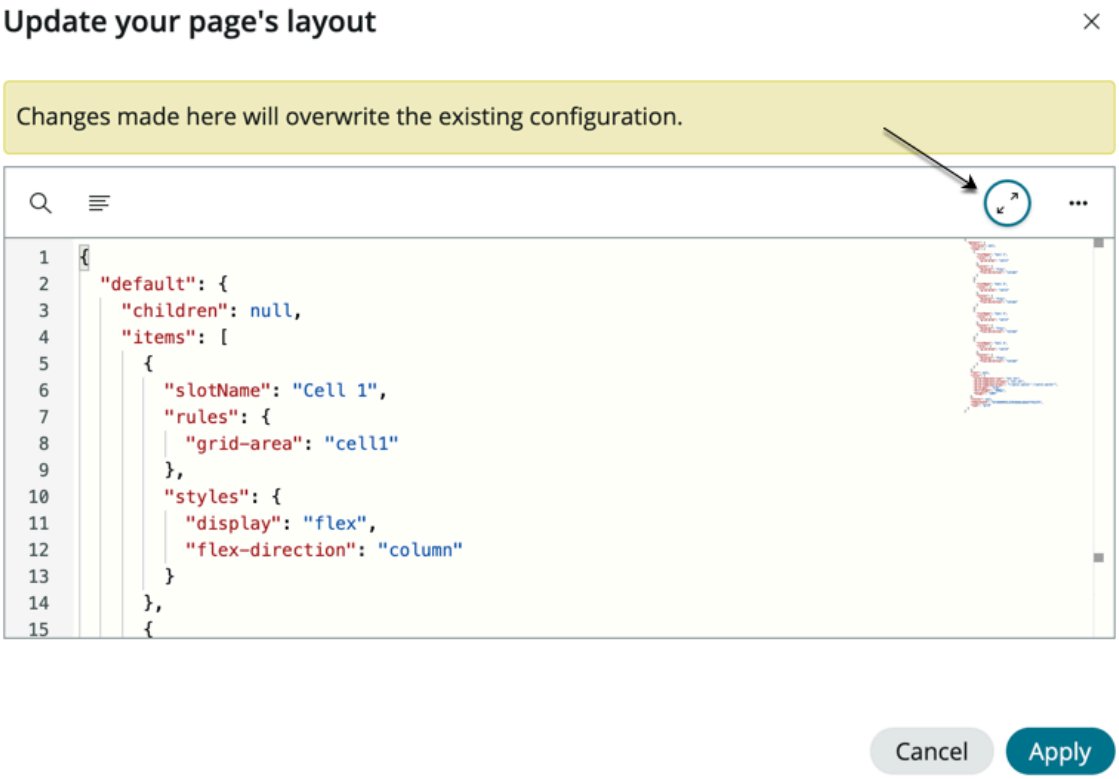
Edit the layout code



6. Expand the code editor so that you can easily view the CSS code.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Expand the code editor



7. Add another slot to the layout by copying the CSS code for an existing slot.

   Copying the CSS code is easier than typing it into the layout.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Copy the CSS code for a slot



```
23          }
24        },
25        {
26          "slotName": "Cell 3",
27          "rules": {
28            "grid-area": "cell3"
29          },
30          "styles": {
31            "display": "flex",
32            "flex-direction": "column"
33          }
34        },
35        {
36          "slotName": "Cell 4",
37          "rules": {
38            "grid-area": "cell4"
39          },
40          "styles": {
41            "display": "flex",
42            "flex-direction": "column"
43          }
44        }
45      ],
46      "root": null,
47      "rules": {
48        "grid-template-rows": "1fr 1fr",
49        "grid-template-columns": "1fr 1fr",
50        "grid-template-areas": "\"cell1 cell2\" \"cell3 cell4\"",
51        "grid-gap": "1rem",
52        "min-height": "400px",
53        "height": "100%"
54      },
55      "styles": null,
56      "templateId": "b719698953c31010e6bcddeeff7b1275",
57      "type": "grid"
58    }
59  }
```

8. Paste the CSS code below the code that you copied and change the *`slotName`* and *`grid-area`* to a unique value.

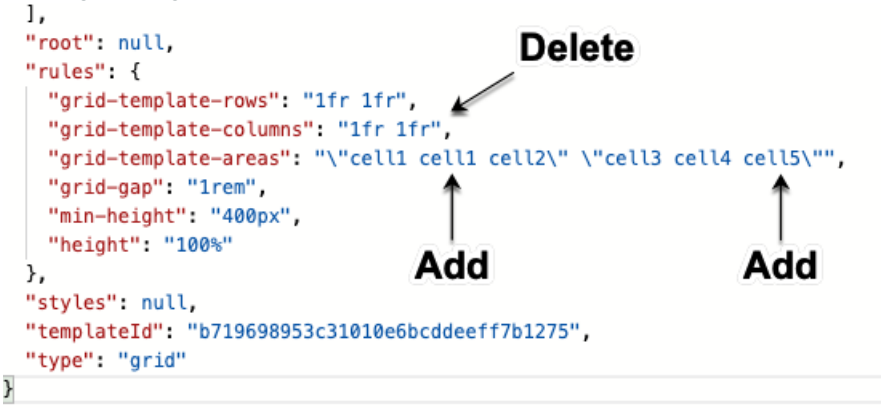
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Paste the CSS code

```
{
  "slotName": "Cell 4",
  "rules": {
    "grid-area": "cell4"
  },
  "styles": {
    "display": "flex",
    "flex-direction": "column"
  }
},
{
  "slotName": "Cell 5",
  "rules": {
    "grid-area": "cell5"
  },
  "styles": {
    "display": "flex",
    "flex-direction": "column"
  }
}
```

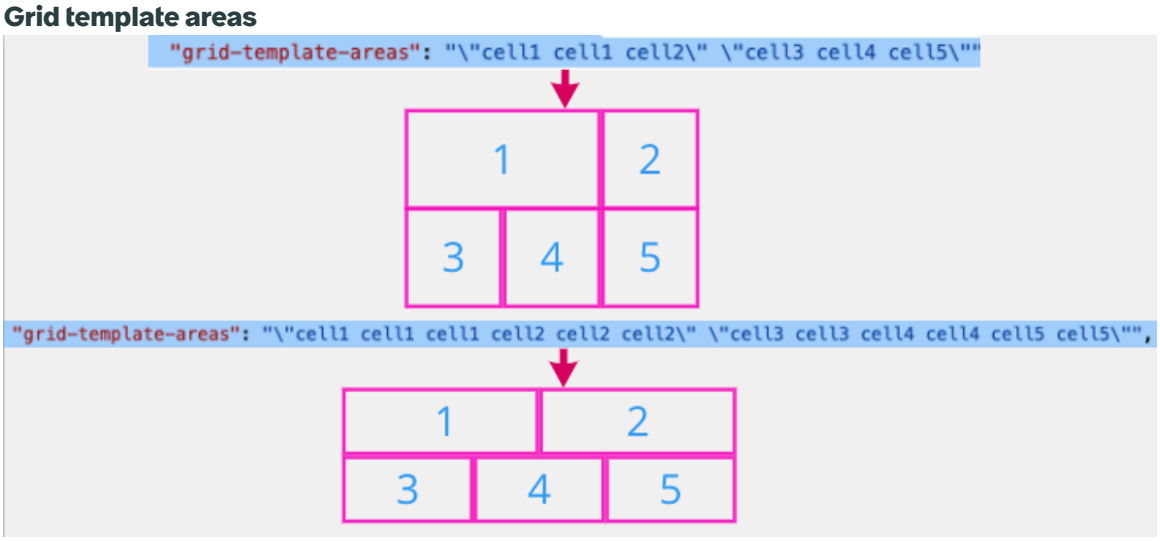
9. Modify the layout rules to include the new column and template area, as follows:
   a. Delete the `"Grid-template-columns": "1fr 1fr",` line.
   b. Add a second `cell1` after `cell1`, and then add `cell5` after `cell4`.

Modify the layout rules

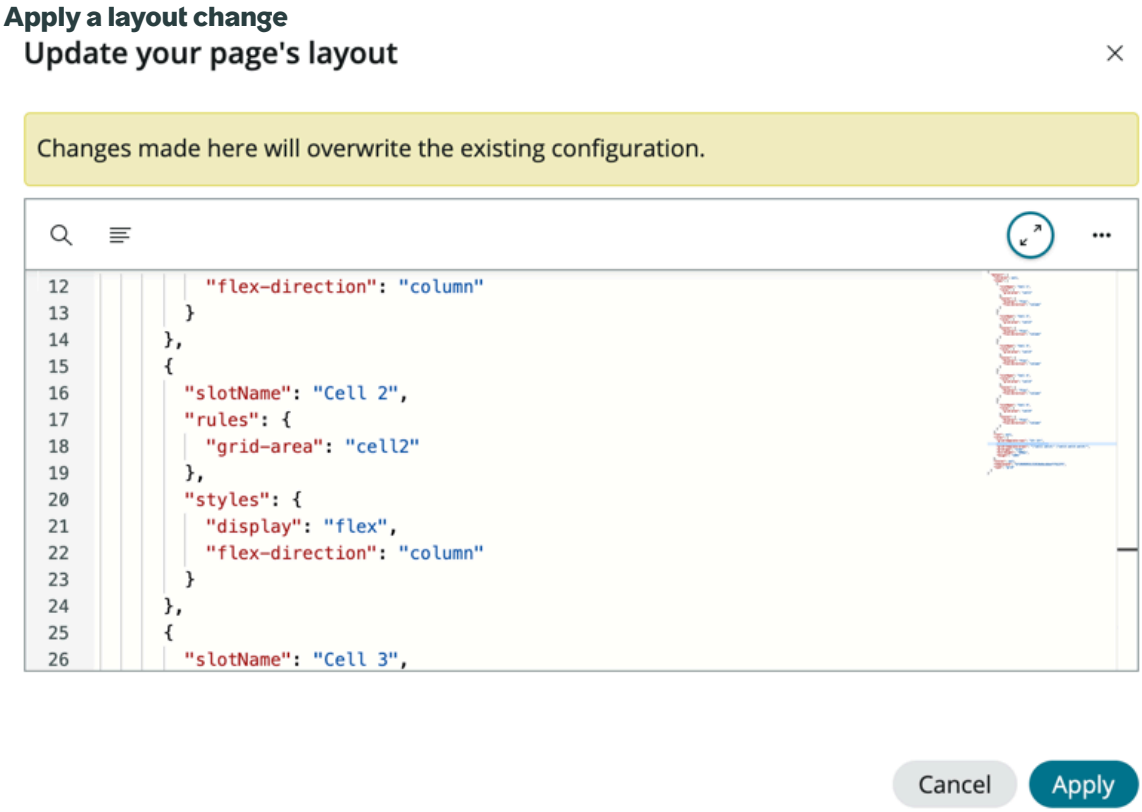


You can set many different configurations. In this example, there are the two resulting grids from two different grid template areas. The *"grid-template-areas"* property being modified is setting the CSS property of the same name. For more information, see MDN grid-templates-areas.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



10. Collapse the expanded view, and then click **Apply**.

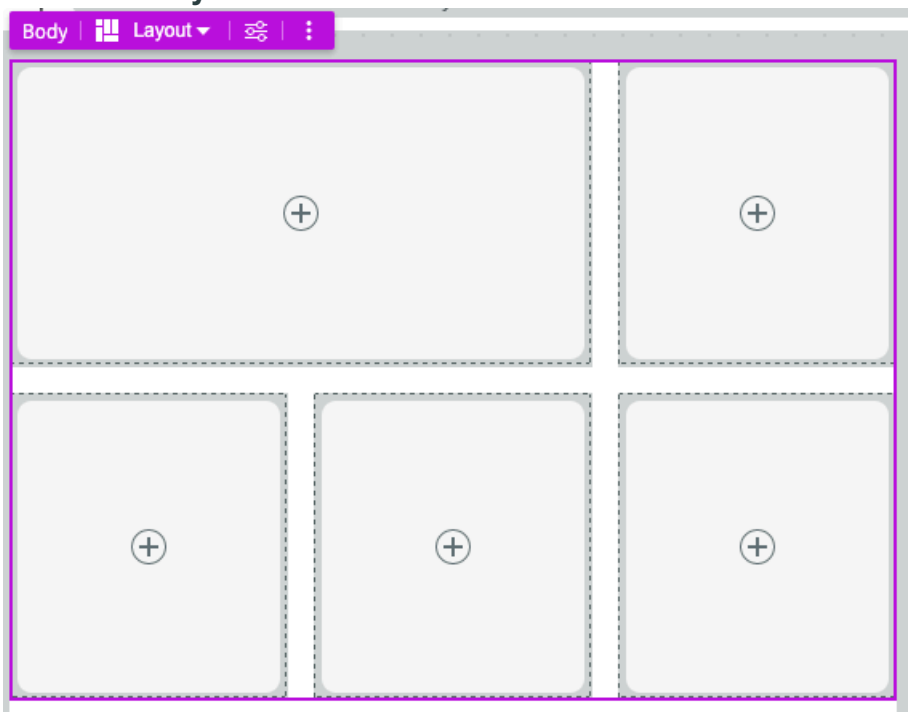


Result

The new CSS Grid layout shows that the new slot was added to the original two-row, two-column layout.

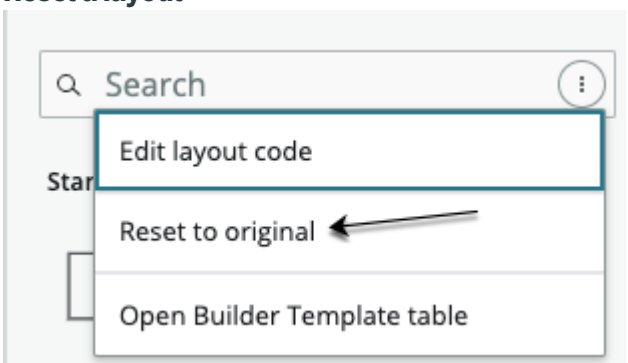
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

New custom layout



Click **Reset to original** to reset the layout changes back to the original at any time.

Reset a layout



Related topics

Create a Flexbox layout with the old layout system

Change the layout of a page created in Quebec or Rome

Add and modify your layout design to change the way your page looks. Choose how components are displayed on a page through Cascading Style Sheets (CSS) web layout technologies, such as Flexbox and CSS Grid.

Before you begin

Role required: ui_builder_admin

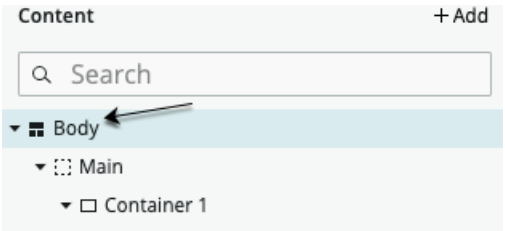
About this task

Layouts control what containers and components are available on a page, and where they are. CSS rules apply to them. You can change the layout as follows.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create or open a page.
4. Select the **Layout** tab.
5. Depending on the type of page, do one of the following to select a container to update.

Type of page	Instruction
Existing page	Select the container on your page for which you want to change the layout. **Note:** Alternately, you can select the container in the content tree to the left of the main page area. When the page structure is in the structured tree, you can more easily find the container you want to change. If you have many com ponents and containers in the content tree, use the search bar to find a specific container.
New page	In the content tree to the left of the main page area, select the **Body** of your page. **Body** is the root element of the page. 

If you are starting a new page, you can select the layout at the page level, and later at the container level. If you are not the owner of an existing page, be aware of the impact of changing the layout at the page level.

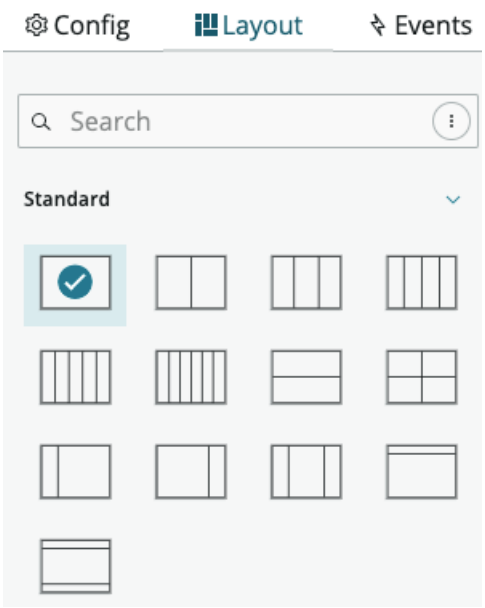
6. Change the layout of your container.

   a. In the Page configuration pane, select the **Layout** tab to select the layout that you want to use.

   You can change the current layout of an existing page into a new one. For example, in a three-column layout, you can click the four-column layout to change the layout.

   You can also set the layout for a container. The following image shows the layout options for the UI Builder.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



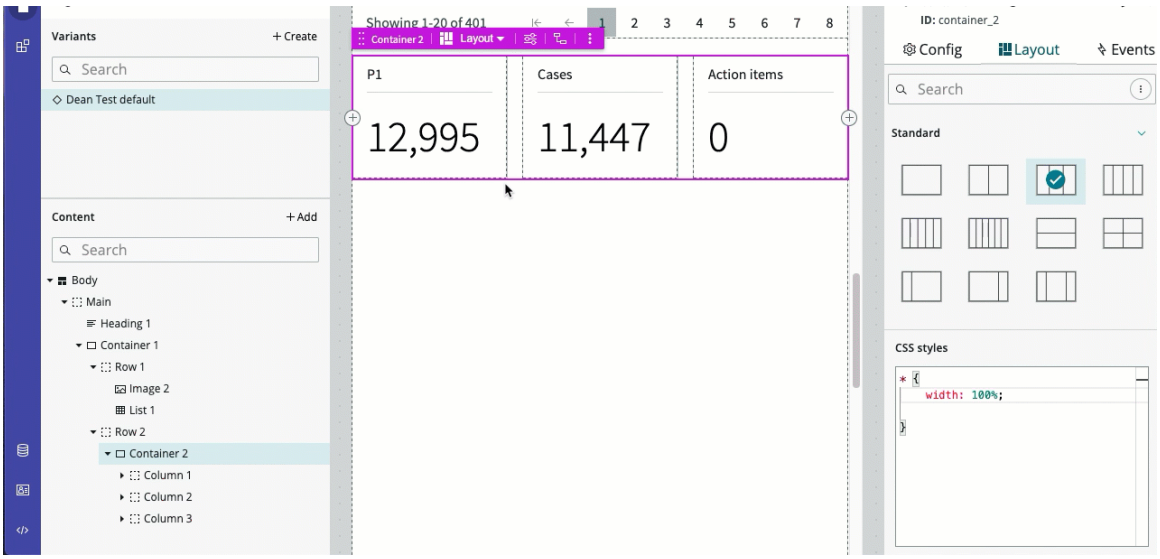
b. If you are creating a page, add components to the new areas in your layout.

c. Click **Save**.

d. View and test your page by selecting Preview.

Example

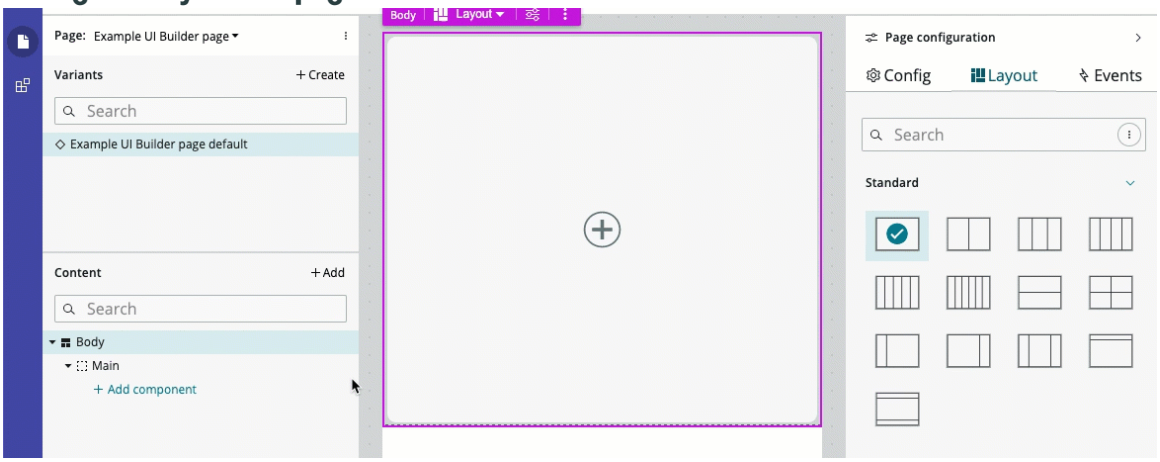
For changing an existing page, the following video shows how you can change the container layout from three to four columns, and then add a new component to the fourth column.



For setting the layout of a new page, the following video shows you how to set the layout that you want for the page. For example, you can set your page to have two slots or columns. Then, you can add containers to each slot and change the layout for each container. You can also set the layout at the container level after you add containers to your page.

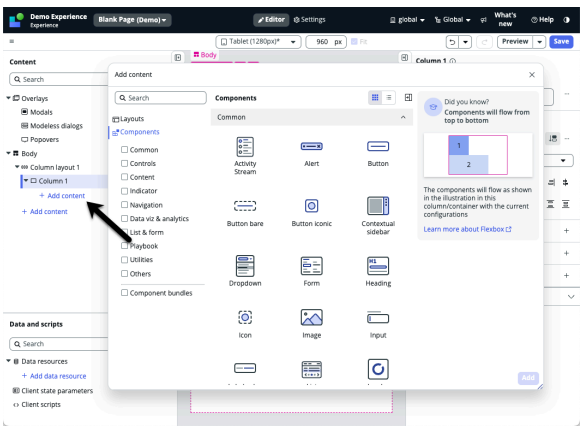
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Change the layout of a page



7. Add components to slots.
   You build your page with containers components. See Customize UI Builder pages using components for more information.

   You can add components by using any of the following ways.

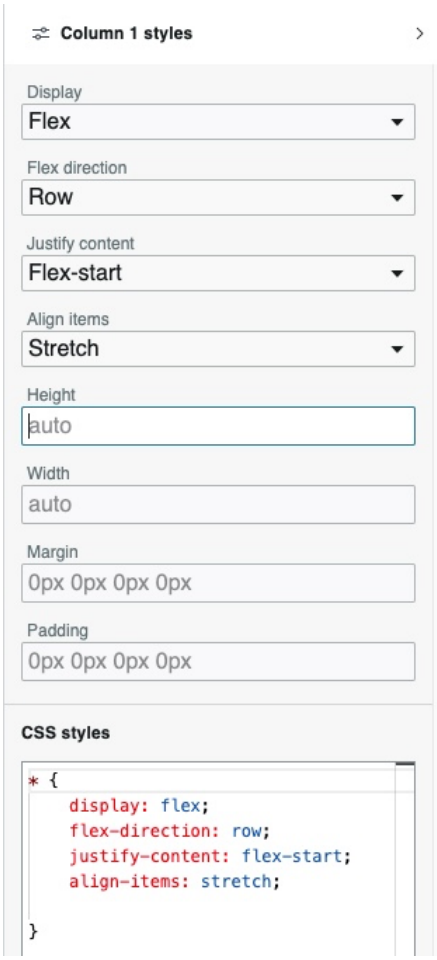
To add a component	Do the following
Directly from your page (option 1)	Select the add content (plus) icon, select the **Components** tab, and then select a compo nent.
Directly from your page (option 2)	On a column layout or component, select the add before (plus) or add after (plus) icon, and then select a component.
From the content tree	In the content tree, select **+ Add content** and then select a component. 
From the floating menu above the page	On a column layout or component, select the Menu icon, select **Add before** or **Add after**, and then select a component.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

8. Modify the layout styling options in any of the following ways.

Option	Instruction
Change the layout styling	**a.** Click a container for which you want to change the layout. You can click the container in the content tree, or you can click the container on the page. **b.** From the floating menu above the page, click **Layout**. **c.** Choose to display your content as either Flex or Grid. For more information about flex or grid options, see Using Flexbox layouts in UI Builder or Using CSS Grid layouts to build a page.
Choose Flex	**a.** From the Display list, select **Flex**.  **b.** Select the following options based on how you want to display your container content on the page. The options can also be en tered in the CSS styles box: ▪ Flex direction: Choose from **Row**, **Row-reverse**, **Column**, or **Column-reverse**. ▪ Justify content: Choose from any of the following: ▪ **Flex-start**: Items are at the start of the flex direction, similar to left-justified content. This is the default setting. ▪ **Flex-end**: Items are at the end of the flex direction, similar to right-justified content. ▪ **Space-between**: Items are distributed evenly. ▪ **Space-around**: Items are distributed evenly with equal space around them. ▪ **Space-evenly**: The spacing between items is equal. ▪ Align items: Defines how you want your flex content displayed along the cross axis. Choices are as follows: ▪ **Stretch**: Stretch your content to fill the container. This is the default setting. ▪ **Flex-start**: Place your content at the start of the cross axis. ▪ **Flex-end**: Place your content at the end of the cross axis.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Instruction
	▪ **Center**: Center your content in the cross axis. ▪ **Baseline**: Align your content the same as your baseline alignment. ▪ Height: Set the height automatically, or manually. ▪ Width: Set the height of your flexbox items automatically or manually. ▪ Margin: Set your minimal distance between flexbox items. ▪ Padding: Set the padding for each side of your flexbox items. 
Select Grid	**a.** From the Display list, select **Grid**.  **b.** Fill in the following information:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

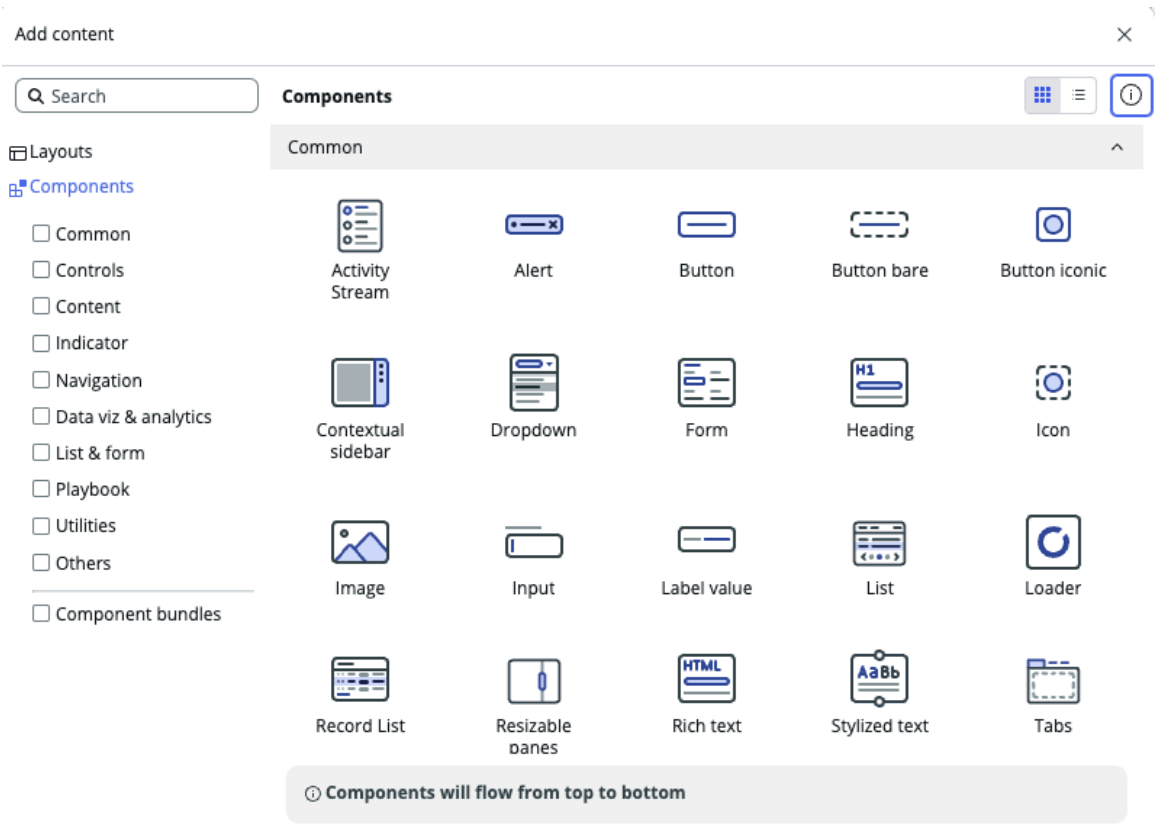
Option	Instruction
	▪ Height: Set the height automatically, or manually. ▪ Width: Set the height of your grid items automatically or manu ally. ▪ Margin: Set your minimal distance between grid items. ▪ Padding: Set the padding for each side of your grid items.

Customize UI Builder pages using components

Learn what a component is in UI Builder. Also, see how components work within UI Builder.

Components are the base elements of your UI Builder page. Components range from core elements like buttons and labels to more complex experience components like lists and forms.

You can add these components to your UI Builder page to build or customize your workspace or portal experience. For example, adding an **Activity stream** component to your page that lets users see their travel request activity.



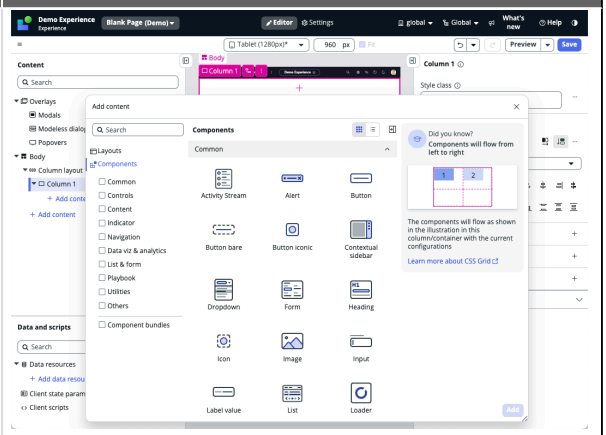
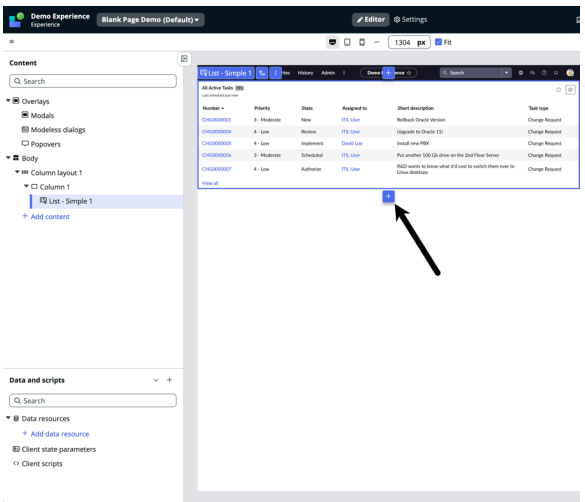
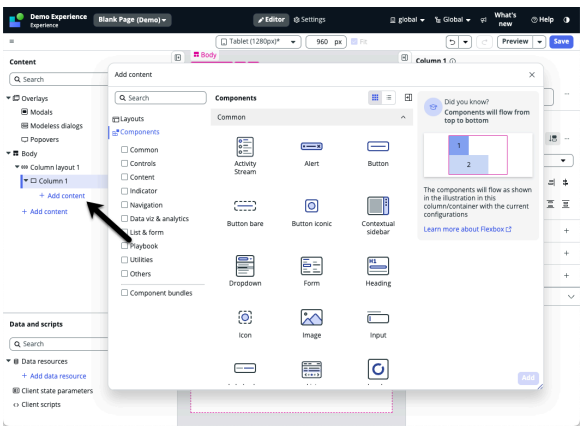
You can add components to your UI Builder page in the following ways.

Ways to add a component to a page

Location	Example
Directly from a page in UI Builder (option 1)	Select the add content (plus) icon, select the **Components** tab, and then select a component.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Ways to add a component to a page (continued)

Location	Example
	
Directly from a page in UI Builder (option 2)	On a column layout or component, select the add before (plus) or add after (plus) icon, and then select a component. 
From the **Content** tree in UI Builder	In the content tree, select **+ Add content** and then select a component. 

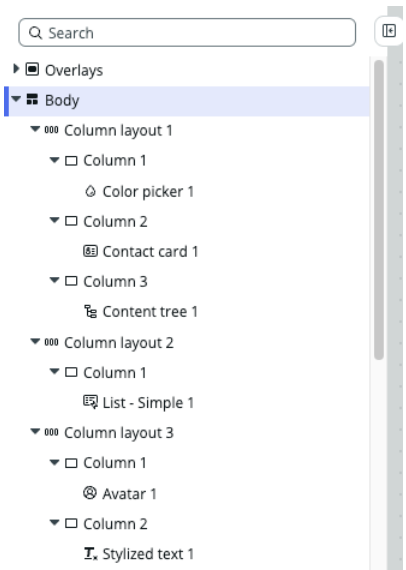
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Ways to add a component to a page (continued)

Location	Example
From the floating menu above the page in UI Builder	On a column layout or component, select the Menu icon, select **Add before** or **Add after**, and then select a component.

Column layouts in UI Builder

Add column layouts and columns to your UI Builder page first. Then add components to the columns in a column layout to build and customize your page. Think of a column layout as a defined part of the screen where you add components like lists and headings. You can have as many column layouts on a UI Builder page as you want, with as many as six columns in any column layout. Multiple components can be added to a single column. View the structure of your page in the **Content** tree.



For more information, see Column layouts.

Configure components in UI Builder

There are three ways to configure a component in the configuration panel.

- Configure the component properties.
- Add CSS style overrides.
- Set up event handlers for the components.



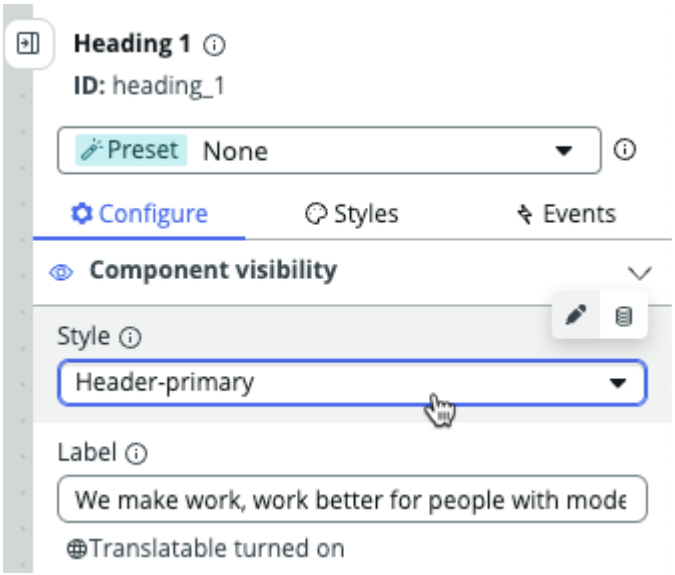
Configure tab

Component properties vary based on each component. Component configuration can be simple, as is the case with simple elements like buttons, headings, and labels. Components like lists and forms require significant configuration. For icon and image components, select from a variety of options or use a custom icon or image. You can modify component properties with UI Builder's low-code JSON

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

editor. For more information about configuring components, see Next Experience Components.

For each property in the component configure tab, you can choose from the following options.



- Static: Use data from a list, or enter your own data. The data doesn't connect to an external data source.
- Dynamic data binding: A way to bind a component property to a data resource, page property, or client state.
- Script: Enter JavaScript code to populate a property value.

Styles tab

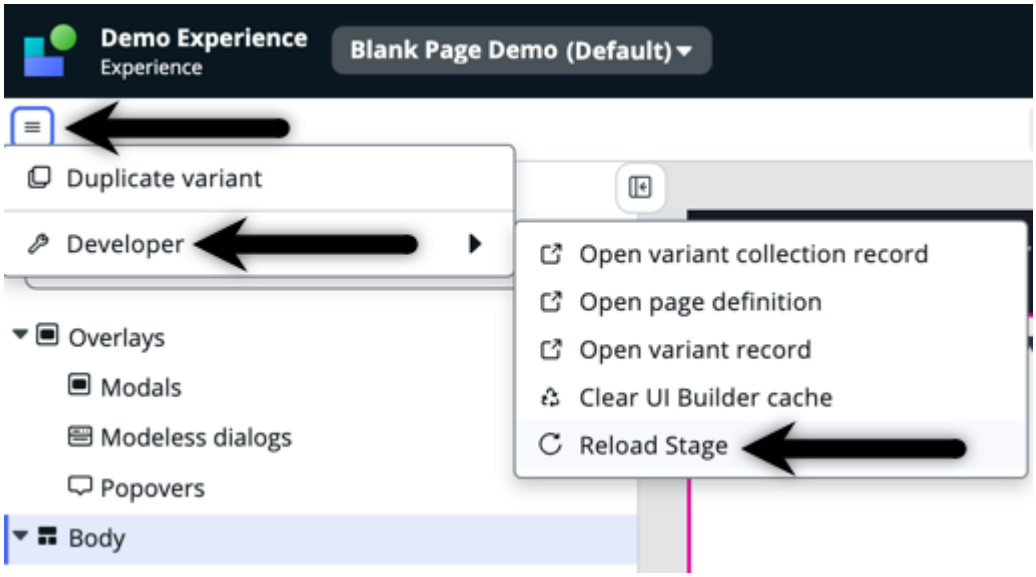
You can change the Cascading Style Sheets (CSS) styles for individual components. Change color borders, font size, and so on.

Events tab

Configure page-level and variant-level event mappings. Also add dispatched events and handled events for your variant. The more complex, experience components rely heavily on dynamic data that is provided by a data resource. Binding dynamic data to a component is an important feature. You dynamically expose data from tables, records, or other elements on your page. Exposing data enables you to reuse your components. Also, you can point to the configuration fields to see icons data or scripting options for each field.

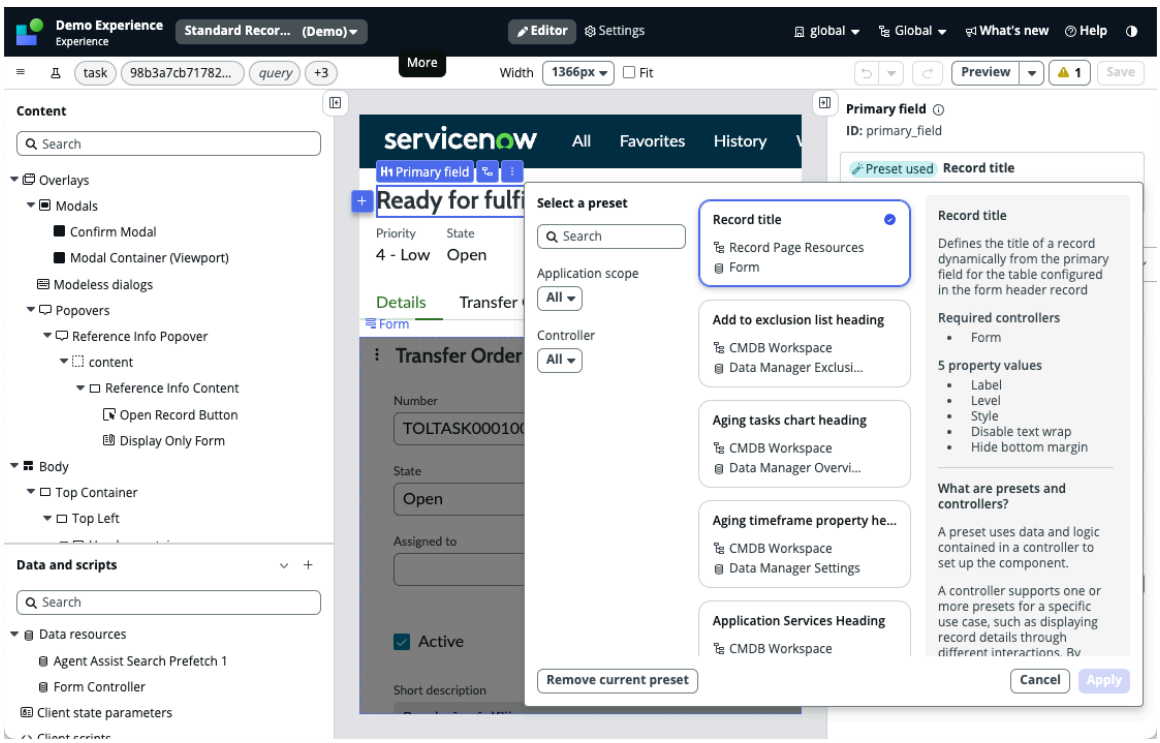
As you add and configure components on the page, the stage shows your work. If you make changes and the updates do not load on the stage, select the **Open menu** icon and then select **Developer > Reload Stage**. Reloading the stage shows your changes, but does not save them. To save your work, select **Save** at the top right.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Component Presets in UI Builder

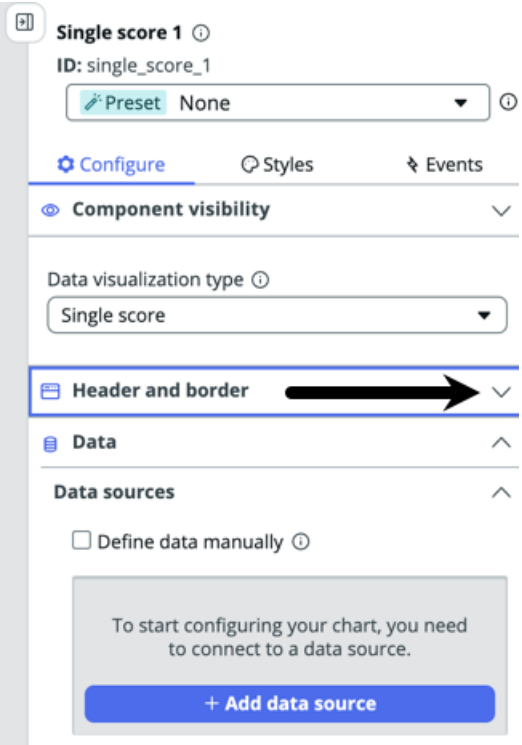
Use component presets to automatically configure components on compatible pages. UI Builder page templates contain controllers that presets use to define component configuration values. For more information, see Automatically configure components using presets.



Component properties sections

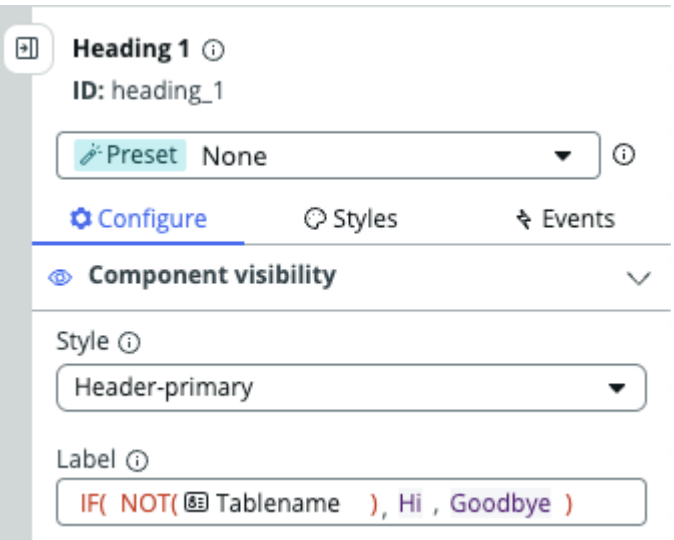
Component properties are grouped into sections of similar properties. Open and close sections by selecting the arrows to the right of the property headings. UI Builder admins can select which component configuration properties are displayed or hidden based on UI policies.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Component formula editor

Use the UI Builder component formula editor in the configuration panel to bind or modify formulas. You can enter text, data bindings, and/or formulas. Formulas can consist of any combination of the three input types. The component formula editor supports logical, arithmetic, comparison, negation, and functional transform types.



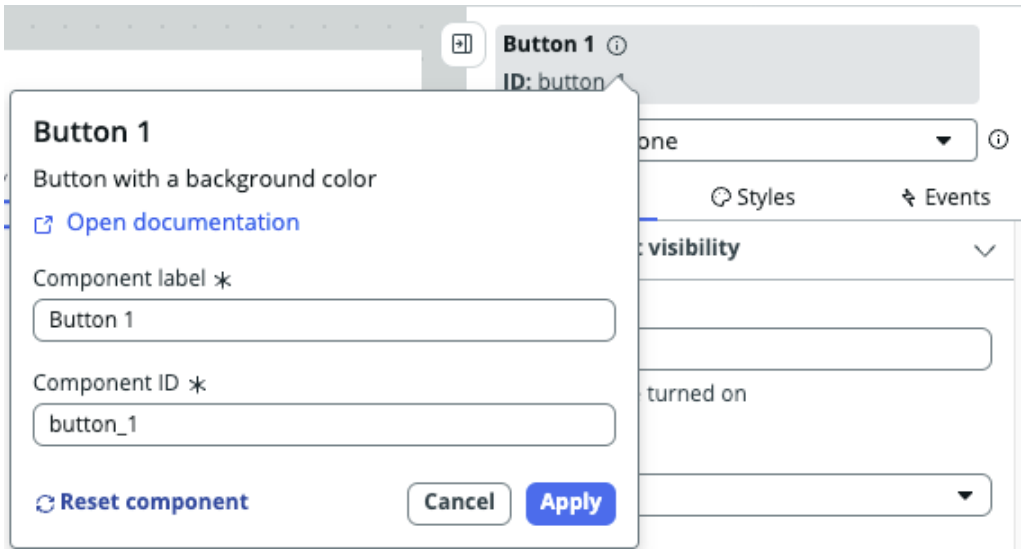
Each function added to the component formula editor auto-completes. The component formula editor displays a label for each input of the function so you know what parameters each function requires. For more information about the supported functions in the component formula editor, see Supported functions in the UI Builder component formula editor.

Component ID

Use the component ID when you add a script or bind data to the component as a way to reference the component. A component ID is automatically created and is based on the

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

component label when you add a component to a page. You can change the component ID to anything you want, as long as it is unique. Select the name of the component in the configuration panel to see the component ID.

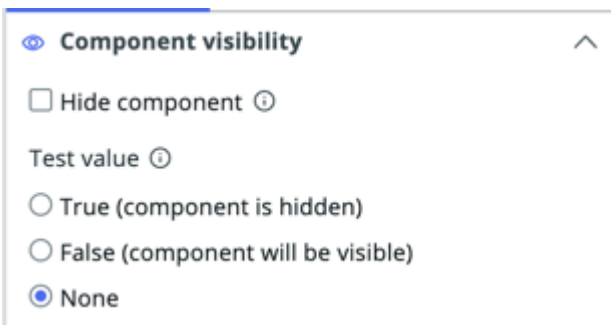


Component visibility

Select the eye icon in the configuration panel to set component visibility. Component visibility is based on a property of the component itself, not who is viewing it. You could show or hide a component based on conditions. For example, hiding an image if it has a broken link.

You can set the visibility based on dynamic data binding, or by editing a scripted property value.

Set the **Test value** to test what happens when the visibility is set to true, false, or none.



Duplicate components

Create an exact copy of a configured component on your UI Builder page except for the name and ID. A duplicated component copies all properties, bindings, and events. For more information, see Duplicate a component.

Automatically configure components using presets

Use component presets to automatically configure components on compatible pages.

Use component presets to apply predefined configuration values and event mappings to components. UI Builder page templates contain controllers that presets use to define component configuration values.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Presets apply pre-built configurations to component properties and event handlers and are only available for certain components. They are based on common use cases for components, such as configuring a **Form** component with fields that are typically included on a record page.

Applying a preset automatically configures components to work immediately, simplifying page creation and maintenance for page authors.

Presets offer the following benefits to page authors:

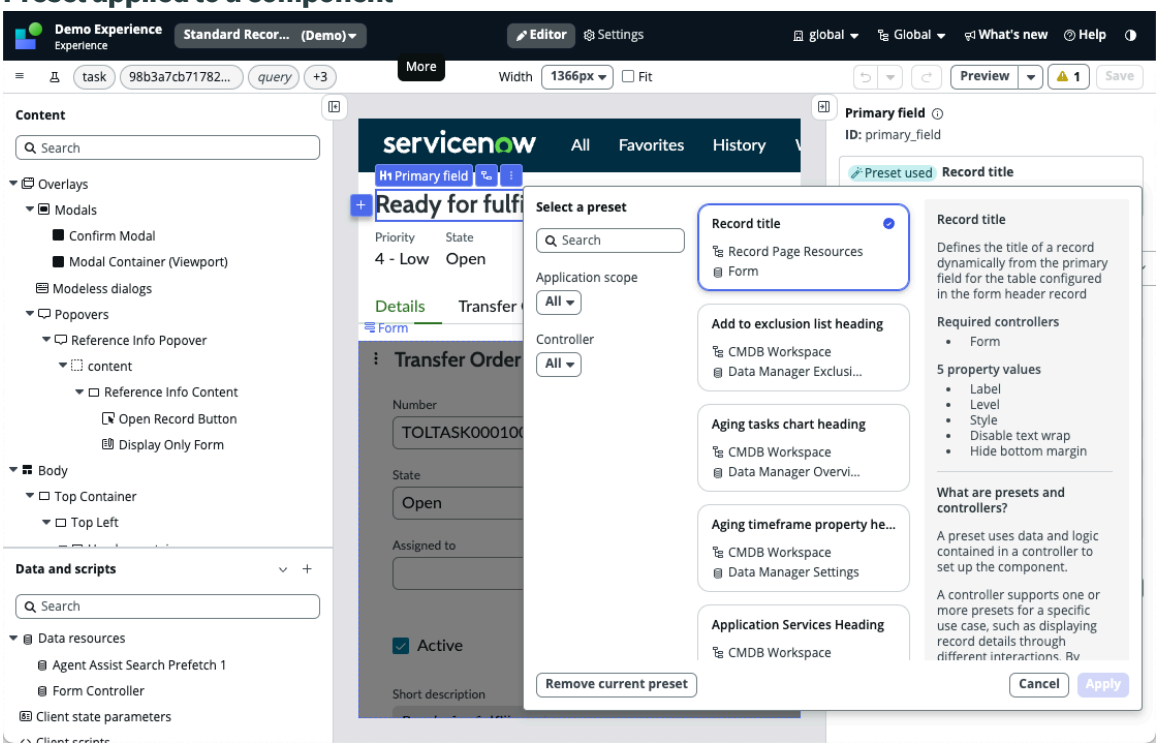
- Handling of complex data and event binding for components
- Reduced cost of ownership and maintenance due to being defined external to pages

From the Configuration pane, you can select whether to apply a preset to the selected component or instead manually configure it. Default presets are automatically applied for components when a page is created from a template or the controller required for the preset is already added to the page. Any properties or events configured by the preset display as read-only when a preset is applied.

You can override values configured by a preset but in doing so you assume ownership of the component configuration and maintenance.

Sub-pages do not inherit controllers and are not able to use presets in Brazil.

Preset applied to a component



Data and event bindings

Presets can include bindings to:

- Controller output properties (@data)
- Event payloads (@payload)
- Session context (@context.session)
- Complex formulas (client transforms)

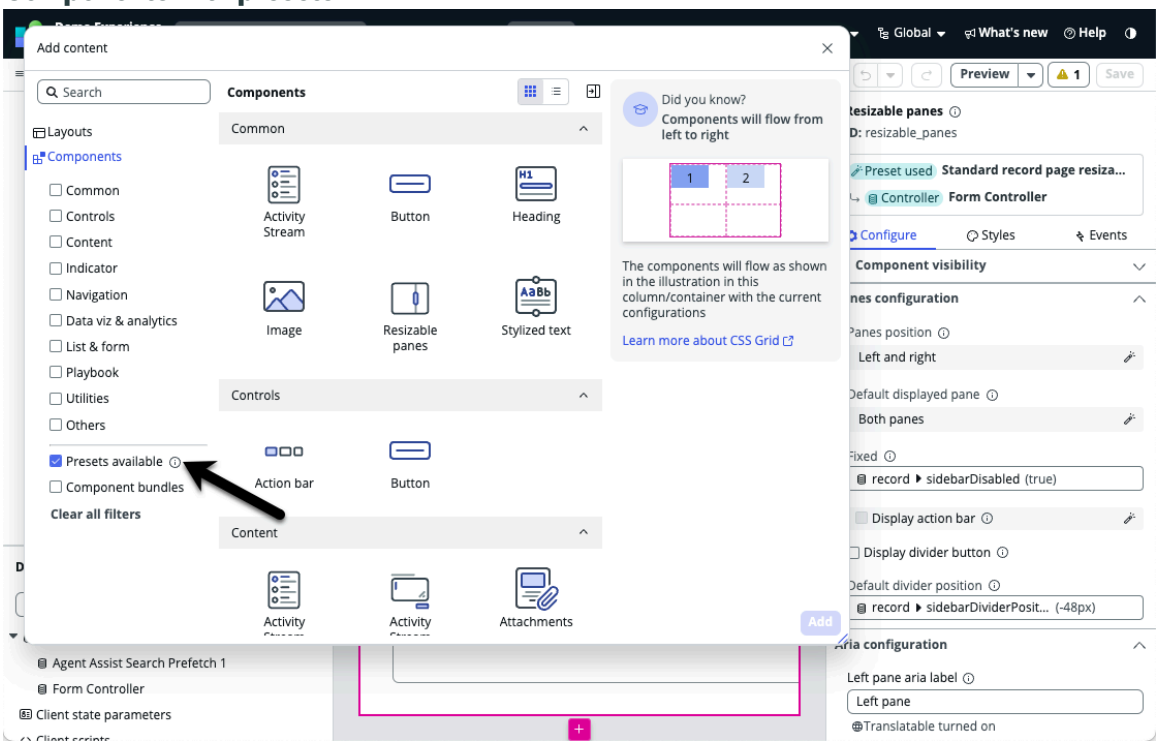
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Presets can also include event mappings to a controller's handled events. For more information, see Manage actions in UI Builder pages.

Controllers

Presets connect components to data and event mappings using a controller. If the controller required by a preset is not already on the page, the preset prompts you to configure the controller's required properties and adds the controller. After a controller is added to a page, components that have presets associated with the controller appear highlighted in the components list. For more information, see Bind data to UI Builder pages using controllers (advanced feature).

Components with presets



Check the **Presets available** box to view components with presets in the toolbox.

Select a component preset

Choose a component preset when adding a component to your page.

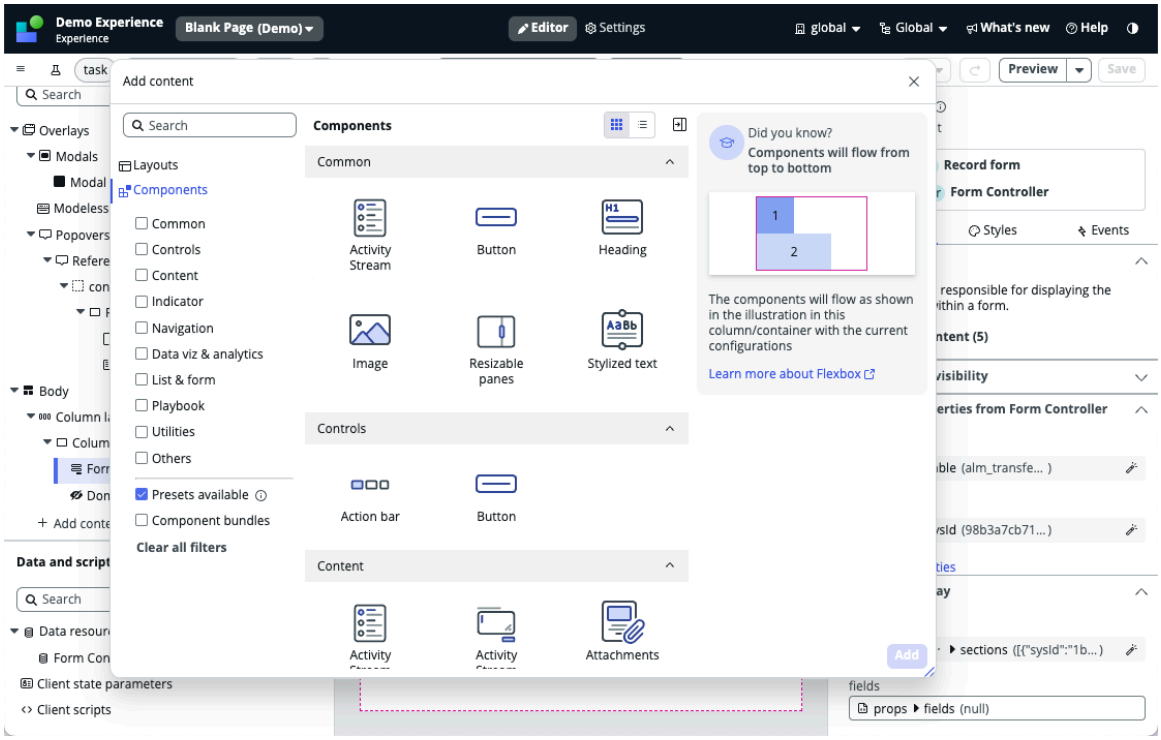
Before you begin

Role required: admin

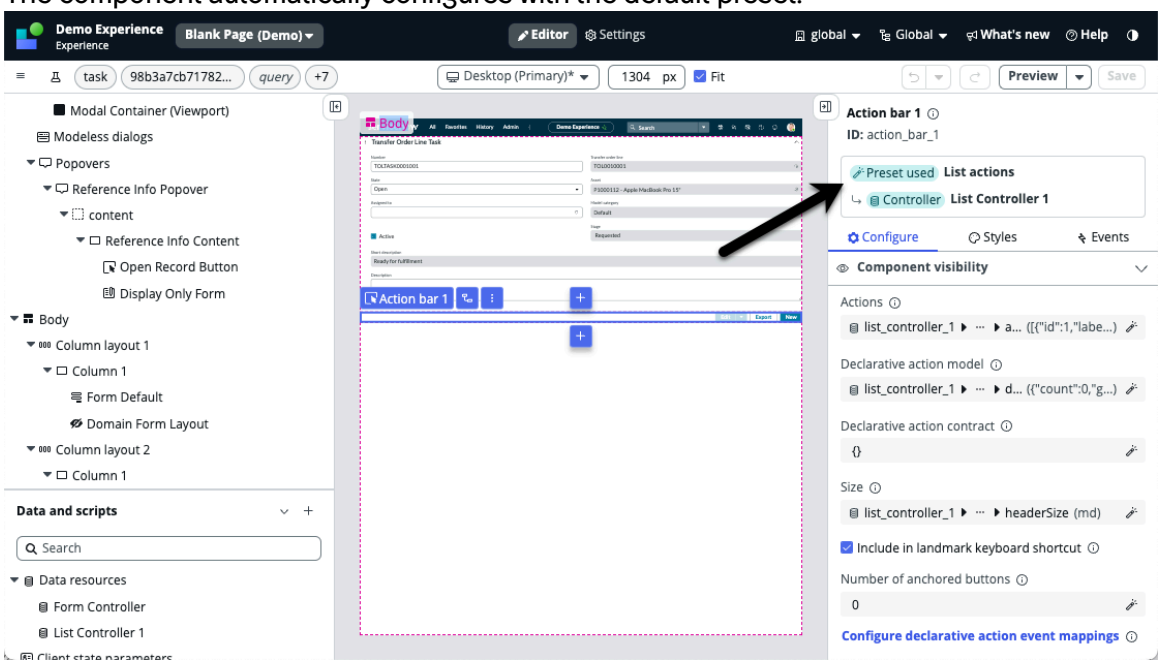
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create or open a page or page variant.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. Open the toolbox.
5. Select the **Presets available** box.
6. Select the component you want to add to your page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



7. The component automatically configures with the default preset.

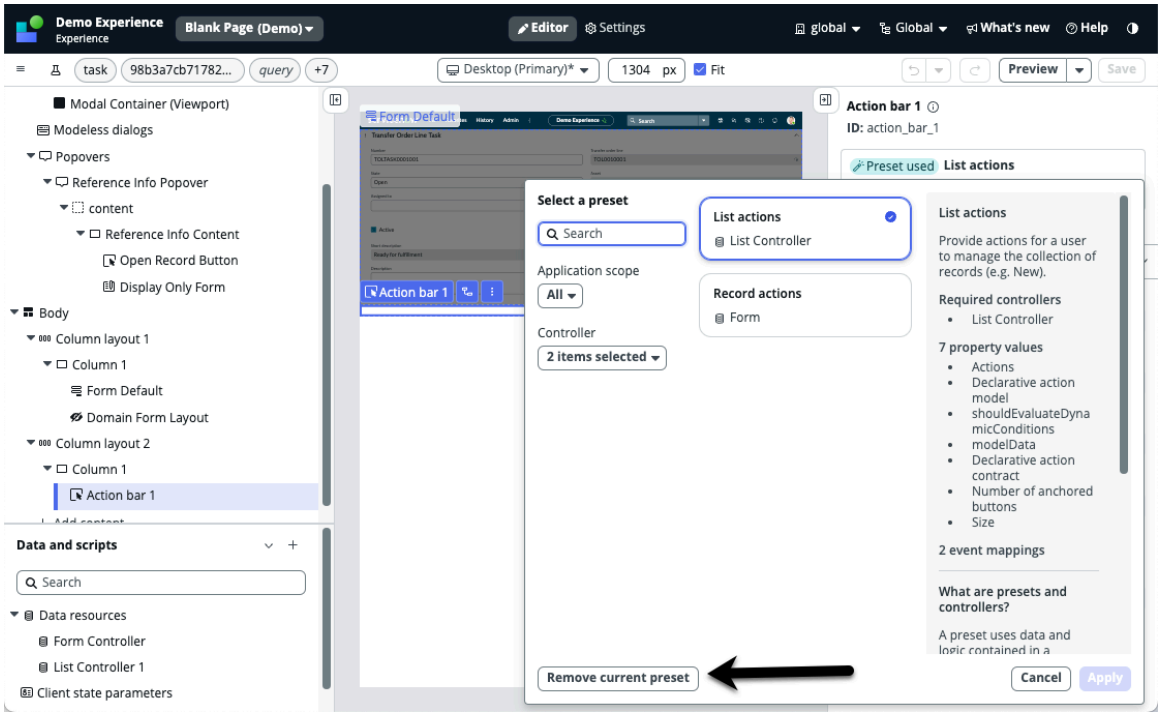


8. **Optional:** Select a different preset from the list in the configuration panel.
   You can configure a component not to use a preset by selecting **Remove current preset**.

   a. Select a new component preset in the configuration panel.

   b. Select **Apply** to replace the existing preset.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



9. Click **Save**.

Override a component preset

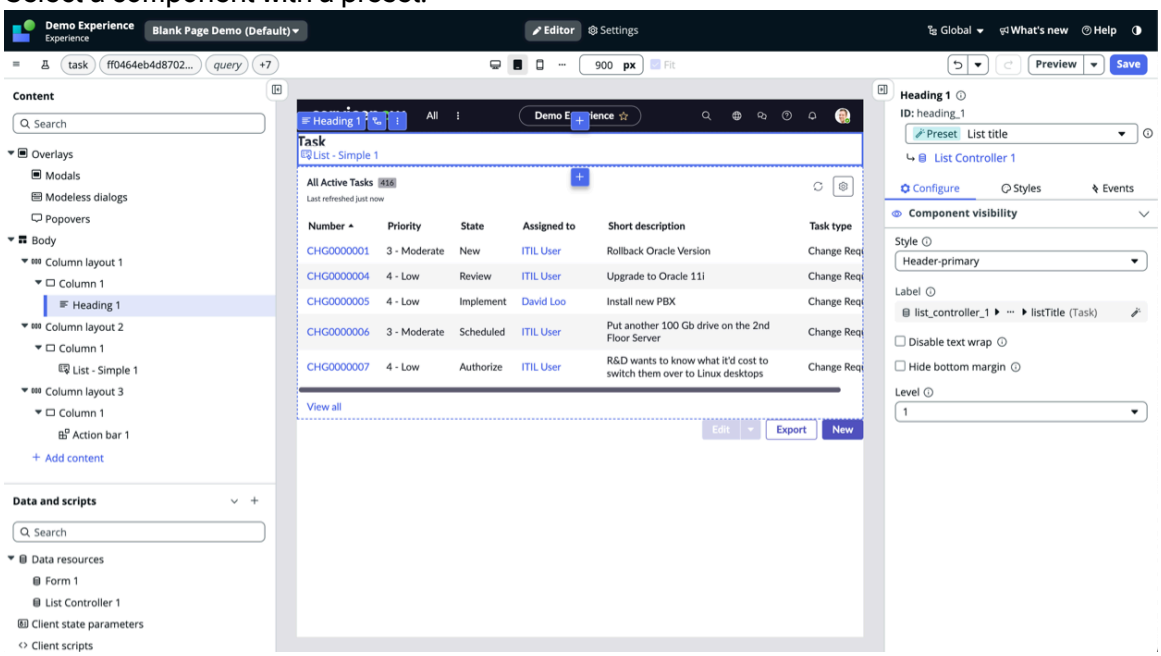
Override a component preset to enter your own custom values.

Before you begin

Role required: admin

Procedure

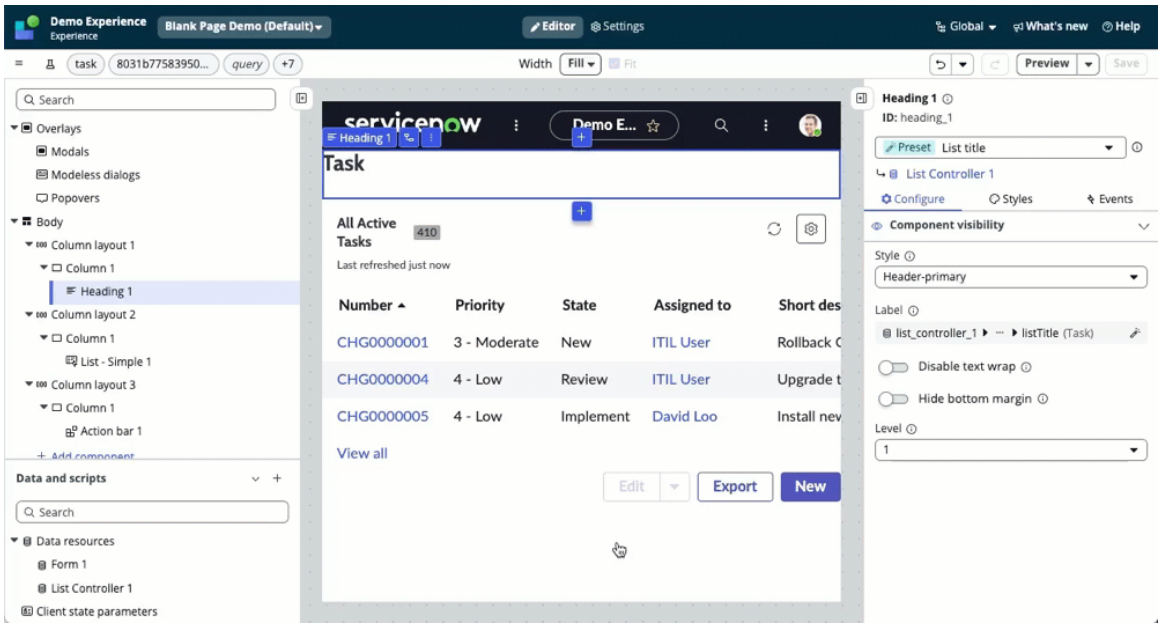
1. Select a component with a preset.



2. Find the data value that you want to override in the configuration panel.
3. Click the lock icon in the data field.
4. Enter the data value that you want to override the preset value.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

5. Select **Save**.



Result

The configuration panel displays the custom value in the field.

Reset a component

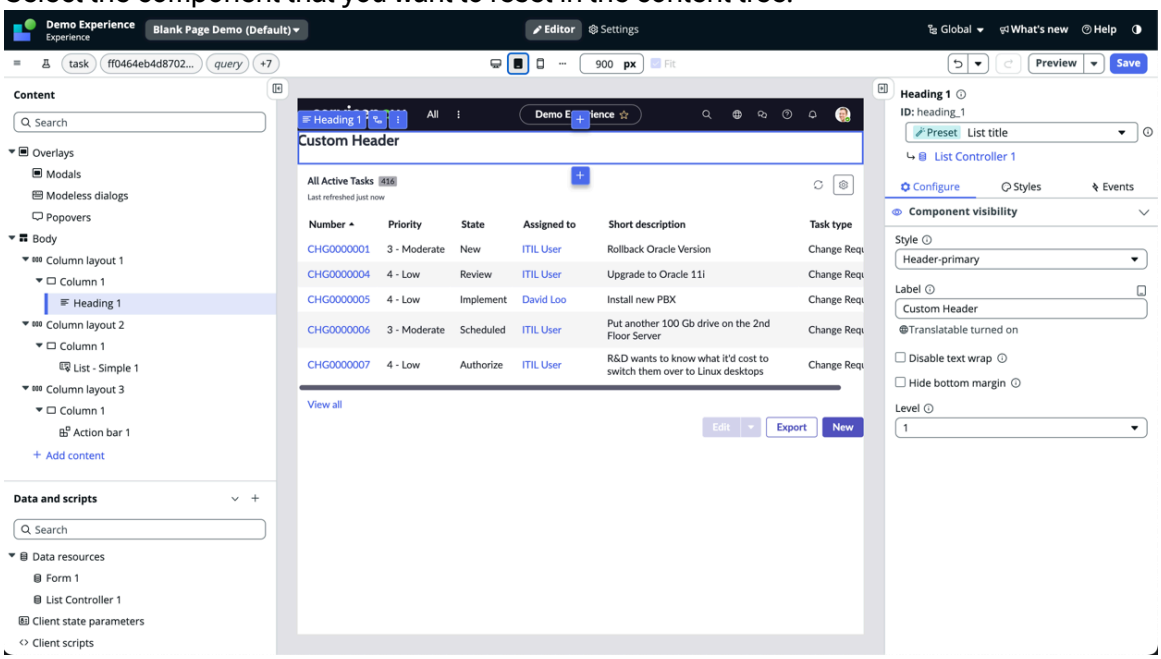
Reset component presets to their default values.

Before you begin

Role required: admin

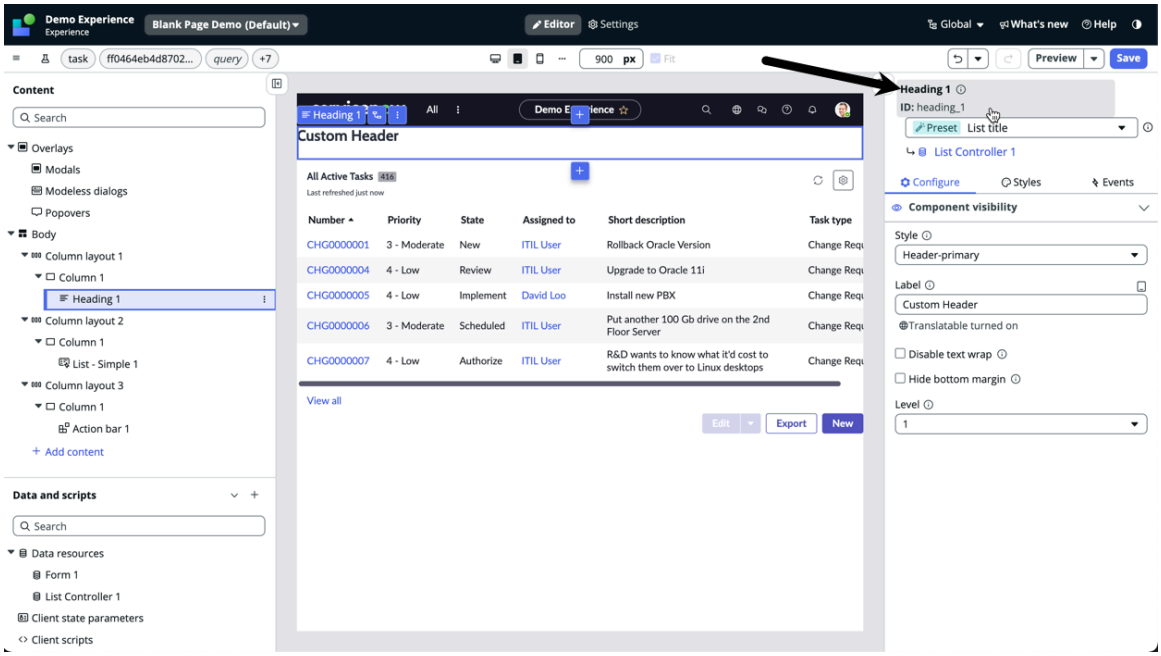
Procedure

1. Select the component that you want to reset in the content tree.

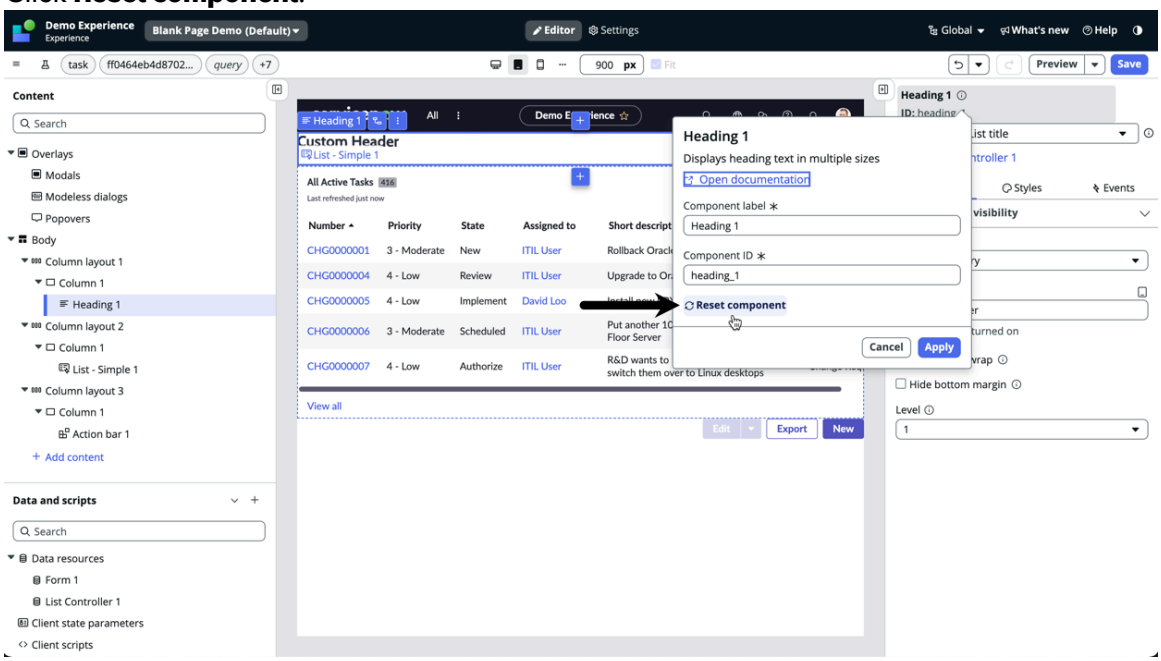


2. Select the component name in the configuration panel.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



3. Click **Reset component**.



4. Click **Reset**.

Result

The component preset resets to the default values.

Create custom presets for components

Create presets to save time and make your component configurations reusable across different experiences.

Before you begin

Role required: ui_builder_admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

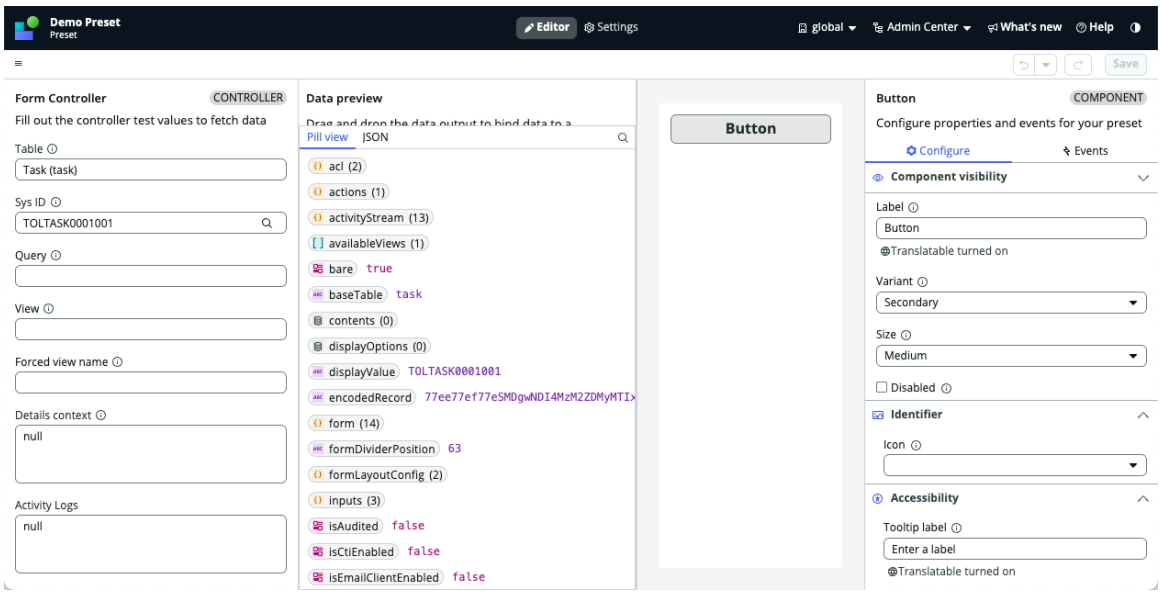
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Select **Create**.
3. Select **Preset**.
4. On the form, fill in the fields.

Create a page collection form

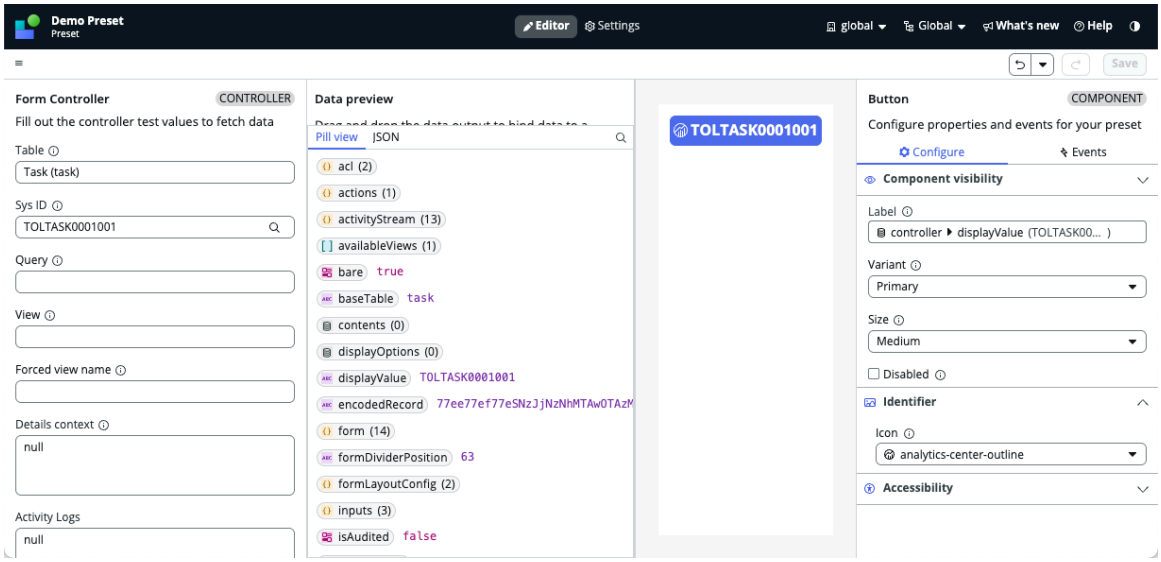
Field	Description
Name	Name to track your preset internally.
Component	Select the component you want to create a preset for.
Controller	Select the controller with data and logic to power the preset.
Description	Short description to help find your preset. Write a description that helps page builders understand what content is included in the preset.

5. Select **Create**.
   The preset building window opens.



6. Enter controller test values to preview data in the component preset.
7. Configure the component properties you want to apply with the preset.
8. Bind data from the controller to the component preset by dragging items from the data preview section to the fields in the configure panel.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



9. **Optional:** Add events to the component.
   For more information, see Configure an event handler manually.
10. Select **Save**.

What to do next

Apply the preset you just created by selecting a preset in UI Builder.

Change the default appearance of components

Set the styles for components and wrappers to change the default appearance.

This video shows you how to perform the following procedure.

Before you begin

Role required: ui_builder_admin

About this task

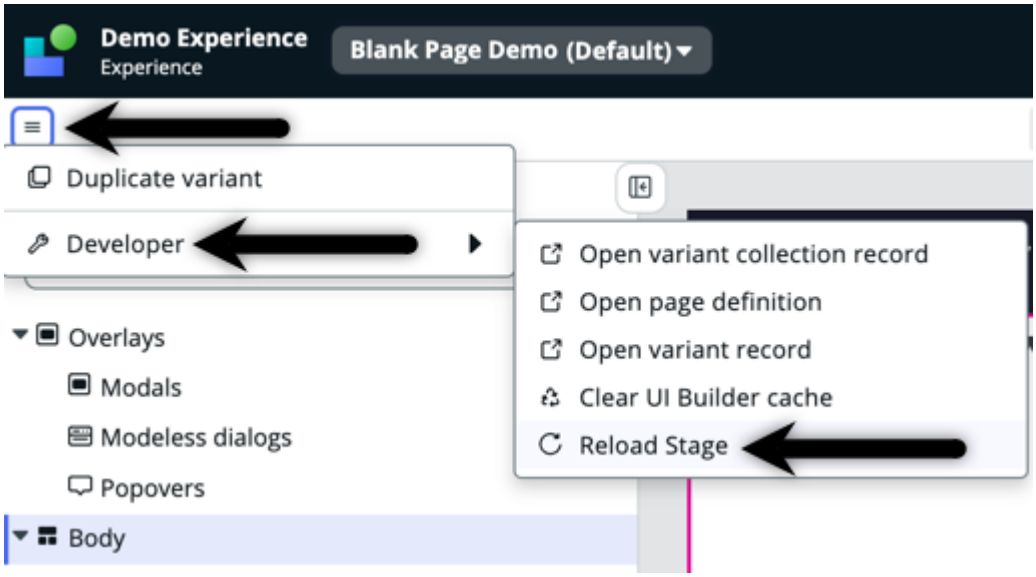
This task describes how to add styles to components and the wrappers (such as Body, column layouts, or a column) containing components. Customize component styling by selecting the component in the Content tree or apply styles to multiple components by placing components inside a wrapper.

For information about adding styling to your entire experience, see Manage the visual style of UI Builder experiences.

This task applies to the new layout system introduced in Brazil. If your page is using the old layout system, see Add styling to a component using the old layout system for more information.

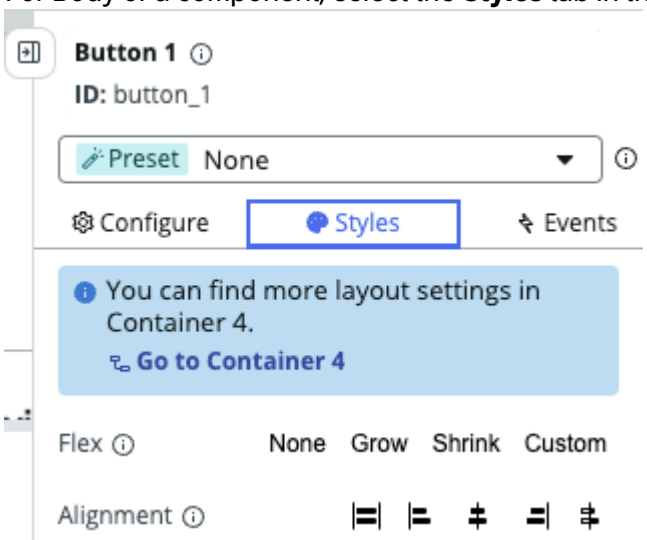
Note: As you add and configure components on the page, the stage shows your work. If you make changes and the updates do not load on the stage, select the **Open menu** icon and then select **Developer > Reload Stage**. Reloading the stage shows your changes, but does not save them. To save your work, select **Save** at the top right.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create or open a page or page variant.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. In the Content panel, select a component or wrapper (for example, Body, a column layout, or a column) for which you want to add styling.
5. For Body or a component, select the **Styles** tab in the configuration panel.



6. Use the visual representations on the **Styles** tab to decide what is best for the look and feel of your page.

   Accessibility (Column layout and Column)

   Specifies accessibility settings such as ARIA Region Name, ARIA Region Heading Level, ARIA Role, and inclusion of ARIA Heading. For detailed information, see the ARIA documentation on developer.servicenow.com.

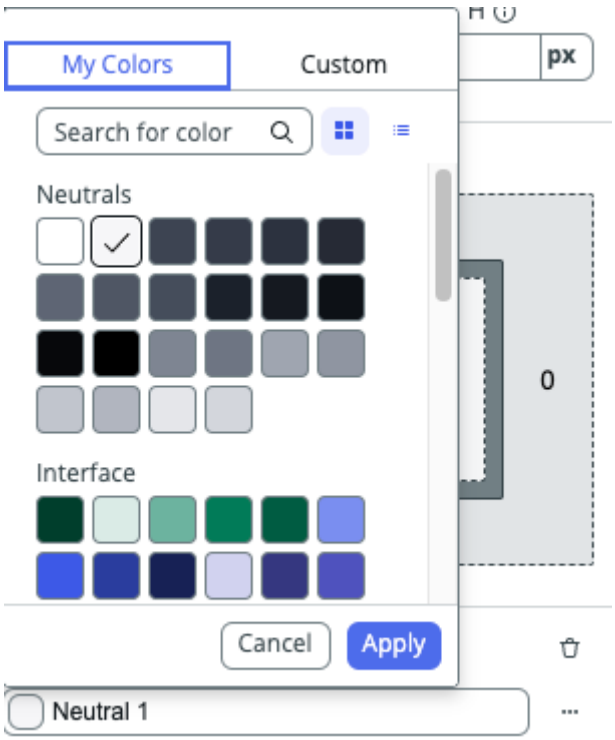
   Alignment (Column and Component)

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For a column, defines how components align within the column as a group. For a component, defines how the component aligns within the parent layout element, such as a column or column layout.

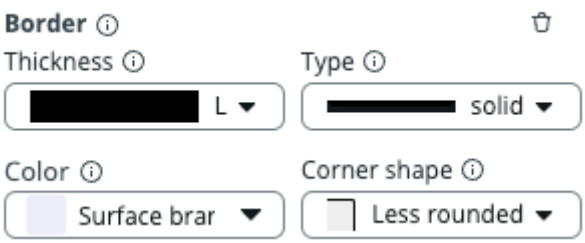
Background (Body, Column layout, Column, and Component)

Sets the background color behind the layout element. Select a color from the color picker **My Colors** tab where you can view available colors in a grid or a list. Alternatively, use the **Custom** tab to specify any color in HEX, RGB, or HSL.



Border (Column layout, Column, and Component)

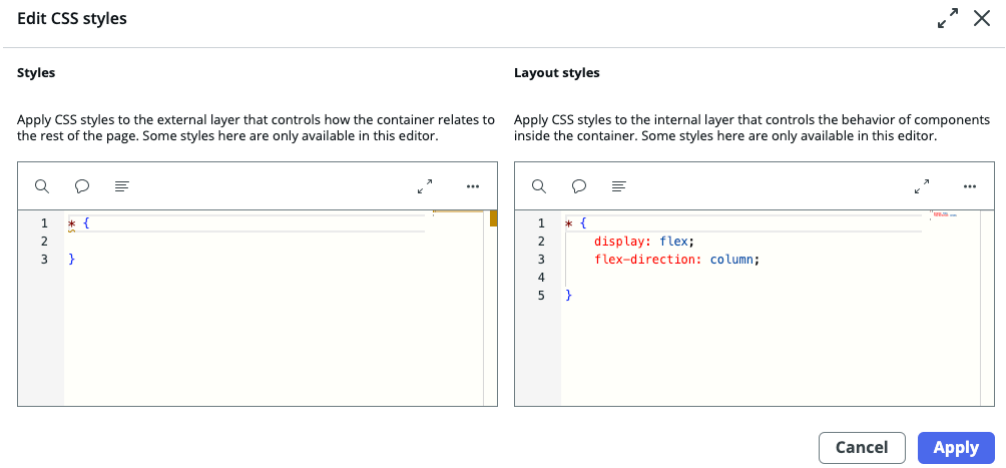
Adds a border around the content. The border is placed on the immediate inside of the margin and the immediate outside of the padding. Specify the border thickness, type, color, and corner shape. For components, you have the option to use custom CSS for the border.



Layout (Body, Column layout, Column, and Component)

For body and column, specifies direction, alignment, and content justification. For column layout, specifies the number of columns and the column gap (space between columns). For some components, such as Card Base Container, select **Edit CSS** to apply specific CSS styles to the container layer and to the internal layer controlling components within the container.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

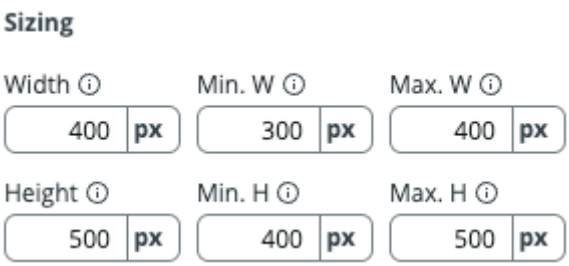


Shadow (Column layout, Column, and Component)

Adds shadow effects around the content. You have the option to use custom CSS.

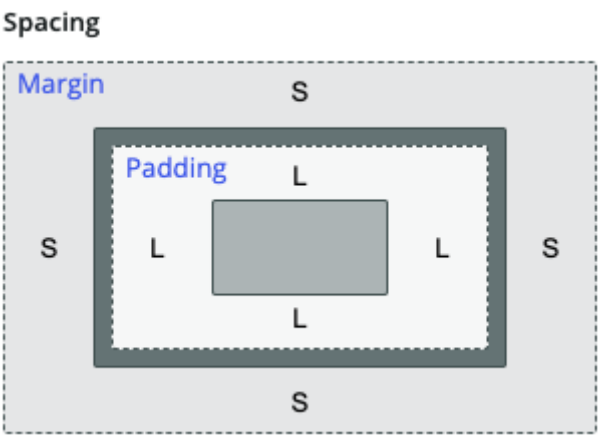
Sizing (Component)

For applicable components, sets the default, minimum, and maximum height and width for the component (in px, %, em, rem, or a custom value). This value is relevant for when the browser window is resized.



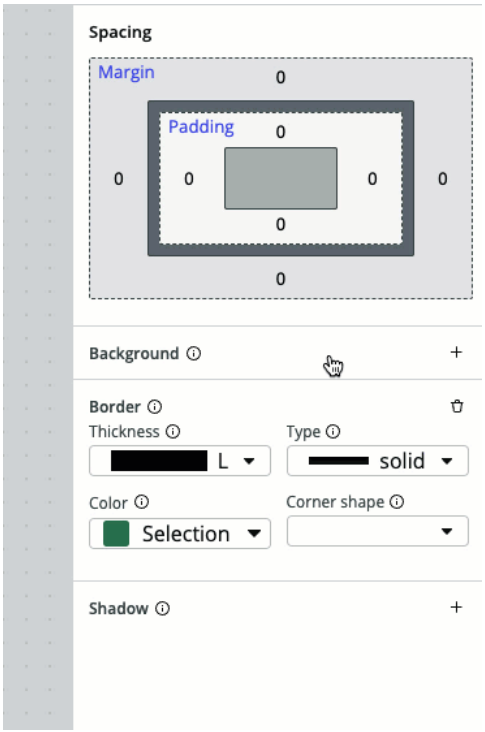
Spacing (Body, Column layout, and Component)

Margin sets the size of the space on the immediate inside of the body, column layout, or component. Padding sets the size of the space on the immediate outside of the body, column layout, or component. Select Margin or Padding to set the same size for all four sides.



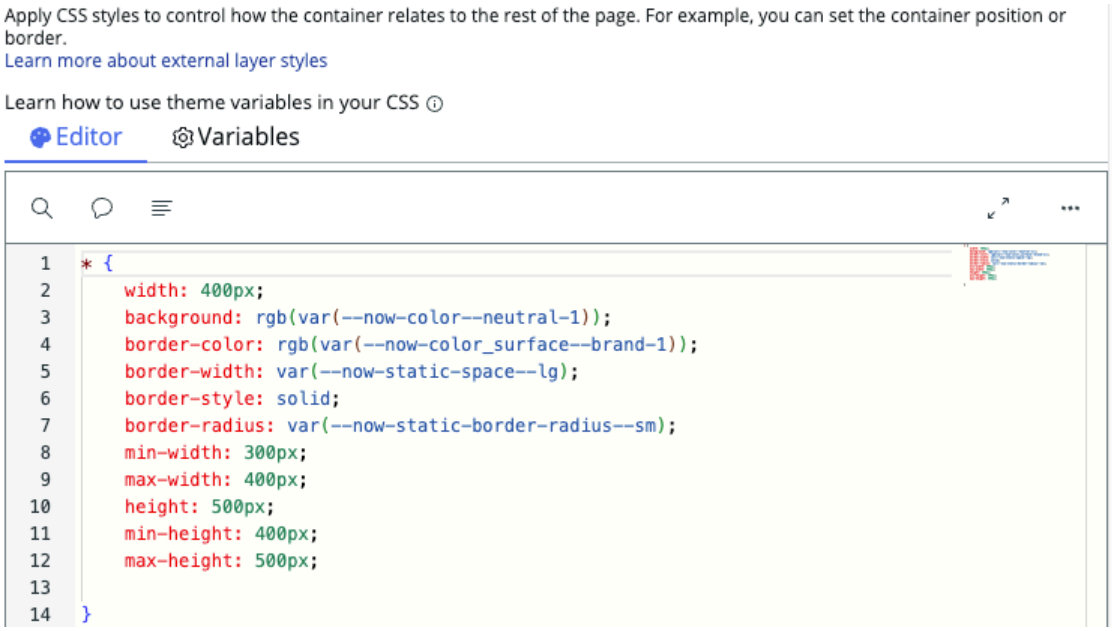
You can set the size for each margin or padding side by selecting the current setting and then selecting an option.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



7. **Optional:** For Body and components, you can edit the CSS code by selecting the **View and edit CSS** link at the bottom of the **Styles** tab.
   The **Styles tab** displays a **CSS styles** code editor. The following CSS properties are the most commonly used:
   - `background-color`
   - `background-image`
   - `border-style`
   - `border-width`
   - `border-color`
   - `border-radius`
   - `box-shadow`
   - `height`
   - `min-height`
   - `max-height`
   - `margin`
   - `overflow`
   - `padding`
   - `width`
   - `min-width`
   - `max-width`
   - `z-index`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Important: Some components contain built-in styling configurations that you can't override with CSS in UI Builder. For information about overriding these style configurations, see Manage the visual style of UI Builder experiences.

8. In the main header, select **Save** to save your changes.

Add styling to a component using the old layout system

Set CSS styles for a component to change its default appearance.

Before you begin

Role required: ui_builder_admin

About this task

This task describes how to add styles to the wrapper containing your component, which is generally recommended instead of applying styles to the component directly. Make sure that the component whose styles that you want to define is placed within a container component to put the component in a wrapper. Your component's wrapper is one level higher than the component in the Content hierarchy and is labeled by default as `Main`.

To add styling to an entire page, you can use standards-based CSS in the wrapper for the page. For information about adding styling to your entire experience, see Manage the visual style of UI Builder experiences.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create or open a page or page variant.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.

   Note: A component must be placed in a container before adding styles to the component.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

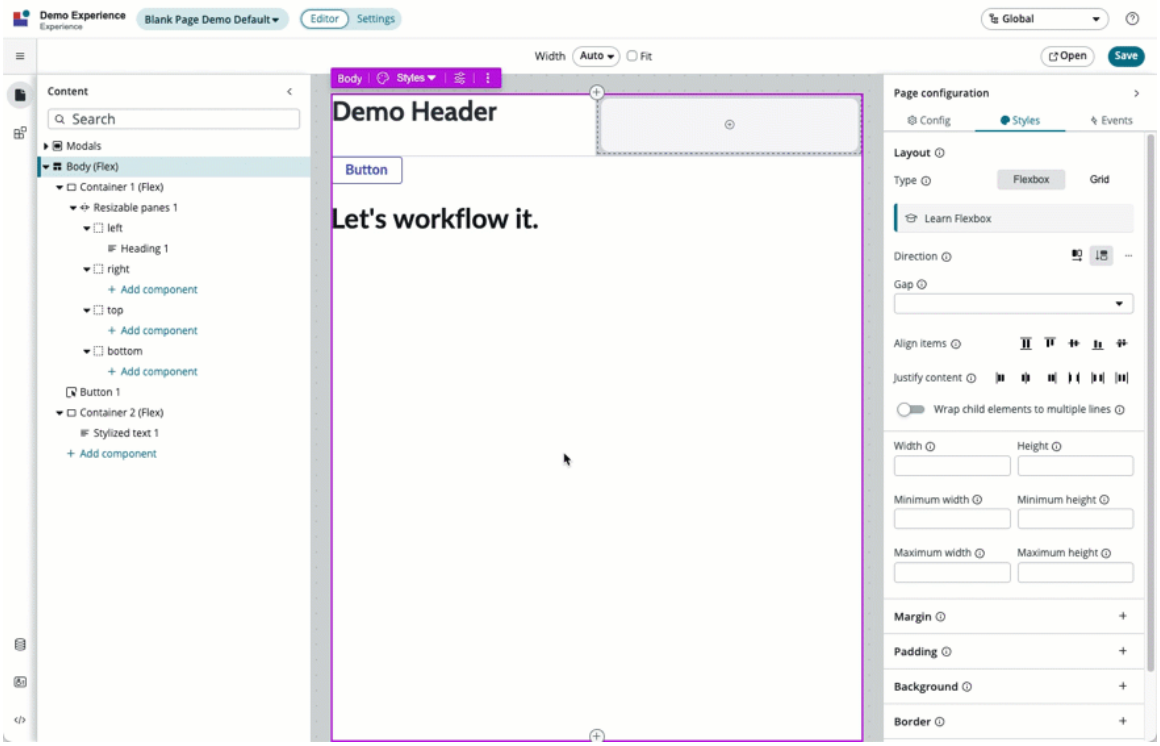
4. Select the container holding the component you want to add styling to.
   The container is typically one level higher than the component in the content tree.
   The Styles tab appears in the configuration panel.
5. In the CSS Styles window, enter standards-based CSS properties and values.
   The following CSS properties are the most commonly used to apply styles for components within containers:
   - `background-color`
   - `background-image`
   - `border-style`
   - `border-width`
   - `border-color`
   - `border-radius`
   - `box-shadow`
   - `height`
   - `min-height`
   - `max-height`
   - `margin`
   - `overflow`
   - `padding`
   - `width`
   - `min-width`
   - `max-width`
   - `z-index`

> **Important:** Some components contain built-in styling configurations that you cannot override with CSS in UI Builder. For information about overriding these style configurations, see Manage the visual style of UI Builder experiences.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

6. In the main header, select **Save** to save your changes.

Add styling to a component in the Styles configuration panel



Enhance accessibility with focus management

Set focus on specific elements, ensuring users can interact with them effectively using screen readers.

Before you begin

Role required: ui_builder_admin

About this task

Focus management can help users navigate experiences built with UI Builder by directing their attention to specific elements on a page. For example, you can point to a component when closing a modal that updated information on a parent page.

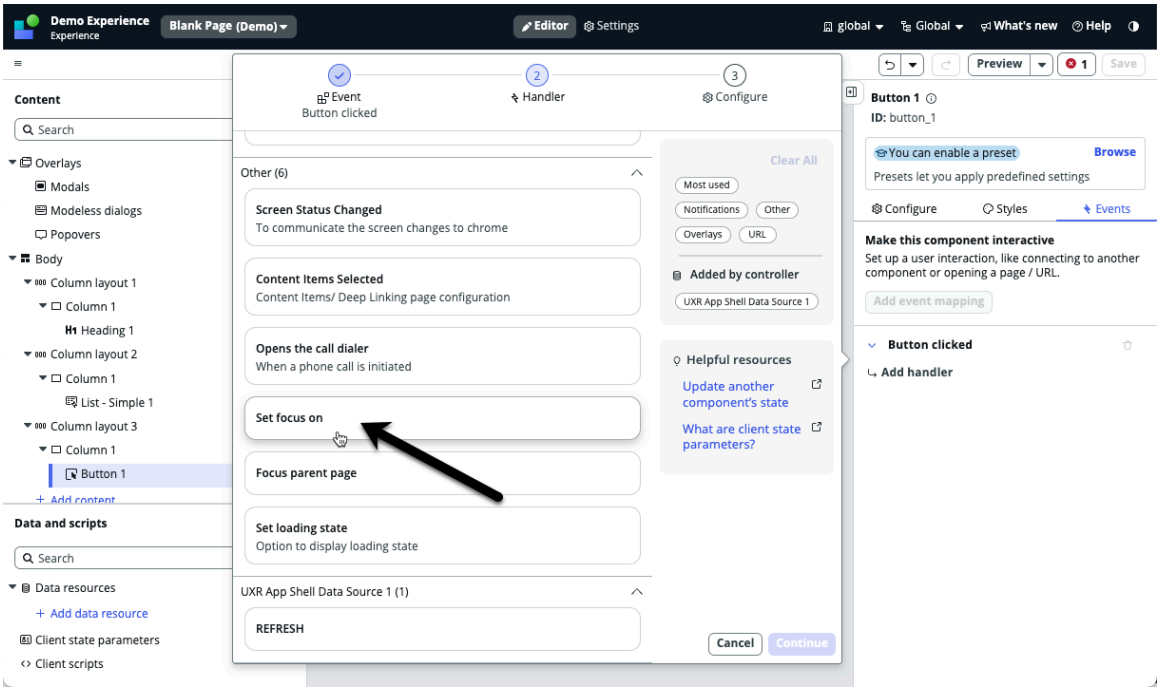
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create or open a page or page variant.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. Select a component from the content tree that you want to trigger the focus.
5. Select the **Events** tab.
6. Select **Add event mapping**.
7. Choose an event from the list.
8. Select **Continue**.
9. Specify the type of focus you want to trigger with the event.

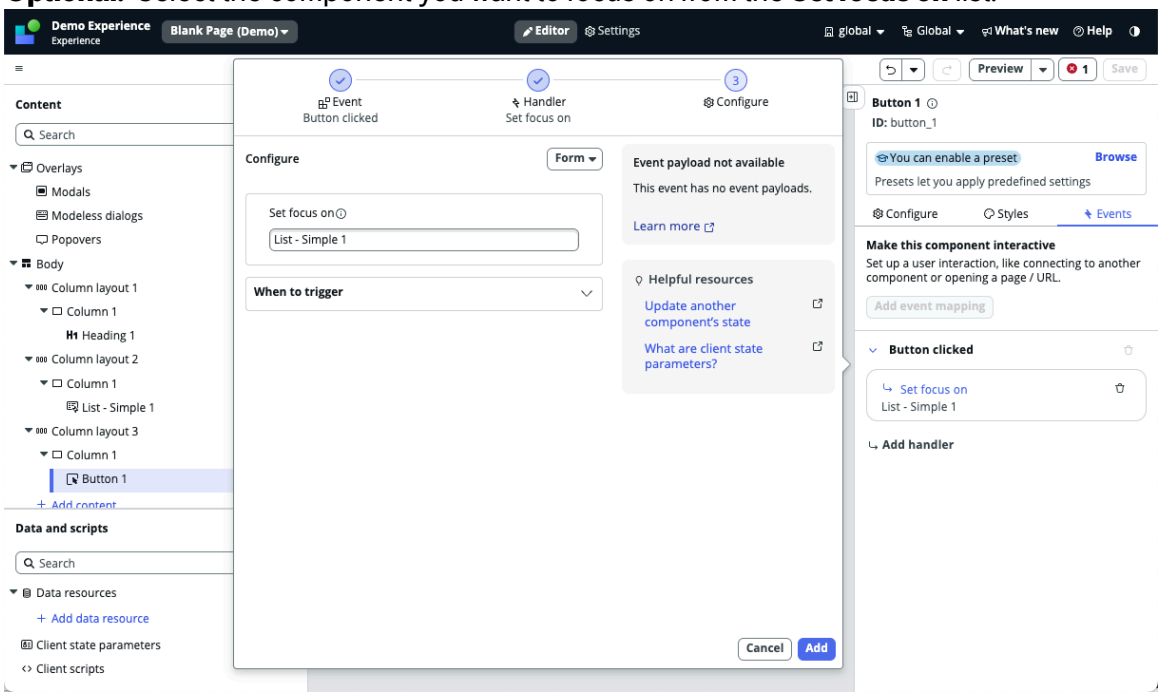
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Select **Set focus on** to direct users to a specific component on your page.

Select **Focus parent page** to direct users to the parent page of a modal, pop-up, or similar overlay.



10. Select **Continue**.
11. **Optional:** Select the component you want to focus on from the **Set focus on** list.



12. Select **Add**.
13. Select **Save**.

Duplicate a component

Duplicate a configured component to reuse on a page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin
Role required: ui_builder_admin or admin

About this task
Create an exact copy of a configured component on your page except for the name and ID. A duplicated component copies all properties, bindings, and events. The duplicated component appears directly after the component that is copied. If a tabs component is duplicated, all tabs and their children are copied as well as their names and IDs are made unique.

Procedure
1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Open the page variant with the component you want to duplicate.
4. Select the component that you want to duplicate.
5. Right-click the component.
6. Select **Duplicate**.
   You can also use the keyboard shortcut **cmd + D** (Mac) or **ctrl + D** (Windows).
   The duplicated component appears below the copied component in the content tree.
7. Select **Save**.

Dynamically display content with conditional renderers

Use the conditional renderer component to display content or components based on specified conditions.

Before you begin
Role required: ui_builder_admin

Procedure
1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create or open a page or page variant.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. Select **+ Add content** in the content tree.
5. Select **Components**.
6. Select **Conditional renderer**.
7. Select **+ Add condition**.
8. Choose how you want to build the condition.

Type of condition	Description
Empty Container	Create a condition by placing components in an empty container.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Type of condition	Description
Card	Create a condition by placing components in a card component.
Single component	Add a condition directly to a component without using a container or card. This works well for showing a single component only when the condition is true.

9. **Optional:** Select a component from the list if you selected **Single component** in the previous step.
10. On the form, fill in the fields.

Field	Description
Component label	Label to identify the conditional content.
Component ID	Unique identifier used when binding data.
Description (Optional)	Description of the condition.
Render content	Select when to render the content on the page.
Condition	Use data or formulas to set the condition for rendering content. The field requires a true boolean value.

11. Select **Apply**.
12. **Optional:** Add components to the container or card.
    For more information about adding components, see Customize UI Builder pages using components.
13. Configure the components in the conditional renderer.
14. Select **Save**.

Learn components by example

Follow these procedures to learn how components create the functionality you need in your UI Builder pages.

Create dynamic filtered card displays

Create an interactive card layout to display records, add a dropdown to filter visible cards by name, and configure each card to open the full record in another UI Builder page when selected.

Before you begin

Role required: ui_builder_admin

About this task

This example shows how to bind data to components so you can display record information in custom layouts. For example, you can present records as selectable cards rather than in a traditional list view. For this procedure, the data comes from the Incident table.

This procedure uses UI Builder components to create dynamic, interactive layouts. For more information on how to configure components, see:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Add and configure components
- UI Builder Quick Bits: Navigating Component Configuration

Components used in this procedure

Component	Documentation links
Card Base Container	• Usage Guidelines • UIB Setup
Dropdown	• Usage Guidelines • UIB Setup
Heading	• Usage Guidelines • UIB Setup
Highlighted value	• Usage Guidelines • UIB Setup
Label value	• Usage Guidelines • UIB Setup
Stylized text	• Usage Guidelines • UIB Setup

Procedure

Part 1: Build the card layout

Add a repeater linked to a data resource and create the card containers that display each record.

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create a page from scratch.
   For more information about how to create a page, see Create a page in UI Builder.
4. Create a data resource to pull records from the Incident table.

   Data resources expose data from tables and records to the page. In this example, we add a data resource to pull records from the incident table so we can display them using components. For more information, see Add and configure data resources to a page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. In the **Data and scripts** drawer, under **Data resources**, select **+ Add data resource**.

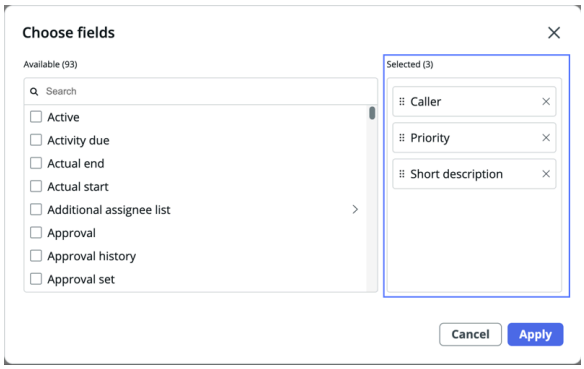
b. Select **Look up multiple records** and then select **Add**.

c. Select the **look_up_multiple_records_1** ID to open the **Data resource details** form.

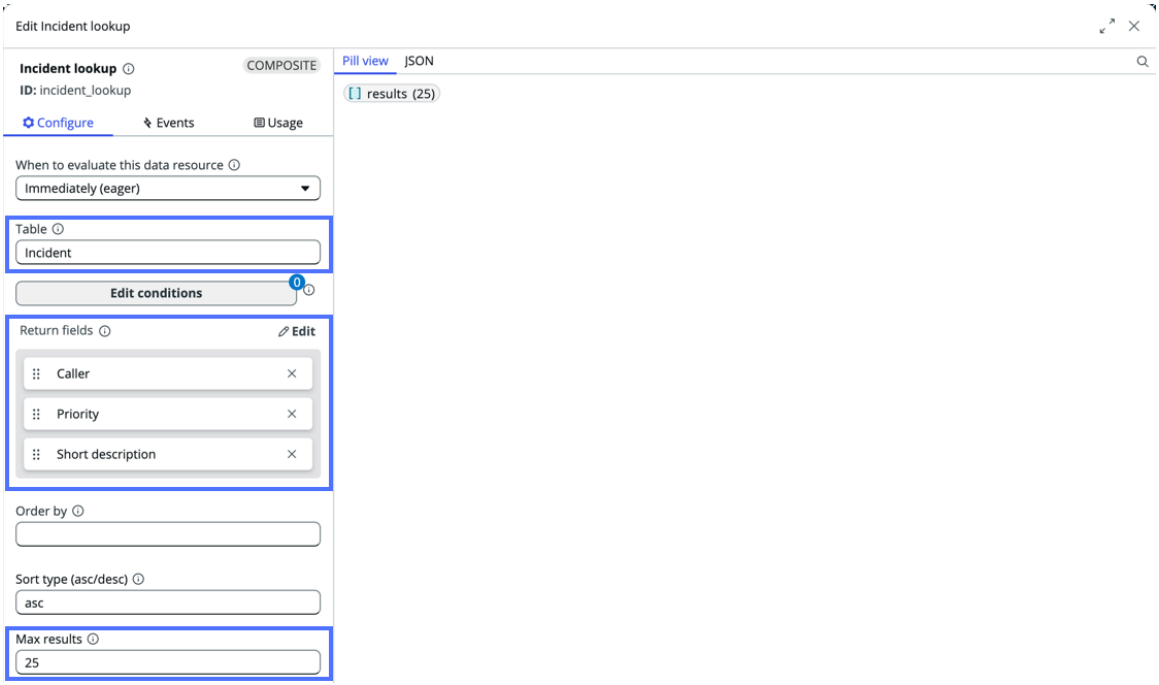
d. Replace the text in the **Data resource label** field with `Incident lookup`.
The **Data resource ID** field auto-populates.

e. Select **Apply**.

f. In the **Edit Incident lookup** dialog, fill in the fields:

Field	Action
Table	Enter `incident`, then select it from the list.
Return fields	i. Select **+ Add**. ii. Enter `Caller`, select it to add it to the **Selected** column, then clear the search box. iii. Repeat the above step for **Priority** and **Short description**.  iv. Select **Apply**.
Max results	Enter 25.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



g. Select the **X** in the upper right to close the dialog.

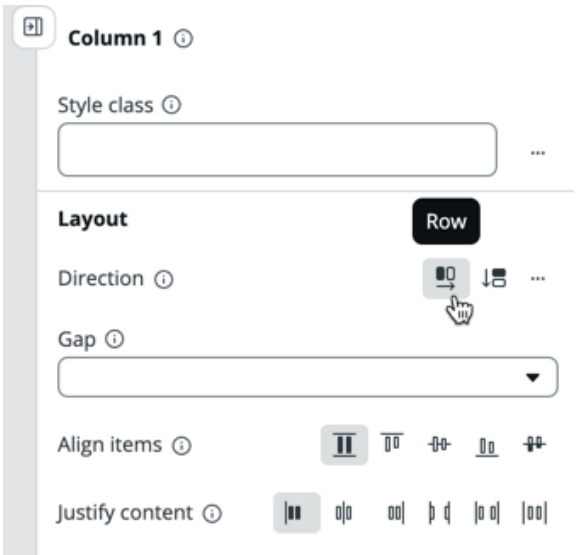
h. Select **Save**.

5. Add a single column layout to hold the components.

a. In the content tree, under **Body**, select **+ Add content**.

b. Select **Single column** and then select **+ Add**.

c. With **Column 1** selected in the content tree, navigate to **Layout** in the configuration panel and set the **Direction** to **Row**.



6. Add and configure a repeater on your page with data binding.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

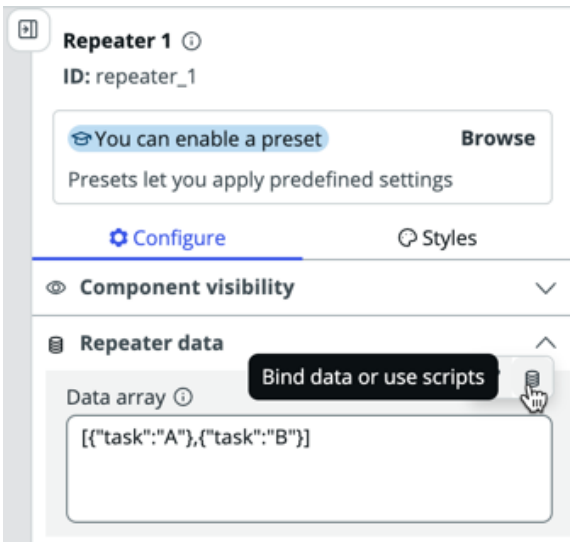
Data binding is the process of associating data exposed by a data resource with a component. In this example, we bind the results of our **Incident lookup** data resource to the **Data array** property of the repeater component. For more information, see Connect data components.

a. In the content tree, under **Column 1**, select **+ Add content**.

b. Enter `Repeater`, then select it in the toolbox and then select **Add**.

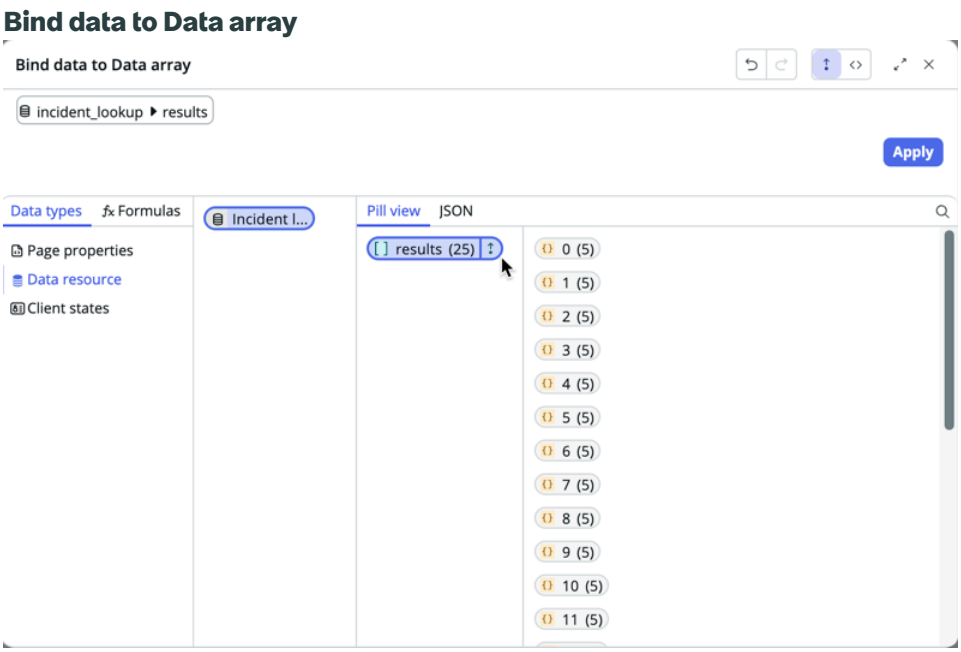
c. Select **Cancel** to close the preset window.

d. Keeping **Repeater 1** selected, in the configuration panel, hover over the **Data array** field and select the bind data icon .



e. In the **Bind data to Data array** dialog, under **Data types**, select **Data resource**.

f. Select the **Incident lookup** pill, then double-click or drag the **results** pill to move it to the area above.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

g. Select **Apply** to confirm the binding.

h. In the configuration panel, select the **Styles** tab, select **Enable styles**, then fill in the fields:

Field	Value
Type	**Grid**
Columns	4
Gap	**S**

i. Select **Save**.

7. Create the card display.

a. In the content tree, under **Repeater 1**, select **+ Add content**.

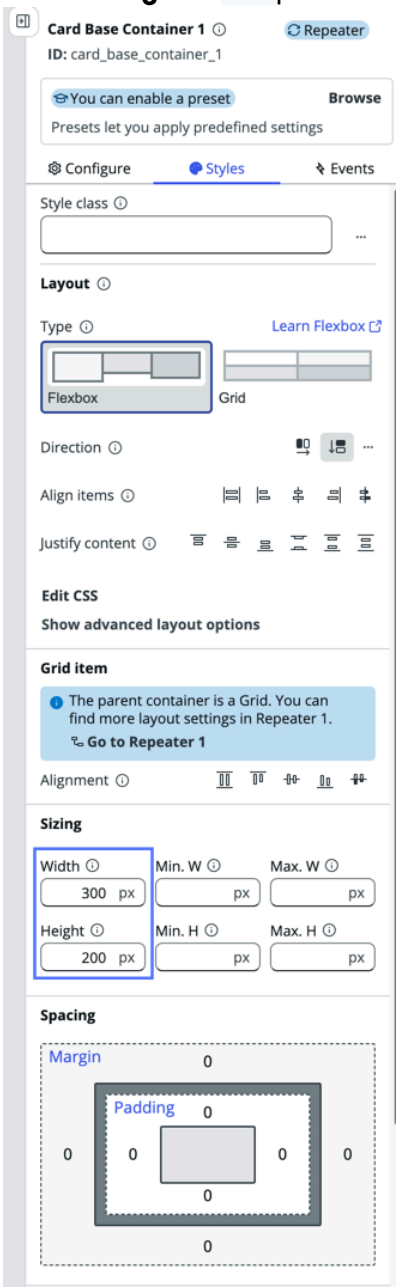
b. Enter `Card Base Container`, select the component, then select **+ Add**.

c. In the configuration panel, select the **Configure** tab and set the **Interaction** to **Click**.

d. In the configuration panel, select the **Styles** tab and set the **Width** to 300px.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

e. Set the **Height** to 200px.



Part 2: Add components and bind data

Add components to each card and configure them to display record details by binding the appropriate fields.

8. Add components to the card base container.

a. In the content tree, under **Card Base Container 1**, select **+ Add content**.

b. Add the following components in the order listed below and edit each in the configuration panel:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Tip: To keep all components nested in **Card Base Container**, add the first component, then use **Add after** from the configure icon ⋮ for each one that follows.

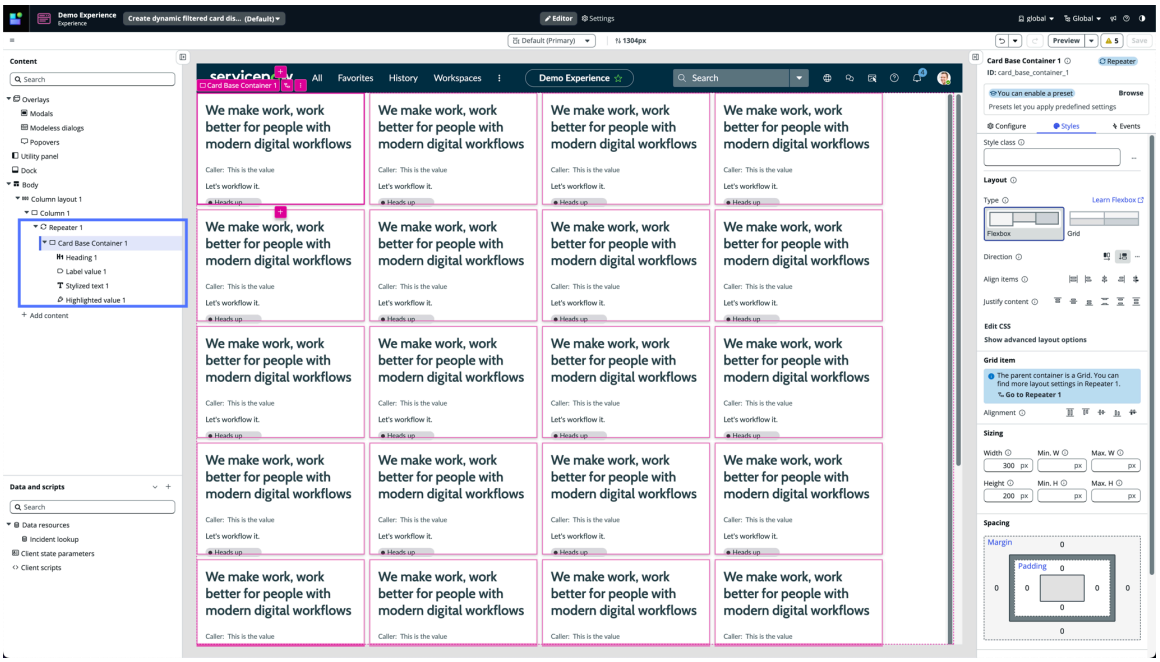
If a preset window appears, select **Cancel** to close it.

i. Heading: No initial configuration required

ii. Label value: In the **Configure** tab, in the **Label** field, enter `Caller:`

iii. Stylized text: In the **Configure** tab, set the **HTML tag** to **Paragraph**

iv. Highlighted value: In the **Styles** tab, set the **Width** to `100`px



c. Select **Save**.

9. Bind data from the incident records to the components we added in the previous step.

Component	Field	Path
Heading	**Label**	`value > number > displayValue`
Label value	**Value**	`value > caller_id > displayValue`
Stylized text	**Text**	`value > short_description > displayValue`
Highlighted value	**Label**	`value > priority > displayValue`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. In the content tree, select the component listed in the table.

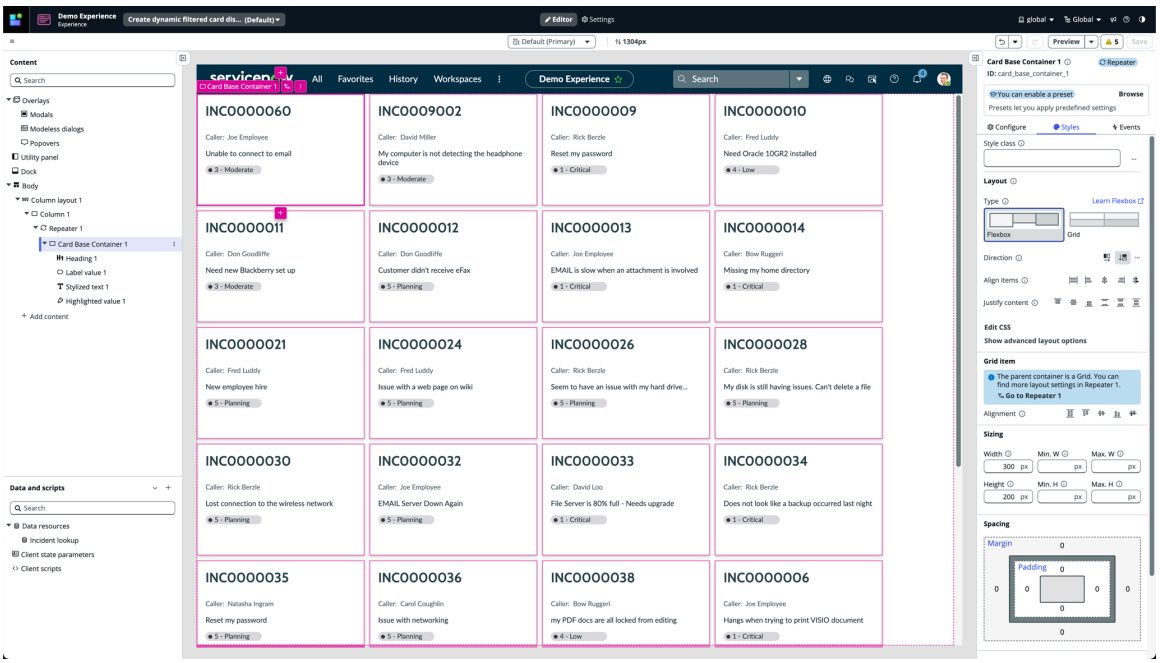
b. In the configuration panel, on the Configure tab, hover over the specified field and select the bind data icon .

c. In the data binding modal:

- On the left, select **Repeater** under **Data types**.
- On the right, under **Pill view**, follow the path listed in the table by selecting each pill in order (for example, for Heading, select **value**, then **number**, then **displayValue**).

d. When you reach the final pill (**displayValue**), double-click or drag it to move it to the area above, then select **Apply**.

e. Repeat the process for each row in the table.



f. Select **Save**.

10. Use a script to define the color of the priority for each record.

a. In the content tree, select **Highlighted value 1**.

b. In the configuration panel, on the Configure tab, hover over **Color** and select the bind data icon .

c. Select the **Use script** icon in the upper right.

d. Replace the existing code with the following:

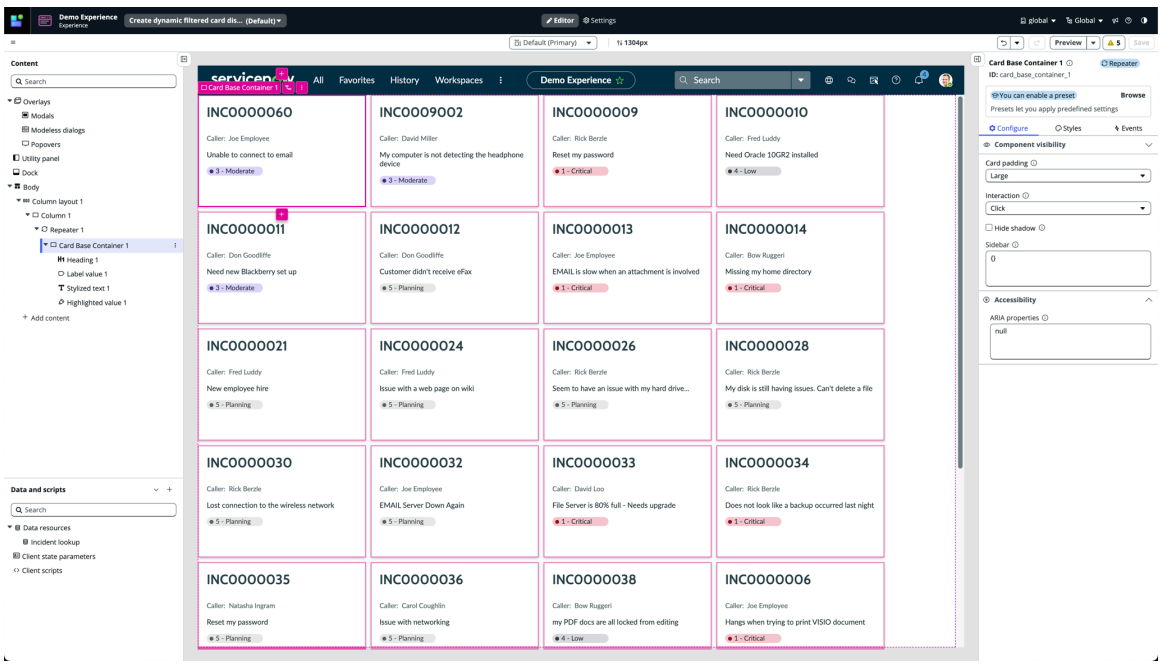
```
function getPriorityColor(record) {
    let priority = record.api.item.value.priority.value;
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    switch (priority) {
        case '1':
            return 'critical';
        case '2':
            return 'high';
        case '3':
            return 'moderate';
        case '4':
            return 'low';
        case '5':
            return 'gray';
        default:
            return 'gray';
    }
}
```

e. Select **Apply**.

Verify that the highlighted value changes color depending on the priority of the record.



Part 3: Add filtering with dropdown

Configure a dropdown event to update a client state parameter and refresh the data resource, filtering the card display.

11. Create a client state parameter to track user selections.
The client state parameter can be modified by components when they trigger events. For more information, see Client state parameters.

a. In the **Data and scripts** drawer, next to **Client state parameters**, select the **+** icon.

b. In the **Edit client state parameters** dialog, fill out the fields:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

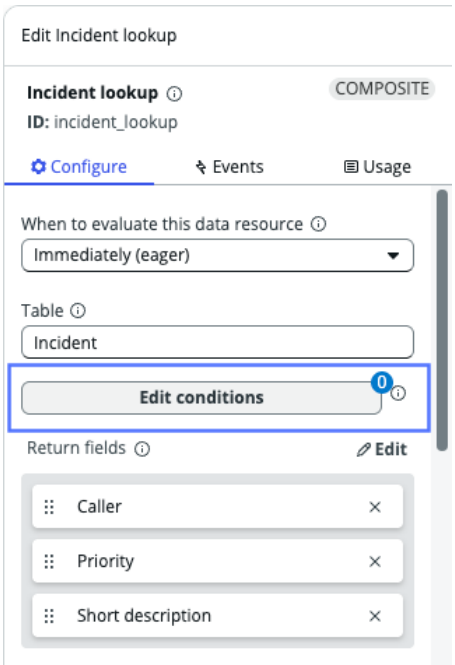
- Name: `selected_caller`
- Type: **String**
- Initial value: empty

c. Select **Apply.**

12. Edit the data resource to use the client state parameter as a conditional value.

a. In the **Data and scripts** drawer, select the **Incident lookup** data resource.

b. Select **Edit conditions**.



c. In the **Conditions** dialog, enter **Caller** in the first field and select it from the list.

d. Leave the second field as is.

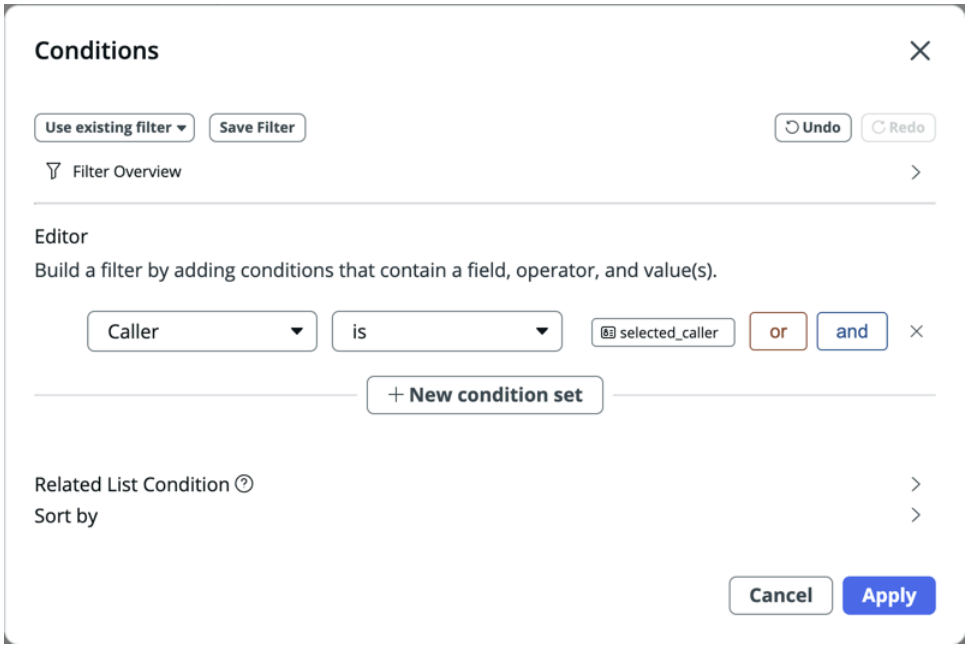
e. Hover over the third field and select the bind data icon .

f. In the data binding modal, select **Client states**, then double-click or drag the `selected_caller` pill to move it to the area above.

g. Select **Apply** to confirm the binding.

h. Select **Apply** to save the condition.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



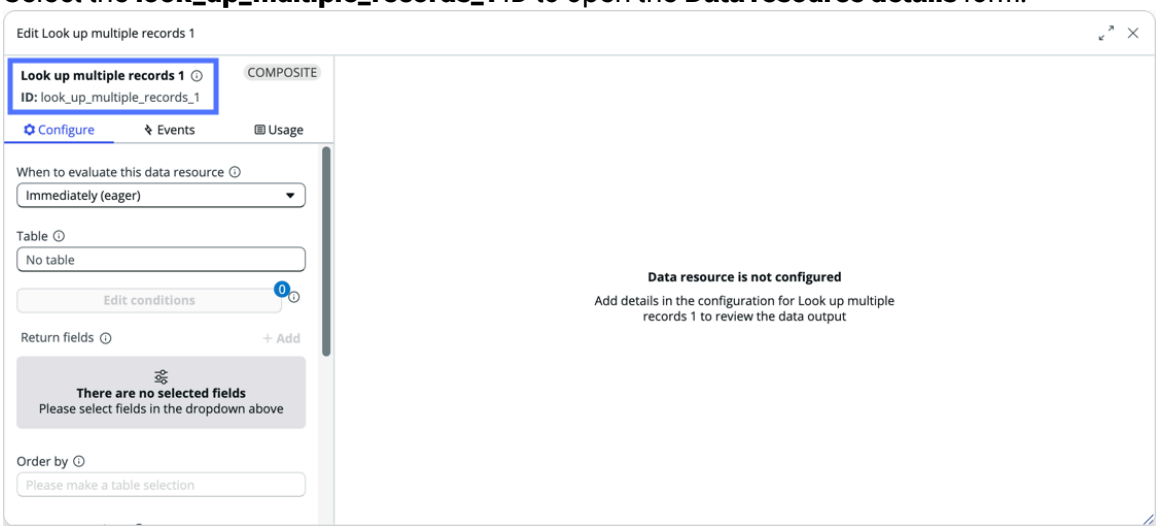
i. Select the **X** in the upper right to close the **Incident lookup** dialog.

13. Create a second data resource to query users from the user table.

a. In the **Data and scripts** drawer, select the **+**, then select **Data resource**.

b. Select **Look up multiple records** and then select **Add**.

c. Select the **look_up_multiple_records_1** ID to open the **Data resource details** form.



d. Replace the text in the **Data resource label** field with `sys_user_lookup`.
The **Data resource ID** field auto-populates.

e. Select **Apply**.

f. In the **Edit sys_user_lookup** dialog, fill in the fields:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Action
Table	Enter `User (sys_user)` and then select it from the list.
Return fields	i. Select **+ Add**. ii. Enter `Name`, then select it from the list to add it to the **Selected** column. iii. Select **Apply**.
Order by	Enter `Name`, then select it from the list.
Max results	Enter `250`.

g. Select the **X** in the upper right to close the dialog.

h. Select **Save**.

This data resource looks up users from the user table. We will reference this data to populate the dropdown in the next step.

14. Add and configure a dropdown component.

a. In the content tree, hover over **Column layout 1** and select the **Configure** icon .

b. Select **Add before**.

c. Search for **Dropdown**, select it from the toolbox and then select **Add**.

d. In the configuration panel, navigate to the specified tab and configure the following fields:

Tab	Action
Configure	In the **Placeholder text** field: Enter `Select a caller`.
Configure	In the **Variant** field: Select **Primary**.
Configure	In the **List items** field: i. Hover over the field and select the bind data icon . ii. In the data binding modal, select the **Use script** icon . iii. Replace the existing code with the following: **Tip:** You can select the format code icon to make the code more readable. `function evaluateProperty({ api, helpers }) { const userArray = api.data.sys_user_lookup.results; let outputArray = [];`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Tab	Action
	`` for (let i = 0; i < userArray.length; i++) { const obj = { 'id': `${userArray[i]._row_data.uniqueValue}`, 'label': `${userArray[i].name.displayValue}` }; outputArray.push(obj); } return outputArray; } ``` **iv.** Select **Apply**.
Styles	Width: Enter 200.
Styles	Padding: Select **Padding**, then choose **M** from the list.

15. Configure the dropdown with events so that it can modify the client state parameter we created.

a. With **Dropdown 1** selected in the content tree, select the **Events** tab in the configuration panel, then select **Add event mapping**.

b. Select the **Selected items changed** event, then select **Continue**.

c. Select the **Set client state parameter** handler, then select **Continue**.

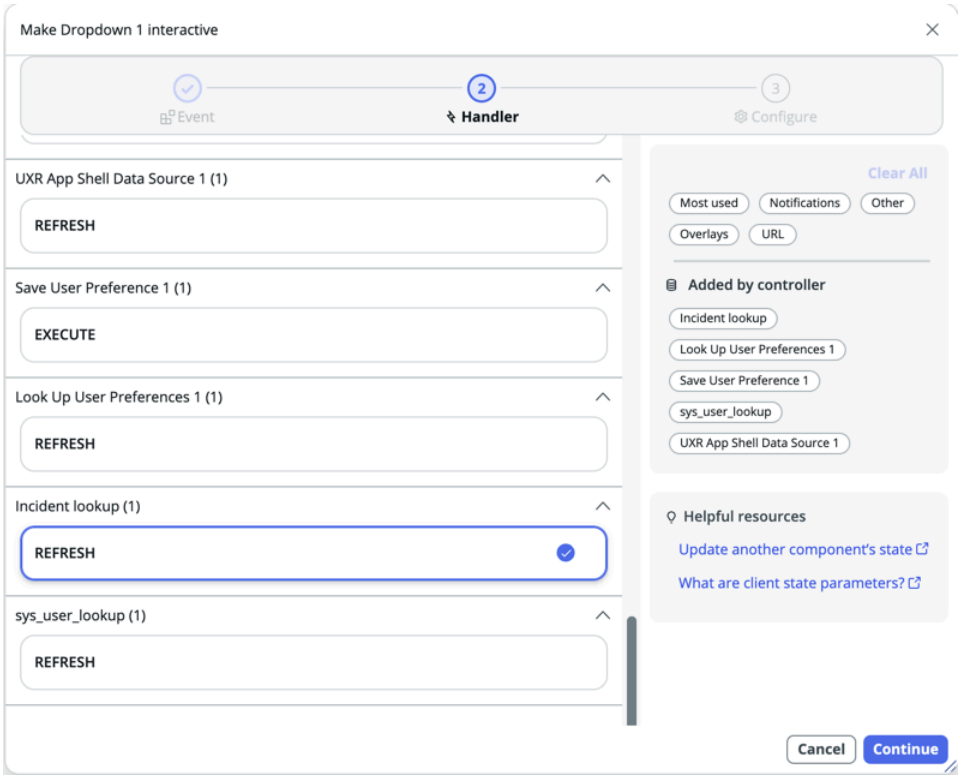
d. Set the following properties:

Field	Action
Client State Parameter Name	Select **selected_caller** from the list.
Value to use after triggering event	**i.** Hover over the field and select the bind data icon . **ii.** Select **Data types**, then **Event payload**. **iii.** Under **Pill view**, double-click or drag the **value** pill to move it to the area above. **iv.** Select **Apply** to confirm the binding. **v.** Select **Add**.

e. In the **Events** tab in the configuration panel, select **Add handler** underneath the event we added in the previous step.

f. Locate **Incident lookup (1)**, select the **REFRESH** handler, then select **Continue**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



g. Select **Add**.

h. Select **Save**.

After configuring the dropdown, selecting a caller updates the `selected_caller` client state parameter and refreshes the data resource to display that caller's records.

Part 4: Enable record navigation

Configure each card so that selecting it opens the record page using the sys_id from the repeater data.

16. Create another page in your experience that opens the record.

a. Return to the experience view by selecting the name of your experience (for example, "Demo Experience") in the upper left.

b. Select the **+** next to **Pages**, then select **Create a new page**.

c. Hover over the **Standard record** template and select **Use template**.

d. Name your page `Incident record page` and select **Continue**.

e. Select **Looks Good**, then select **Create**.

17. Return to your original page by selecting the name of your experience in the upper left, then choosing your page from the experience view.

18. Add an event to the cards so that selecting a card opens its record on the page we created.

a. Select **Card Base Container 1** in the content tree.

b. In the configuration panel, select the **Events** tab, then select **Add event mapping**.

c. Select the **Card clicked** event, then select **Continue**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

d. Select the **Open page or URL** handler, then select **Continue**.

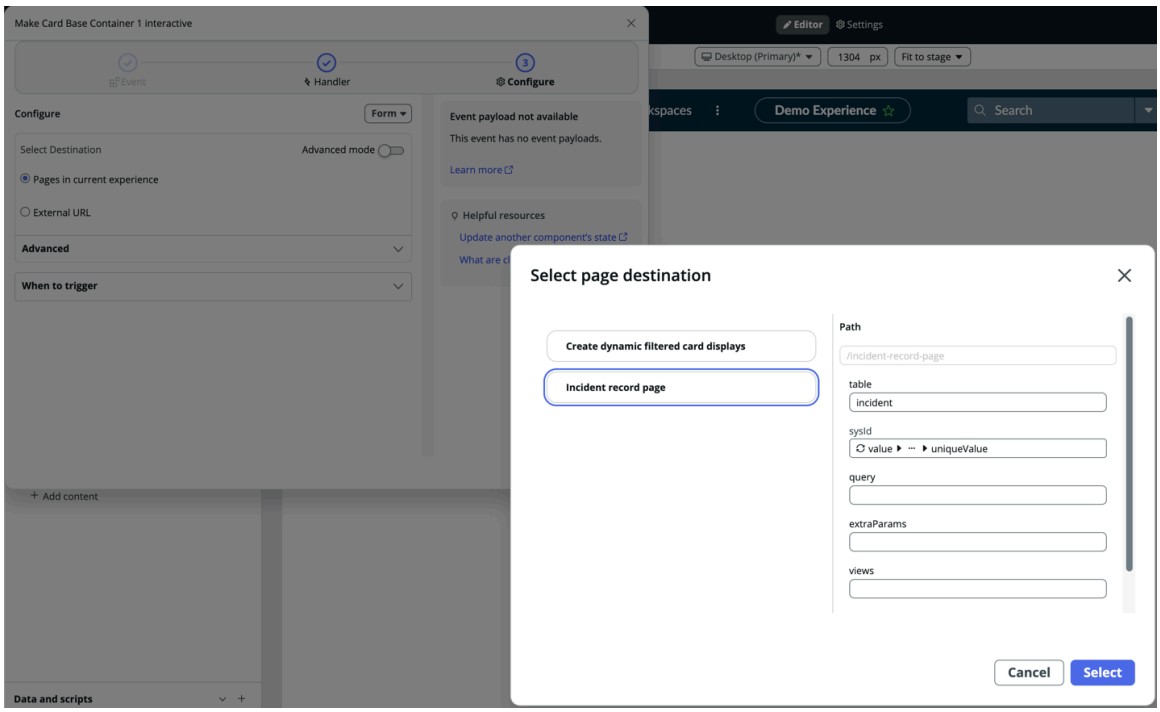
e. Select **Pages in current experience**.

f. Select **Incident record page**.

g. In the form, fill in the fields:

Field	Action
table	Enter `incident`
sysId	**i.** Hover over the **sysId** field and select the bind data icon . **ii.** Locate **Data types** on the left and select **Repeater**. **iii.** In the **Pill view**, select **value**, then **_row_data**, then double-click **uniqueValue**. **iv.** Select **Apply**.

h. Select **Select**.



i. Select the check box to enable the **Open in new tab** property.

j. Select **Add**.

k. Select **Save**.

19. Preview and test the page.

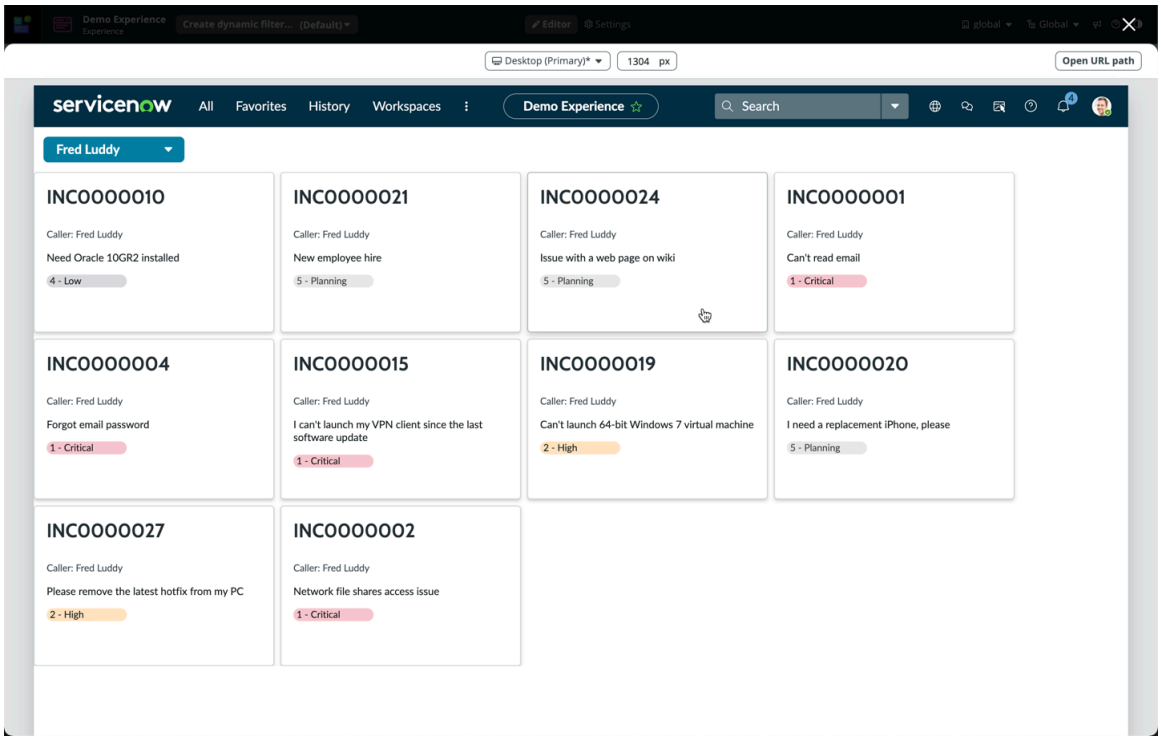
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. Select **Preview**.

b. In the dropdown, select a user such as "Fred Luddy," "Carol Coughlin," or "Bow Ruggeri" to filter the cards.

> **Note:** Some users do not have related incident records, so selecting those users will result in no cards being displayed.

c. Select a card to view the full record in a new tab.



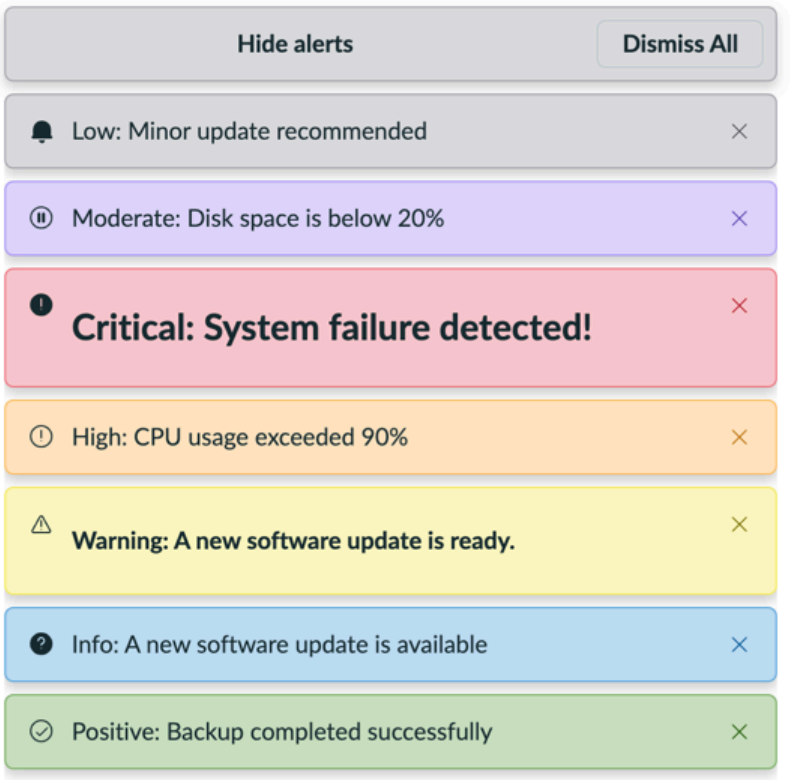
Create and customize alert messages in UI Builder

Learn how alert messages help you communicate feedback and status updates using both default and scripted approaches.

Alerts are components that display standardized notifications, such as feedback, warnings, and confirmations. They're easy to configure without scripting, but you can add a script for more advanced functionality.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Alert message types



UI Builder supports several types of alert messages. To see how each one behaves, open the Alert component documentation and try out different settings in the interactive preview.

Create alert messages in UI Builder

Add and configure alert messages for simple notifications without scripting.

Before you begin

Role required: ui_builder_admin

About this task

Use the default alert message configuration for simple notifications. In this example, the alert displays a personalized greeting using the logged-in user's full name.

This procedure uses UI Builder components to create dynamic, interactive layouts. For more information on how to configure components, see:

- Add and configure components
- UI Builder Quick Bits: Navigating Component Configuration

Components used in this procedure

Component	Documentation links
Alert	• Usage Guidelines • UIB Setup

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

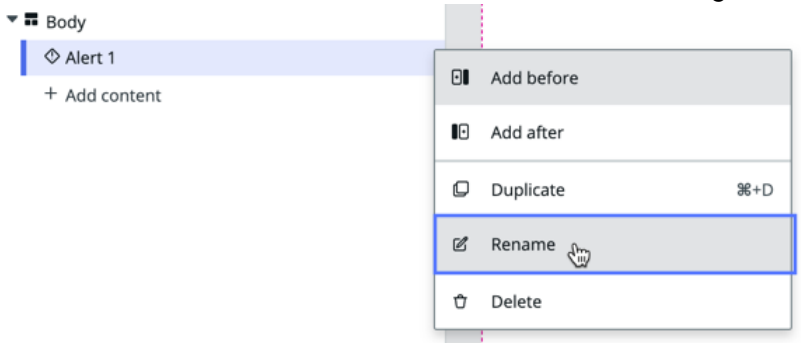
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create a page from scratch.
   For more information about how to create a page, see Create a page in UI Builder.
4. Add an alert.

   a. In the content tree, select **+ Add content**.

   b. Search for `alert` and then select **Add** to place the component on your page.

   c. In the content tree, hover over **Alert 1** and select the configure icon ⋮, then select **Rename**.



   d. Replace the text in the **Component label** field with `Welcome Message`.
   The **Component ID** field auto-populates.

   e. Select **Apply.**

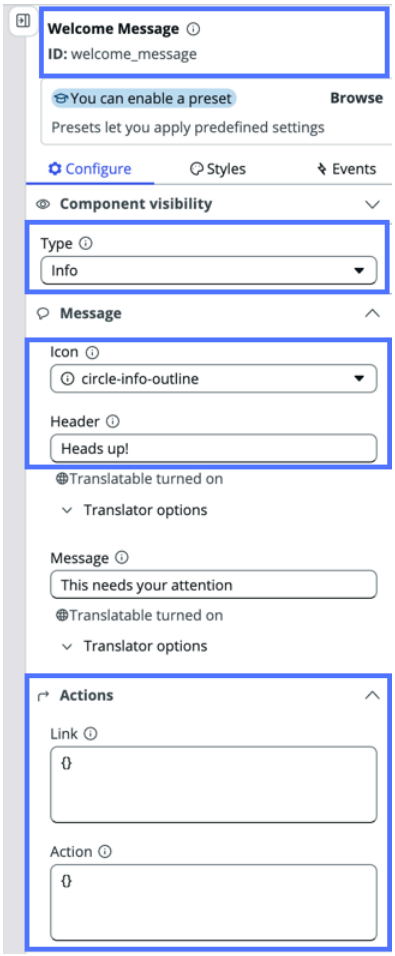
5. Configure the alert message.

   a. With **Welcome Message** selected in the content tree, navigate to the configuration panel and set the following properties:

Field	Value or Action
Type	**Info**
Icon	**circle-info-outline**
Header	**Heads up!**
Link	(Leave blank) — select **Edit**, then delete any text inside **Label** and **Href**
Action	(Leave blank) — select **Edit**, set **Type** to **-- None --**, then delete any text inside **Label** and **Href**

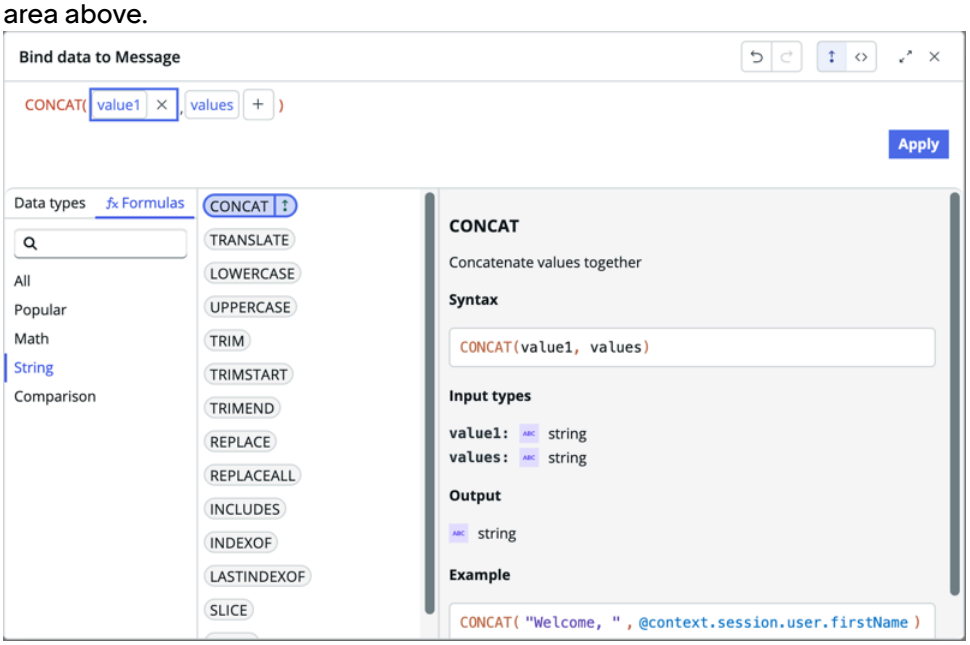
   Your configuration panel should look like this:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



b. Hover over the **Message** field and select the bind data icon .

c. Select **Formulas**, then **String**, then double-click or drag **CONCAT** to move the formula to the area above.



d. Double-click **value1** to select the field, then select again to insert text.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

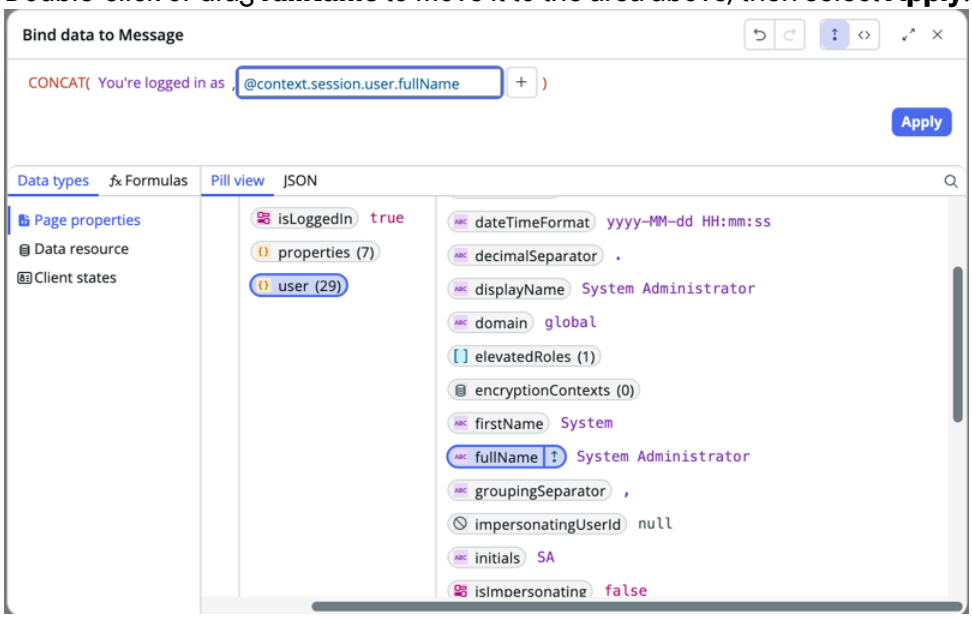
e. Enter `"You're loggeded in as "`, making sure to include a trailing space after the exclamation point.

f. Double-click **values** to select the field, then select again to insert text.

g. Select **Data types**, then **Page properties**.

h. Under **Pill view**, select the pills in the following order: **session** > **user** > **fullName**.

i. Double-click or drag **fullName** to move it to the area above, then select **Apply**.

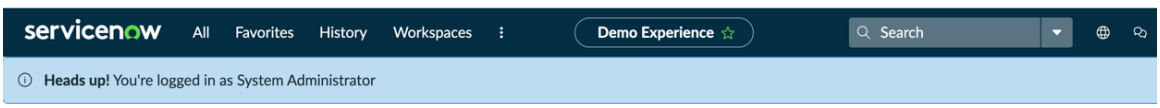


6. Save and test your page.

a. Select **Save** in the upper-right.

b. Select **Preview**.

The alert appears at the top of the page with the text "Heads up! You're logged in as " followed by the logged-in user's name.



Customize alert messages with a client script in UI Builder

Use a client script to create dynamic, context-sensitive alert messages.

Before you begin

Role required: admin

About this task

Scripted alerts provide notifications that respond to events or external data. They are highly flexible and can react to complex conditions, but require scripting knowledge and more effort to maintain. In this example, a button triggers a client script that generates multiple alerts with custom messages.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

This procedure uses UI Builder components to create dynamic, interactive layouts. For more information on how to configure components, see:

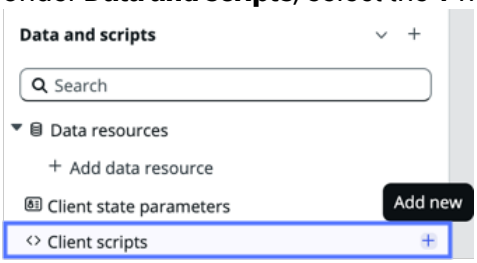
- Add and configure components
- UI Builder Quick Bits: Navigating Component Configuration

Components used in this procedure

Component	Documentation links
Button	• Usage Guidelines • UIB Setup
Alert	• Usage Guidelines • UIB Setup

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create a page from scratch.
   For more information about how to create a page, see Create a page in UI Builder.
4. Add a button.
   a. In the content tree, select **+ Add content**.
   b. Search for `button` and add it from the toolbox.
   c. In the content tree, hover over **Button 1** and select the configure icon , then select **Rename**.
   d. Replace the text in the **Component label** field with `Showcase Alerts`.
   The **Component ID** field auto-populates.
   e. Select **Apply**.
5. Create a client script.
   a. Under **Data and scripts**, select the **+** next to **Client scripts**.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

b. Replace the text in **Script name** with `Alerts`.

c. Replace the code with the following:

Tip: You can select the format code icon to make the code more readable.

```
/**
* @param {params} params
* @param {api} params.api
* @param {any} params.event
* @param {any} params.imports
* @param {ApiHelpers} params.helpers
*/
function handler({
    api,
    event,
    helpers,
    imports
}) {
    api.emit("NOW_UXF_PAGE#ADD_NOTIFICATIONS", {
        items: [
            {
                id: "alert1",
                status: "critical",
                icon: "circle-exclamation-fill",
                content: {
                    type: "html",
                    value: "<h2>Critical: System failure
 detected!</h2>"
                },
                action: { type: "dismiss" }
            },
            {
                id: "alert2",
                status: "high",
                icon: "circle-exclamation-outline",
                content: {
                    type: "string",
                    value: "High: CPU usage exceeded 90%"
                },
                action: { type: "dismiss" }
            },
            {
                id: "alert3",
                status: "moderate",
                icon: "circle-pause-outline",
                content: {
                    type: "string",
                    value: "Moderate: Disk space is below
 20%"
                },
                action: { type: "dismiss" }
            },
            {
                id: "alert4",
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
                status: "warning",
                icon: "triangle-exclamation-outline",
                content: {
                    type: "html",
                    value: "<h4> Warning: A new software
 update is ready. </h4>"
                },
                action: { type: "dismiss" }
            },
            {
                id: "alert5",
                status: "info",
                icon: "circle-question-fill",
                content: {
                    type: "string",
                    value: "Info: A new software update is
 available"
                },
                action: { type: "dismiss" }
            },
            {
                id: "alert6",
                status: "positive",
                icon: "check-circle-outline",
                content: {
                    type: "string",
                    value: "Positive: Backup completed
 successfully"
                },
                action: { type: "dismiss" }
            },
            {
                id: "alert7",
                status: "low",
                icon: "bell-fill",
                content: {
                    type: "string",
                    value: "Low: Minor update recommended"
                },
                action: { type: "dismiss" }
            }
        ]
    });
}
```

d. Select **Apply**.

6. Link the button to the client script with an event.

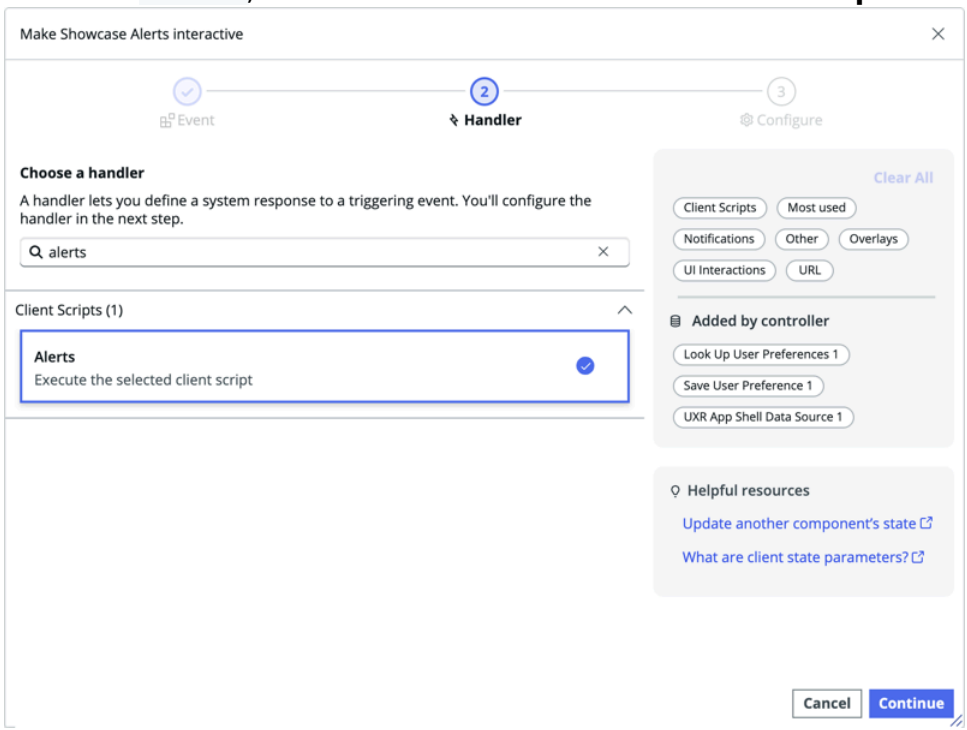
a. In the content tree, select the button we created: **Showcase Alerts**.

b. In the configuration panel, select the **Events** tab.

c. Under the **Button clicked** event, select **Add handler**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

d. Search for `alerts`, then select the **Alerts** handler under **Client Scripts**.



e. Select **Continue**.

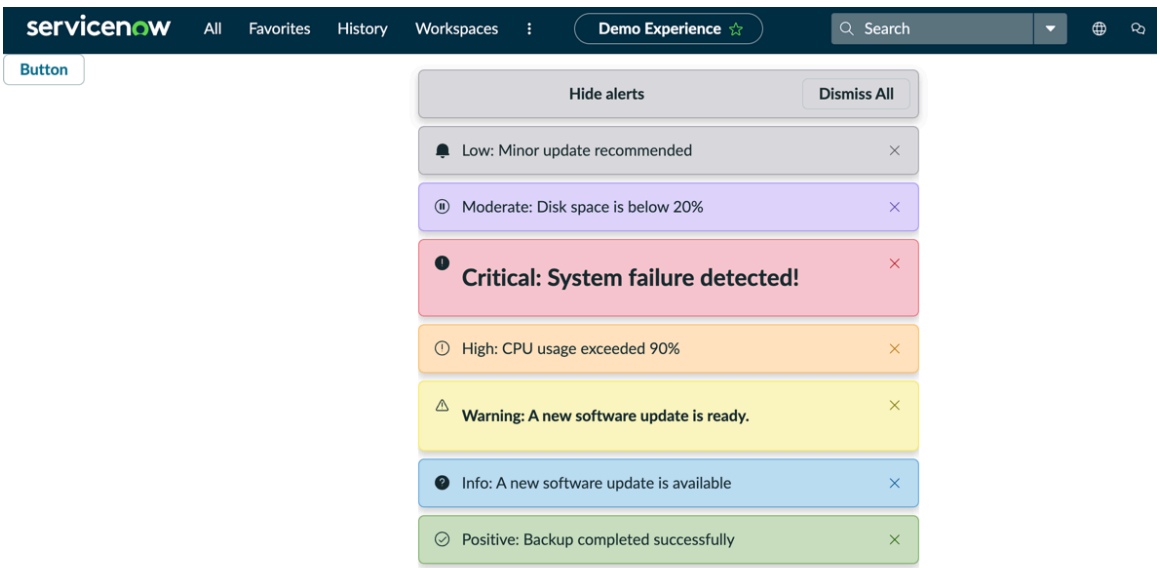
f. Select **Add**.

7. Save and test your page.

a. Select **Save** in the upper-right.

b. Select **Preview**.

Selecting the button displays multiple alerts with messages that you define in the client script.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Filter records by priority with a flyout menu

Configure a flyout menu that filters a task record list based on priority.

Before you begin

Role required: ui_builder_admin

This procedure uses UI Builder components to create dynamic, interactive layouts. For more information on how to configure components, see:

- Add and configure components
- UI Builder Quick Bits: Navigating Component Configuration

Components used in this procedure

Component	Documentation links
Flyout menu	• Usage Guidelines • UIB Setup
Record list	Usage Guidelines

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create a page from scratch.
   For more information about how to create a page, see Create a page in UI Builder.
4. Select **Add content** on the stage or in the content tree to open the toolbox, then add a **Record list** component to the page.
5. In the **Data and scripts** panel, select the **List Controller** data resource that was auto-created with the record list component.
   a. Set **Table** to Task.
   b. Set **Record count limit** to `25`.
   c. Under **Returned fields**, add the fields you want to display in the list (for this example, add **Number**, **Short description**, **Priority**, and **State**).
6. In the **Data and scripts** panel, select **Client state parameters** and then select **Add**.
7. Configure the client state parameter:
   a. Set **Name** to `priorityFilter`.
   b. Set **Type** to **JSON**.
   c. Set **Initial value** to `["1","2","3","4","5"]` so all task records display on page load.
8. Select the **List Controller** data resource again, then select **Edit fixed filter**.
9. Configure the filter condition:
   a. Set the first field to `Priority`.
   b. Set the operator to `is one of`.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

c. Hover over the third field and select the bind data icon .

d. In the data binding modal, select **Client states**, then double-click or drag the `priorityFilter` pill to move it to the area above.

e. Select **Apply** to confirm the binding.

f. Select **Apply** to save the condition.

10. Add and configure a Flyout menu component.

a. In the content tree, hover over **Record list** and select the **Configure** icon .

b. Select **Add before**.

c. Search for **Flyout menu**, select it from the toolbox and then select **Add**.

d. In the configuration panel, hover over **List items**, then select **Edit**.

e. Set the **Mode** to **JSON** and replace the existing code with the following array:

```
[
  {"id": "1", "label": "1 - Critical"},
  {"id": "2", "label": "2 - High"},
  {"id": "3", "label": "3 - Moderate"},
  {"id": "4", "label": "4 - Low"},
  {"id": "5", "label": "5 - Planning"}
]
```

f. Select **Apply**.

11. In the content tree, locate the **trigger** slot under **Flyout menu** and select **Add content**.

12. Add a button from the toolbox.
This button will open the flyout menu.

13. Configure the flyout menu to update the list and refresh the data resource when the user selects an item.

a. In the configuration panel, select the **Events** tab.

b. Under **Flyout menu selected items set**, select **Add handler**.

c. Select the **Set client state parameter** handler, then select **Continue**.

d. Set the following properties:

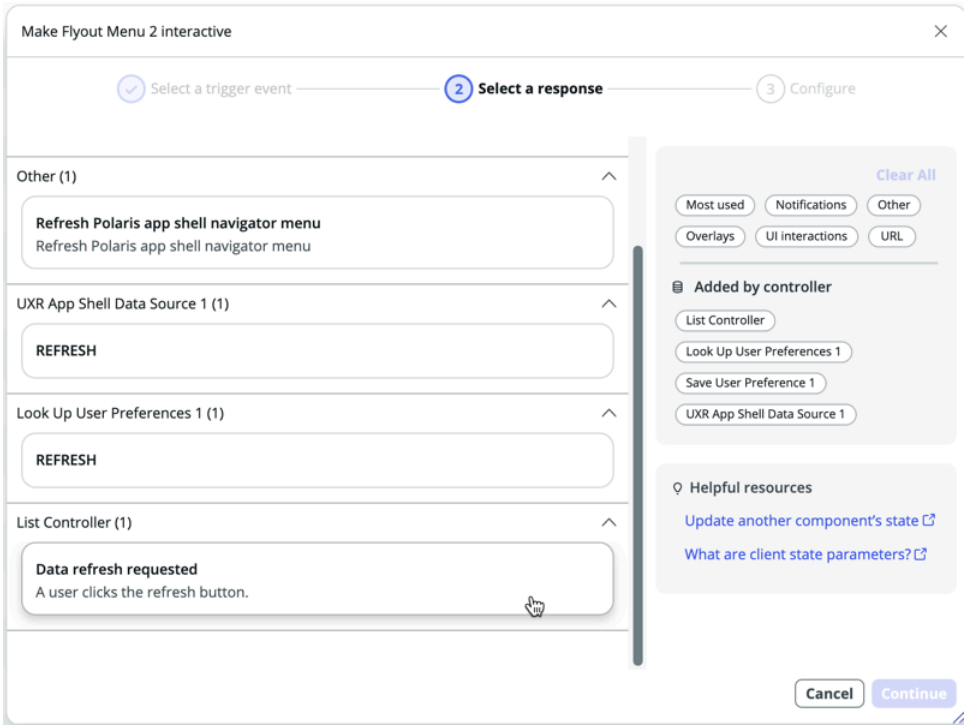
Field	Action
Client State Parameter Name	Select **priorityFilter** from the list.
Value to use after triggering event	i. Hover over the field and select the bind data icon . ii. Select **Data types**, then **Event payload**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Action
	iii. Under **Pill view**, double-click or drag the **value** pill to move it to the area above. **iv.** Select **Apply** to confirm the binding. **v.** Select **Add**.

e. In the **Events** tab in the configuration panel, select **Add handler** underneath the event we added in the previous step.

f. Locate **List Controller**, select the **REFRESH** handler, then select **Continue**.

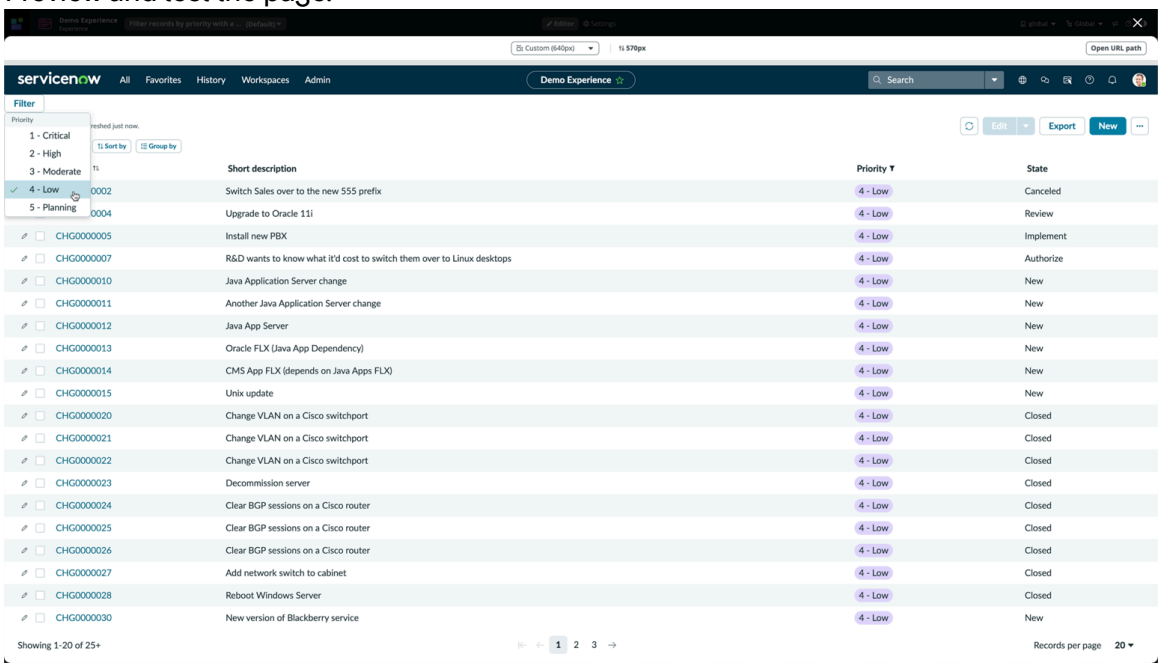


g. Select **Add**.

h. Select **Save**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

14. Preview and test the page.



Page collections

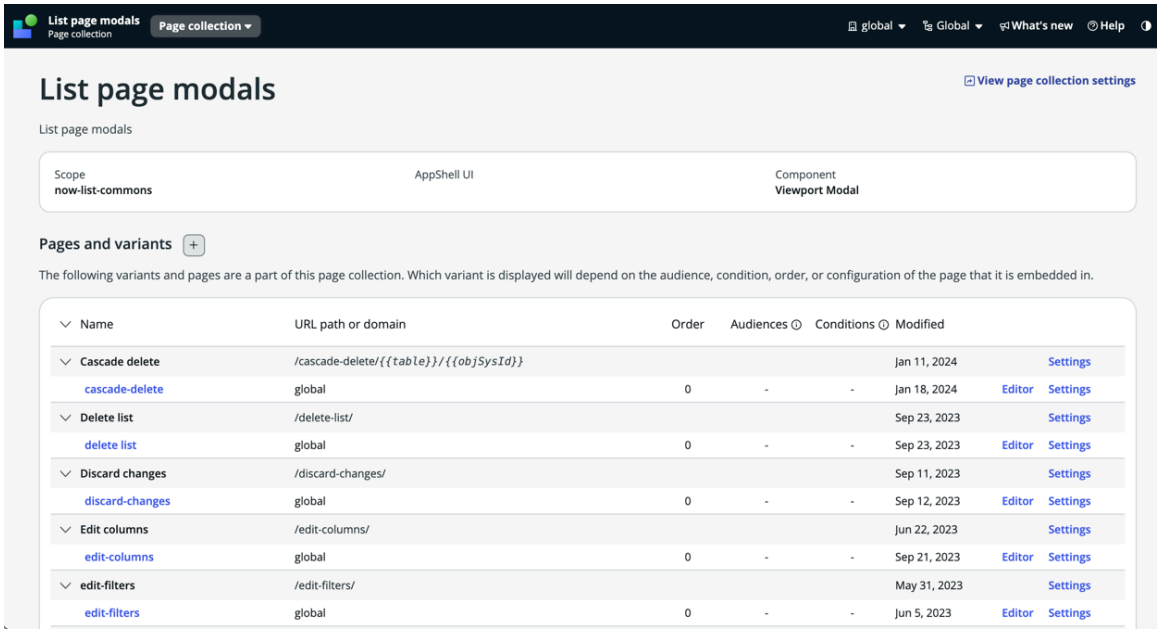
Page Collections are groups of pages that can be used across multiple experiences.

What is a page collection

Page Collections (sys_ux_extension_point) are groups of pages that are meant to be re-used across multiple experiences. These sets of pages can be used in a modal viewport or in a tabs component. These pages are sandboxed and do not have access to things like URL params or data resources from the page that is calling them. All they have access to is the Controller record that is specified on the Page Collection record and that is passed down by the calling page in runtime.

Page collections are sandboxed inside of the pages they are within, this means that they do not have direct access to change the parent page. However, because of the way seismic works, your page collection page can dispatch an event that the calling page can listen for to take action. This is the only way to achieve a change in the calling page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Why should you use page collections

Page collections allow you to reuse custom pages across experiences. This will save you time if you need to replicate a page for multiple experiences and help pass information to different groups of users.

You can also use page collections to provide additional information within an existing topic by using a viewport modal. A viewport modal will allow your audience to view pages outside of their experience to make changes without leaving the parent page.

Create collection of tabs that can be shared across experiences. Each tab is an individual page in a page collection.

Where can you use page collections

You can access a page collection from the UI Builder home screen. The list of page collections details which component and type each page collection is.

You can create a page collection from the UI Builder home screen, or from a component that uses page collections.

Page collections can be used with the following components:

- Viewports
- Viewport Modals
- Tabs

Things to consider

The page collection creation screen requires a component and controller selection.

Data resources need to be selected to pass information to a page collection. Need to bind data from the parent page.

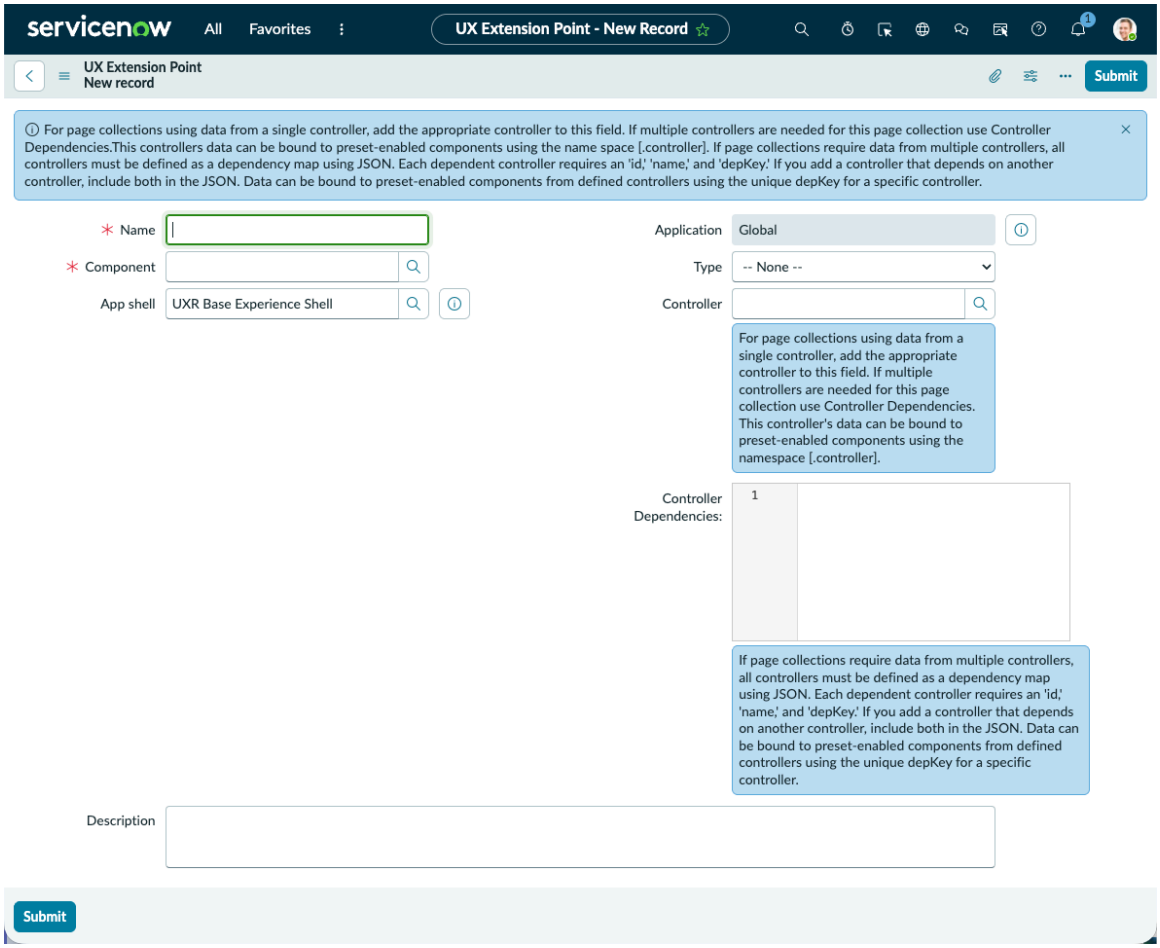
Need to route to the page in a page collection you want to display in a viewport.

Add a link to creating a page collection.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Page collection on the platform

You create a page collection within the ServiceNow platform. At this time, only the Tab and Viewport component can be used when creating a page collection. You can select a controller or pass controller dependencies using JSON.



Create a page collection across multiple UI pages

Create a page collection to accommodate tabbed content that can be used across experiences.

Before you begin

Role required: ui_builder_admin

About this task

Use page collections to create tabbed content in your experiences with the Tabs component in UI Builder.

Pages within a page collection don’t have access to the parent page's URL parameters or data resources. You can set conditions for pages in a page collection to define which page to display to certain audiences. Page templates are not supported inside a page collection.

Procedure

1. Click **Create** > **Page collection**.
2. On the form, fill in the fields.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Create a page collection form

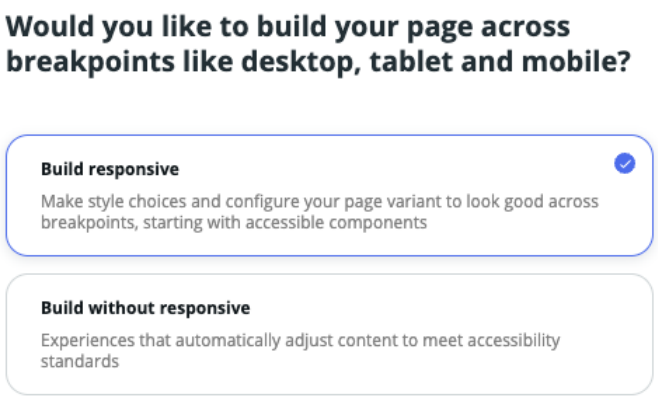
Field	Description
Name	Name to track your page collection internally.
App shell UI	Type of app shell UI that you want to use with the page collection. The app shell is the wrapper of the page contents, which is similar to the functionality of a web page. The app shell can show things like the logo of your company, user preferences, and the search icon. For more information, see Define UI experiences using app shells.
Component	Select the component that will use the page collection. Page collections can be used with the following components: ○ Viewports ○ Viewport Modals ○ Tabs
Description	Short description to help find your page collection. Write a description that helps page builders understand what content is included in the page collection.

3. Click **Next**.
4. Select a controller to connect data to your page collection.
   If the controller you selected depends on another controller, you will need to add both.
5. Enter a label in the **Label** field.
6. Enter a unique Id in the **Id** field.
7. Click **Create**.
8. Click **Create new page**
9. Enter a unique name for the page in the **Name** field.
10. Specify a path for your page in the **Path** field. UI Builder generates a default path based on the name that you provided in the previous step.

    Note: The application scope defaults to the scope that the user is in. For more information about the application scope, see Learn about security and roles.

11. Click **Continue**.
12. **Optional:** Add one or more audiences for this page.
    If an audience you need is not listed, you can choose the Open audiences in platform link to create one.
13. **Optional:** Add one or more conditions for this page.
14. Select the **Build responsive** option (default) for greater control of how the page appears at different screen form factors or select **Build without responsive** for automatic adjustment.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



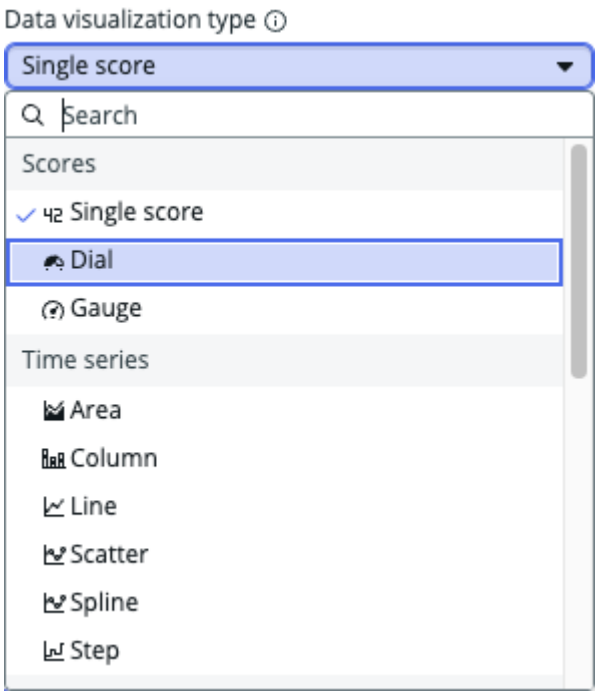
For more information, see Responsive authoring.

15. **Optional:** Add more pages to your page collection.

Change data visualizations in UI Builder

Change data visualizations in real time using a drop-down list to preview data in your experience.

You can change data visualization types via a drop-down in the configuration panel to preview different data visualization types. Each data visualization can be configured to display different data depending on who is using your experience. Select the drop-down under **Data visualization type** to preview different types of data visualization.



A preview of the data visualization you selected appears in your page. For information about adding data visualization components to your experience, see Add and configure components.

Add tabbed content to UI Builder pages

Use the Tabs component to add tabbed content to pages in UI Builder.

Create additional navigation on your UI Builder page by adding a Tabs component to your page. Use the Tabs component to nest components within tabs on your page. Tabs can be configured

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

as either horizontal tabs at the top of the content, or as vertical tabs on the left or right side of the content.

Tabs can be organized in the configuration panel to reorder how they are displayed on the UI Builder page. Add custom labels and icons to tabs to provide unique visual identifiers for navigating across tabs. These changes can be viewed in real-time as you edit the Tabs component.

You can add one of the following types of tabs to your UI Builder page:

- Empty container tab
- Repeater tab
- Related list tab
- Page collection tabs

The tabs component supports any combination of static tabs, repeated tabs, related list tabs, and page collection tabs. Alternatively, you can still use the Viewport tab mode to add a series of viewports.

Add empty container tabs

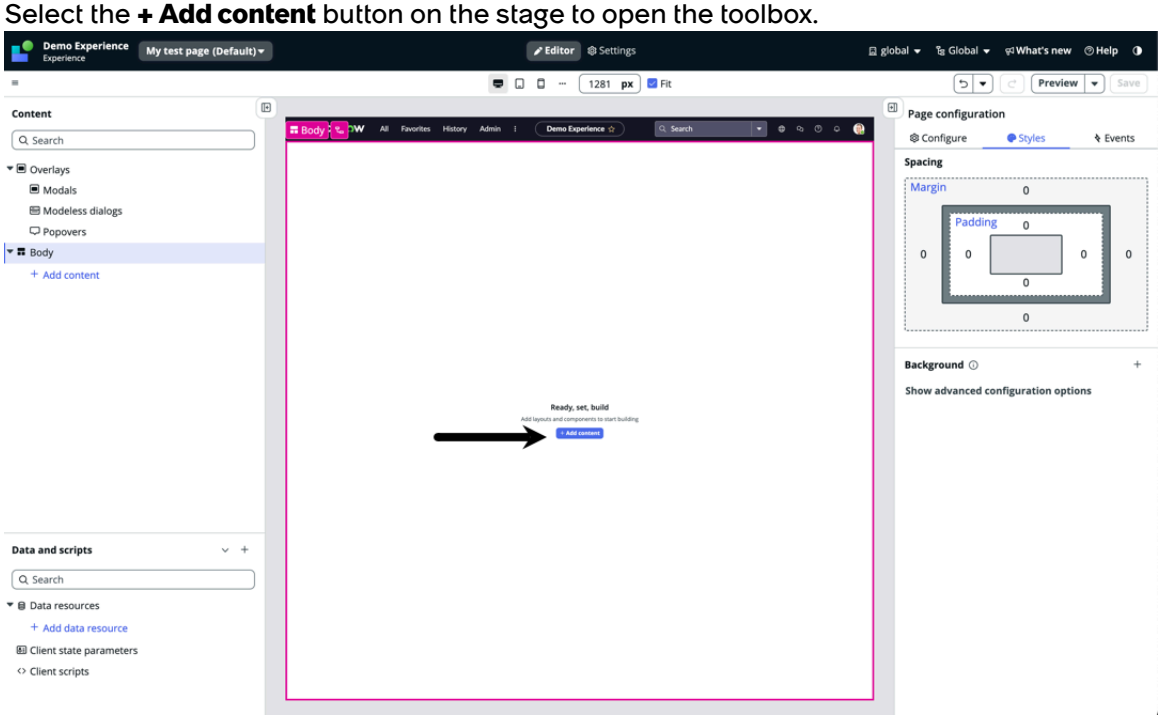
Use an empty container tab to manually create a static set of tabs on your page.

Before you begin

Role required: admin

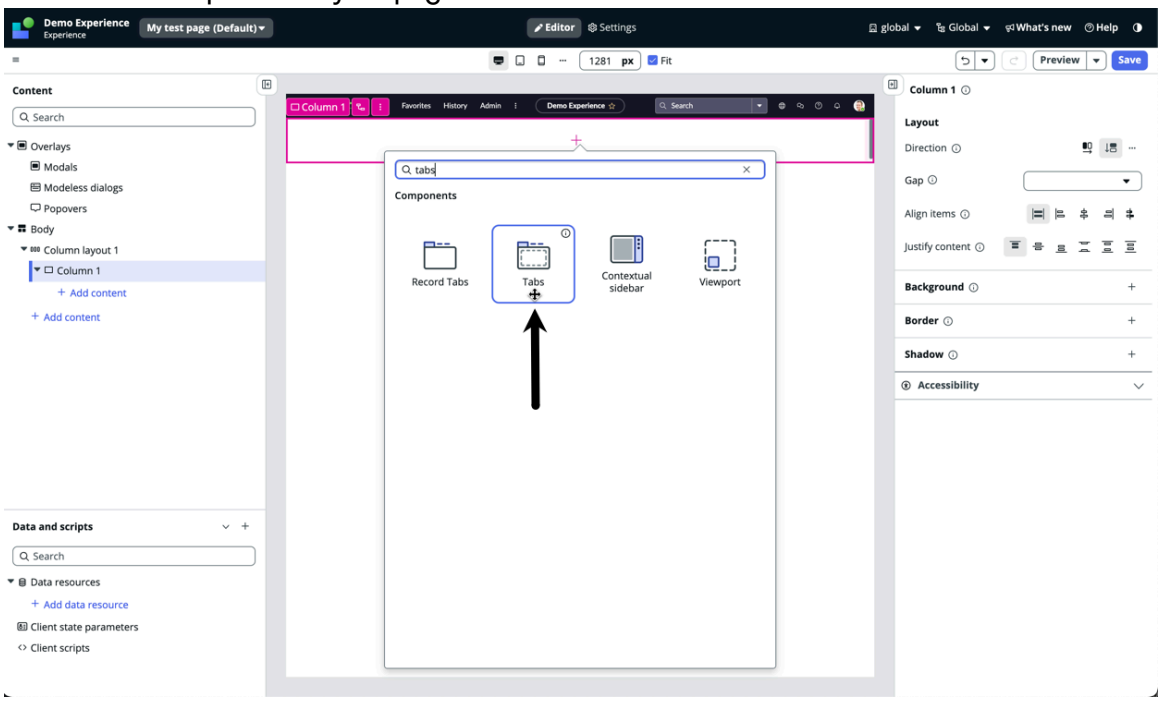
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create or open a page or page variant.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. Select the **+ Add content** button on the stage to open the toolbox.



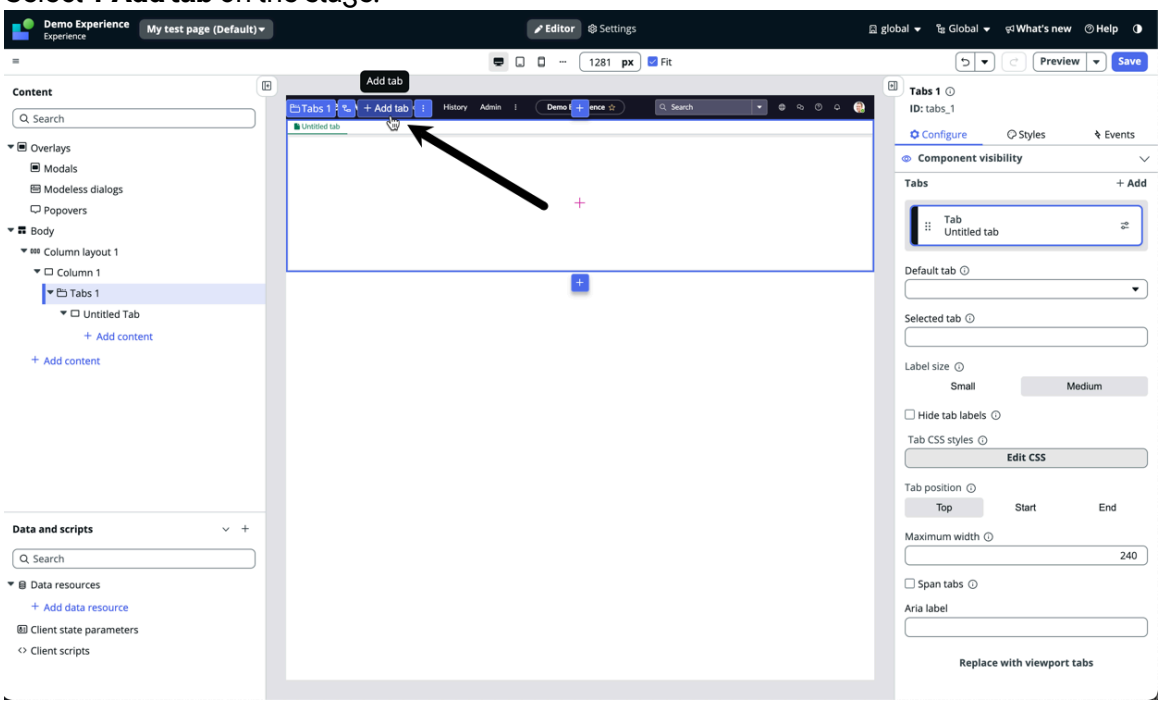
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

5. Select a **Single column** layout.
6. Next, select the **+** icon in the column to open the toolbox.
7. Add a **Tabs** component to your page.



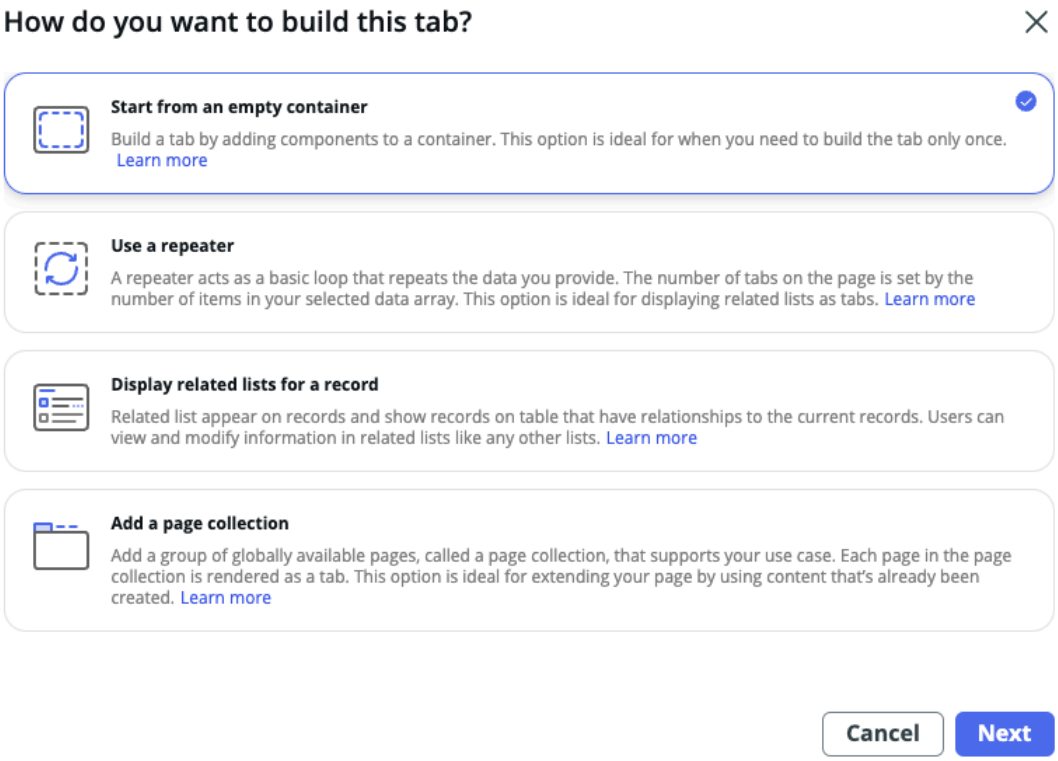
For more information on how to add a component to a page, see Add and configure components.

8. Select **+ Add tab** on the stage.



9. Select **Start from an empty container**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



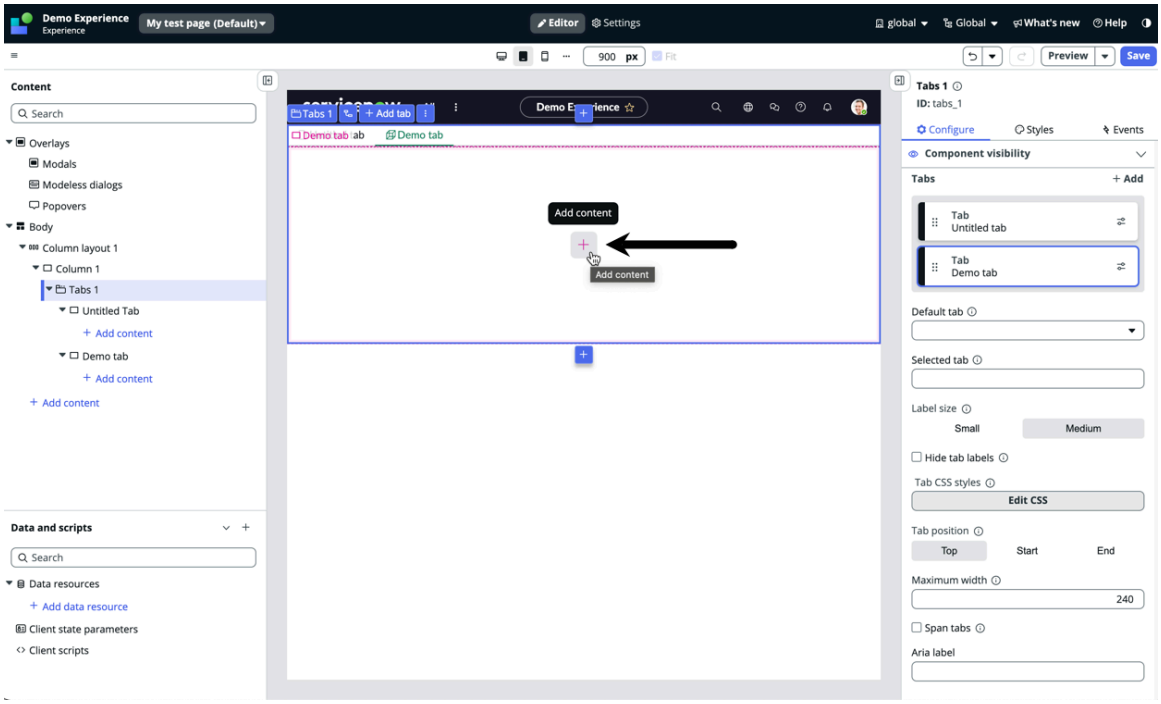
10. Select **Next**.
11. On the form, fill in the fields.

Tab settings form

Field	Description
Tab label	Label for the tab to display on your page.
Icon	Icon to appear next to the tab label. Icons are not required.
Hide tab	Whether to hide or display the tab.

12. Select **Create**.
    The new tab displays in the **Tabs** component.
13. Select the new tab.
14. Select **+** icon.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



15. Select the component that you want to add to the tab.
    Components display under the tab in which they're nested in the content tree. For more information, see Add and configure components.

Result

Your page shows the two tabs that you created. Select each tab to further configure them, add styling, or add an event handler. For more information on styling, see Change the default appearance of components. For more information on adding an event handler, see Manage actions in UI Builder pages.

Add repeater tabs

Use the Tabs component to create a set of repeater tabs by linking tabs to a data array.

Before you begin

Role required: admin

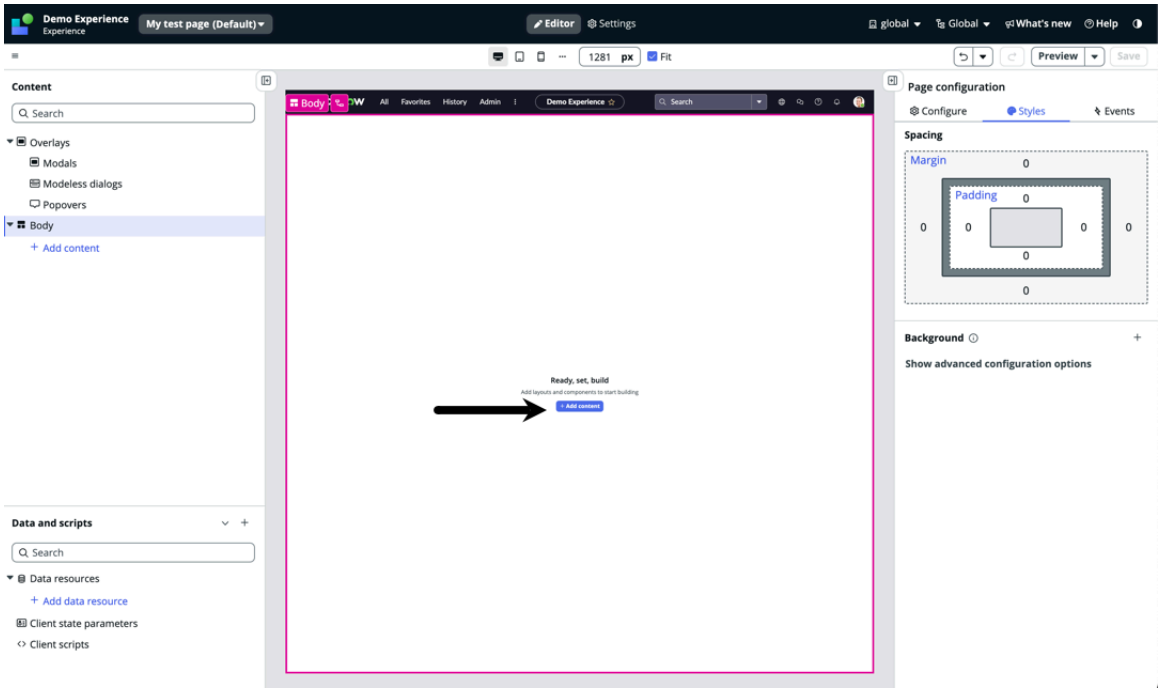
About this task

You can use repeater tabs to create multiple tabs based on the data array you provide. You can pass the icon, label, count, and fields as a key in the object. Use the label key to add names to tabs. Use the field key to pass information to repeater tabs. You can bind a data broker, client state, or a client script to the data array provider to return the array of objects with the correct schema.

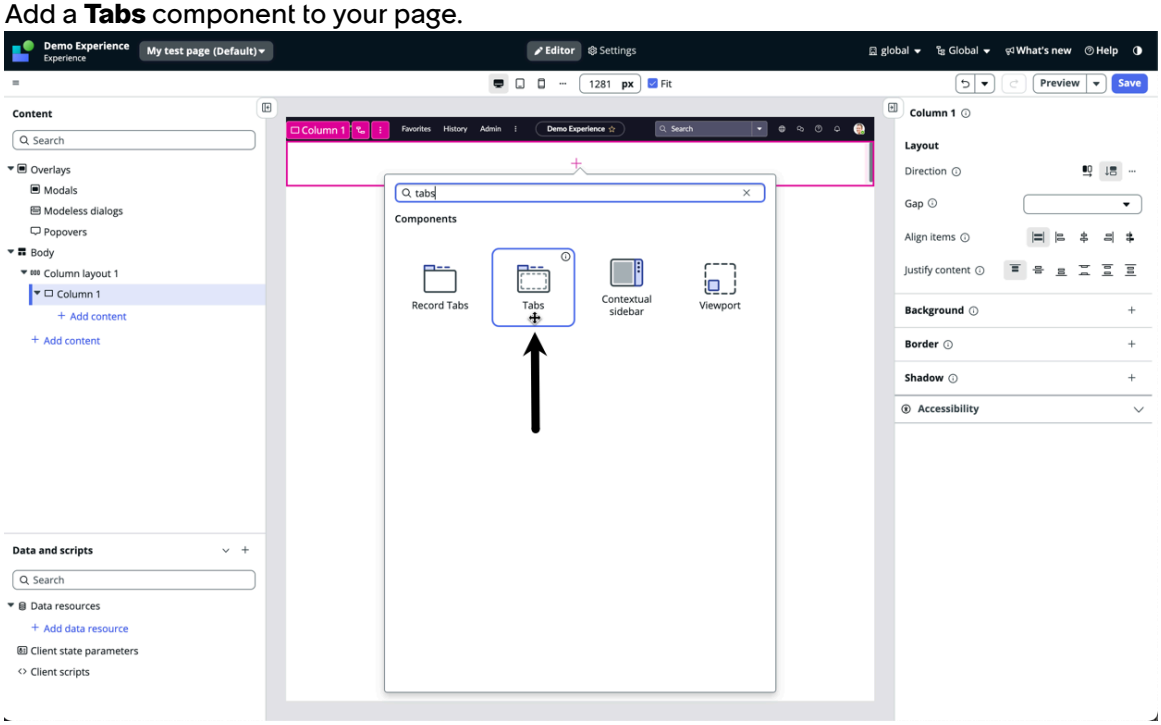
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create or open a page or page variant.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. Select the **+ Add content** button on the stage to open the toolbox.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



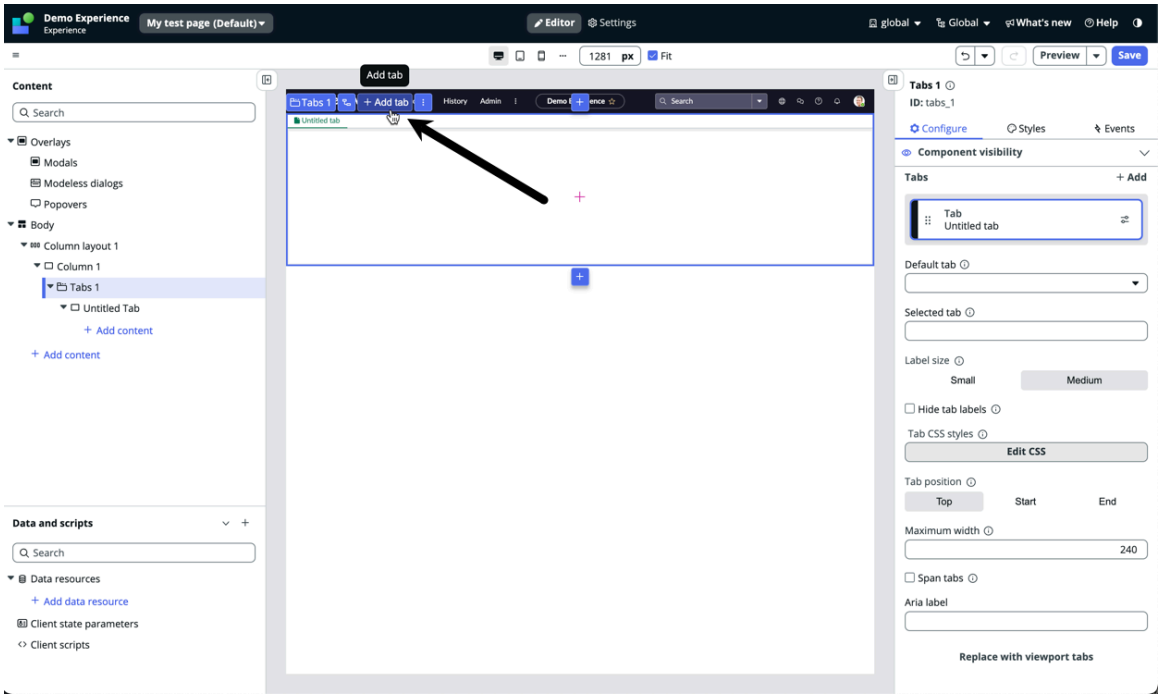
5. Select a **Single column** layout.
6. Next, select the **+** icon in the column to open the toolbox.
7. Add a **Tabs** component to your page.



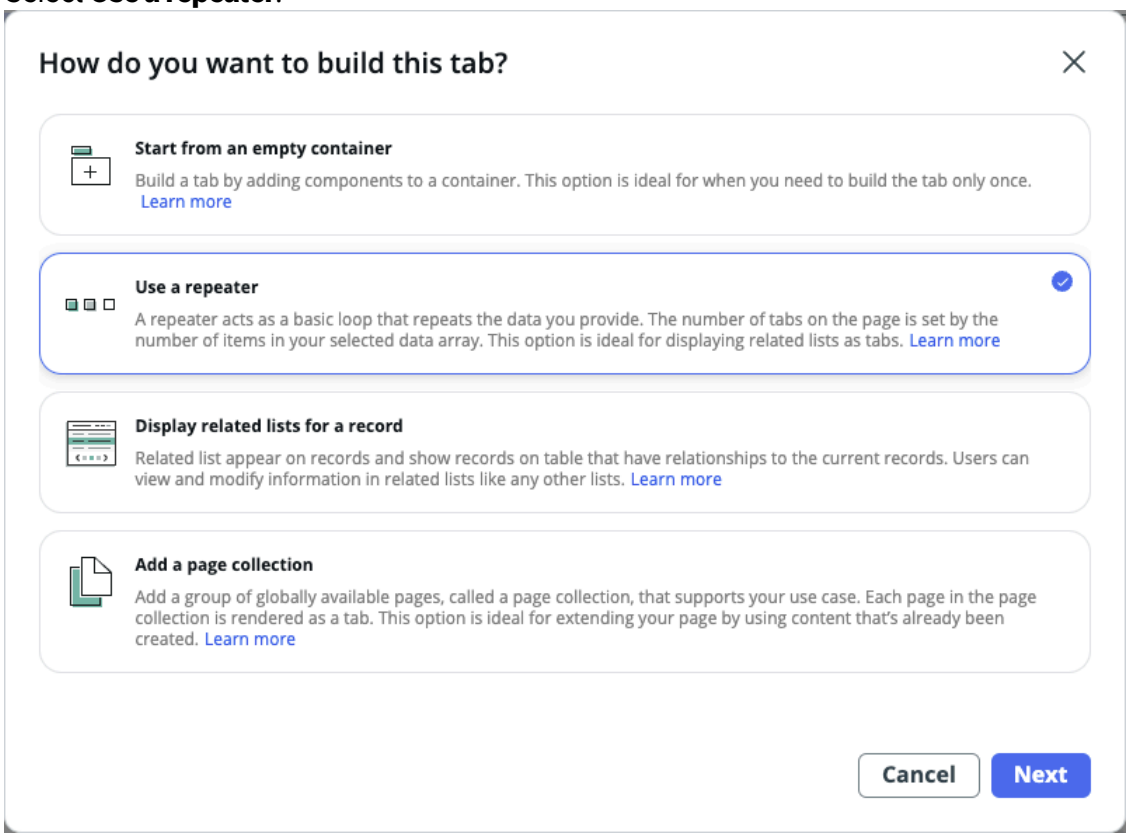
For more information on how to add a component to a page, see Add and configure components.

8. Select **+ Add tab** on the stage.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

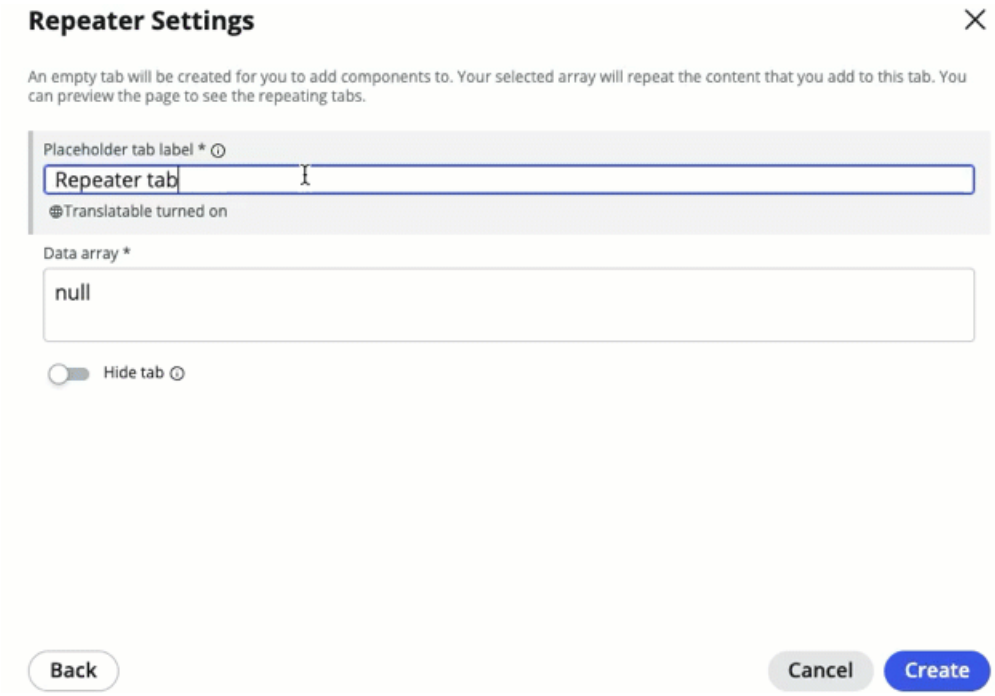


9. Select **Use a repeater**.



10. Select **Next**.
11. Enter a label for the repeater tab in the **Placeholder tab label** field.
    This label is assigned to the first repeater tab and is the only tab displayed in the page preview.
12. Select **Edit** in the **Data array** field.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

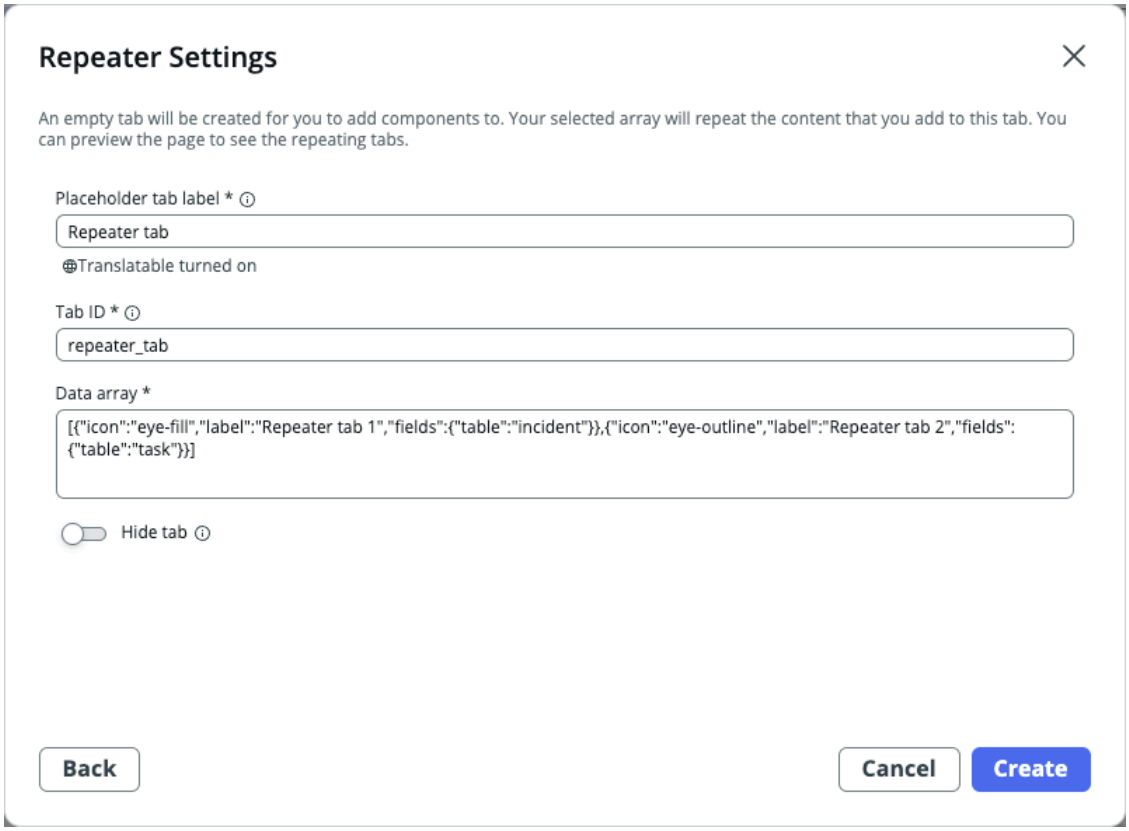


13. Configure the data array in the JSON editor.

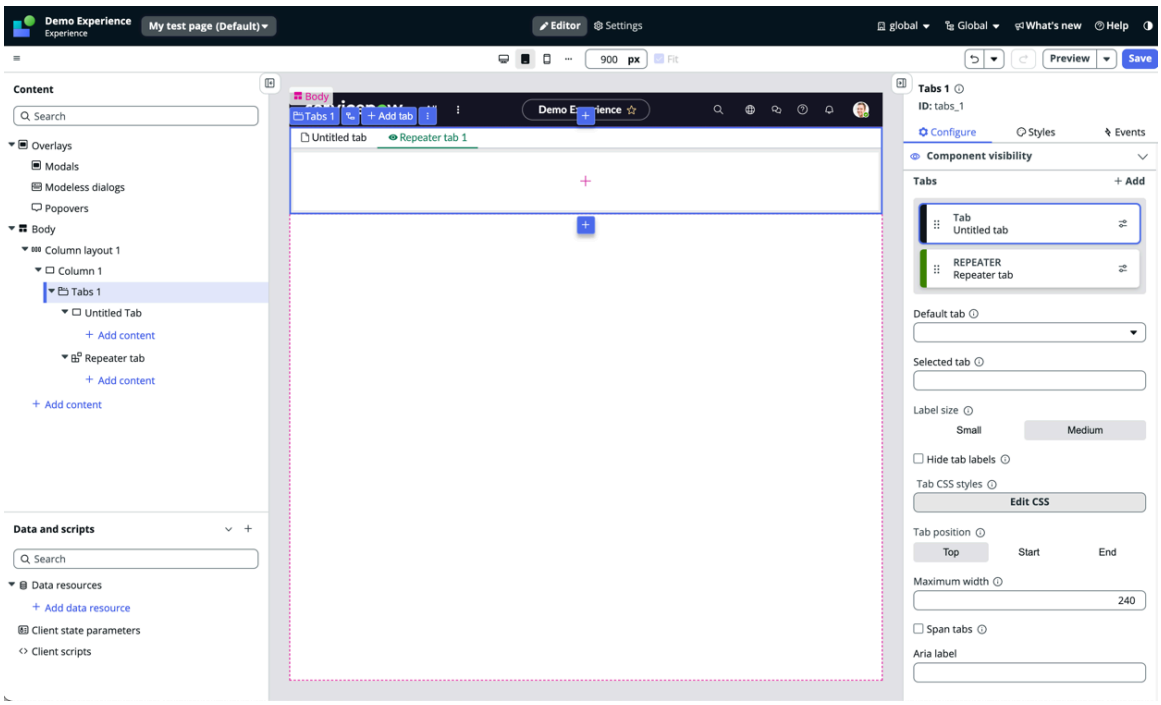
Key	Data type	Description
icon	String	Name of the icon that appears in UI Builder.
label	String	Required. Display name of the tab in UI Builder.
count	Number	Value to provide on the tab label. Dynamically bind to a data broker or a client script that will be displayed on the label.
fields	Object	Object that can be used to store information and to pass down to the components in the tabs.

14. Select **Apply**.
The data array field displays the data array configuration.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



15. Select **Create**.
    The new repeater tab appears on the page and in the content tree.



16. Select **Save**.
17. Add components to the repeater tab.
18. Select Preview to verify that the repeater tabs display correctly on your page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add related list tabs

Create a set of tabs on a page by linking to a related list.

Before you begin

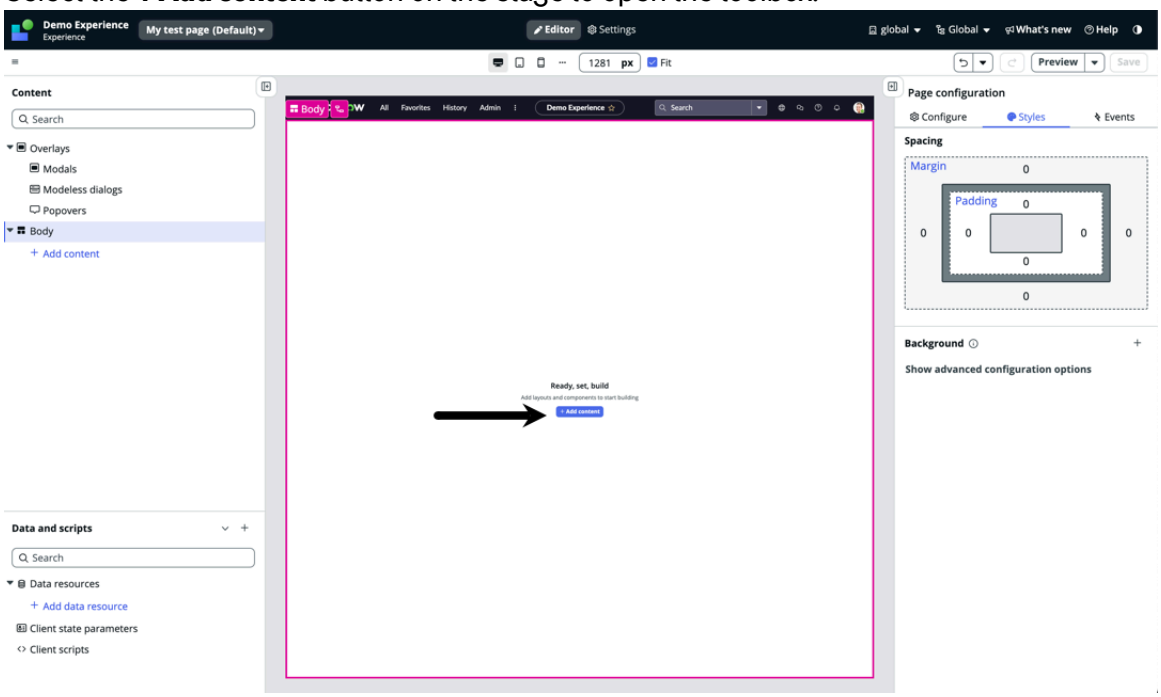
Role required: admin

About this task

The related list tab automatically populates tabs based on the record that your page is displaying. For example, by default the related lists for the user table are Roles, Groups, Delegates, and Visibility domains. If your page is displaying a user record, then creating a related list tab adds all four of these related lists as tabs on your page. You can preview the page to see the related lists that were added. A record controller is required to create related list tabs. To add a controller to your page, see Bind data to UI Builder pages using controllers (advanced feature) for more information.

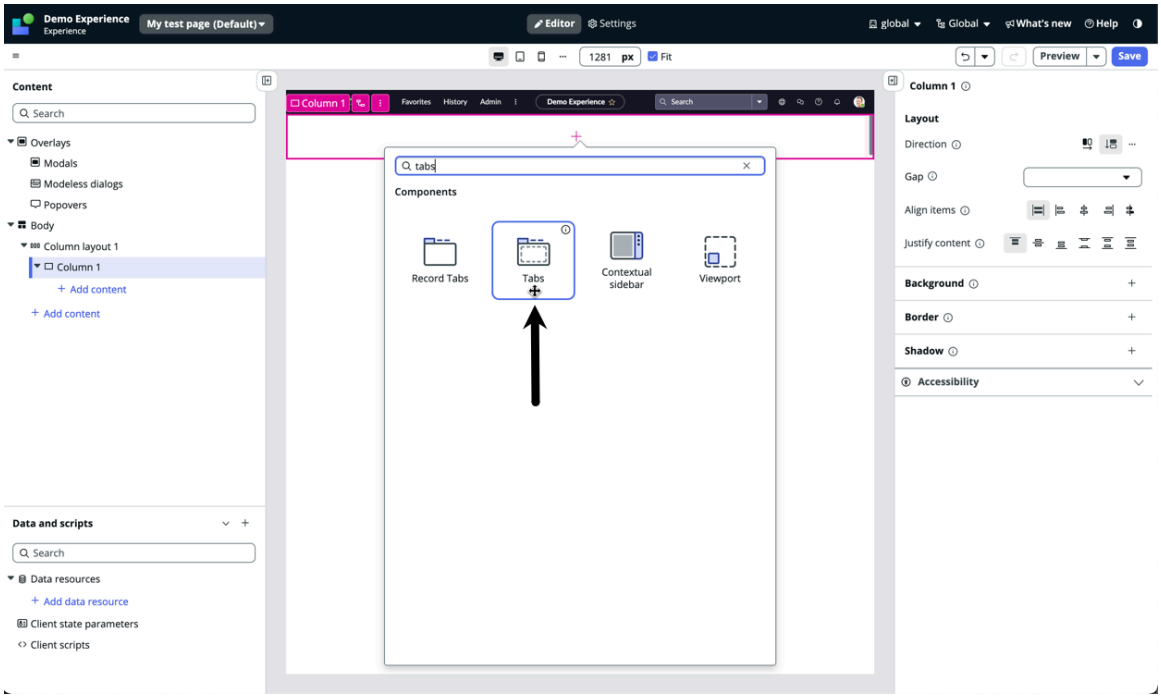
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create or open a page or page variant.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. Select the **+ Add content** button on the stage to open the toolbox.



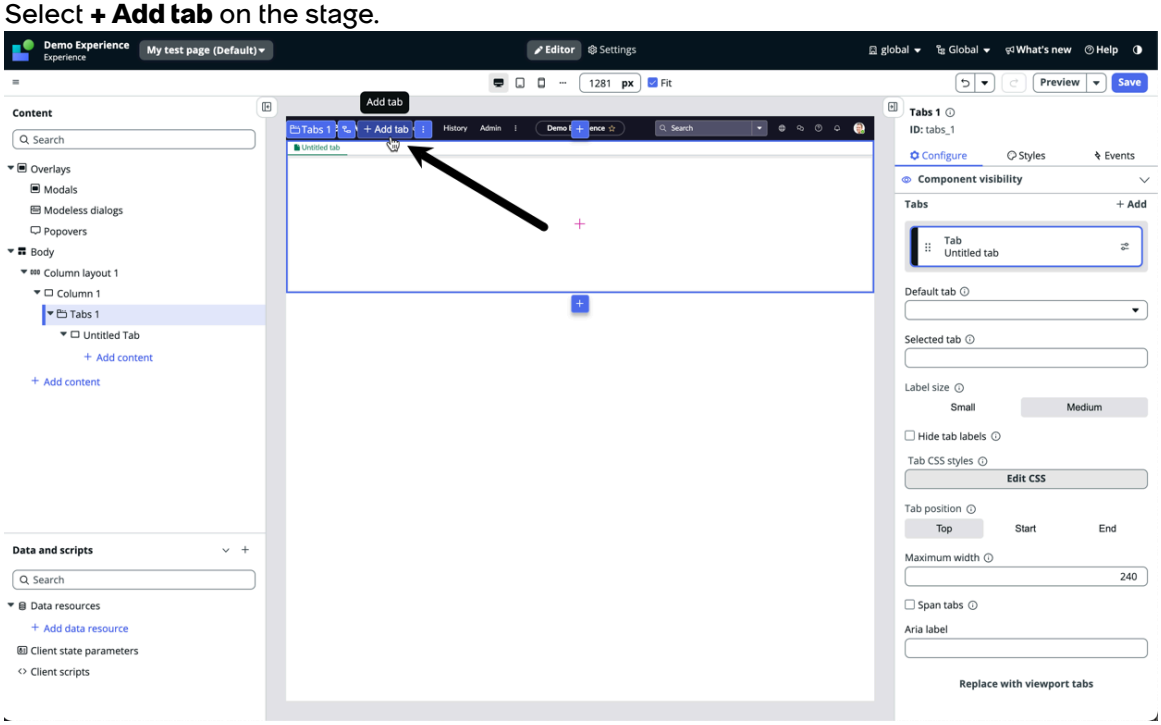
5. Select a **Single column** layout.
6. Next, select the **+** icon in the column to open the toolbox.
7. Add a **Tabs** component to your page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



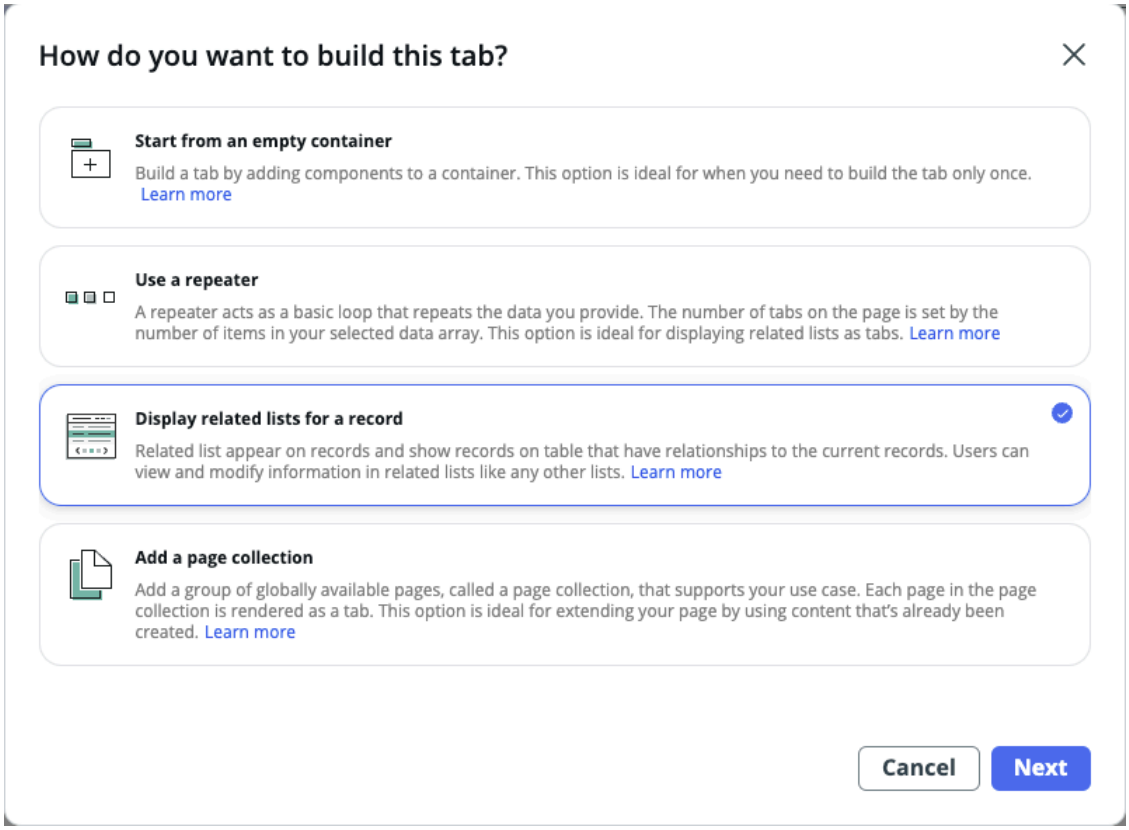
For more information on how to add a component to a page, see Add and configure components.

8. Select **+ Add tab** on the stage.



9. Select **Display related lists for a record**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



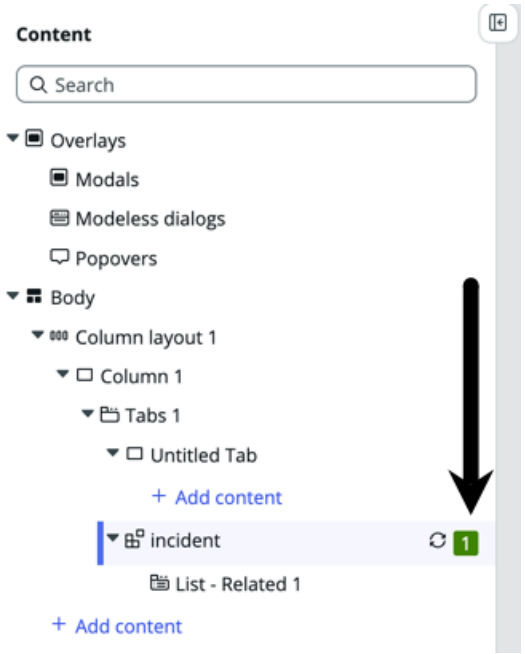
10. Select **Next**.
11. On the form, fill in the fields.

Tab settings form

Field	Description
Placeholder tab label	Placeholder label to appear on the preview tab of your page.
Hide tab	Whether you want to hide or display the tab.
Record Controller	Record controller with the related list.

12. Select **Create**.
    The new related list tab displays in the **Tabs** section of the configuration panel. Only one placeholder tab appears in the page preview. The green icon next to the related list tab in the content tree shows how many tabs appear on your page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



13. Select **Save**.
14. View and test your page by selecting Preview.

Add page collection tabs

Use page collection tabs to recreate the same tabs across multiple pages in UI Builder.

Before you begin

Role required: admin

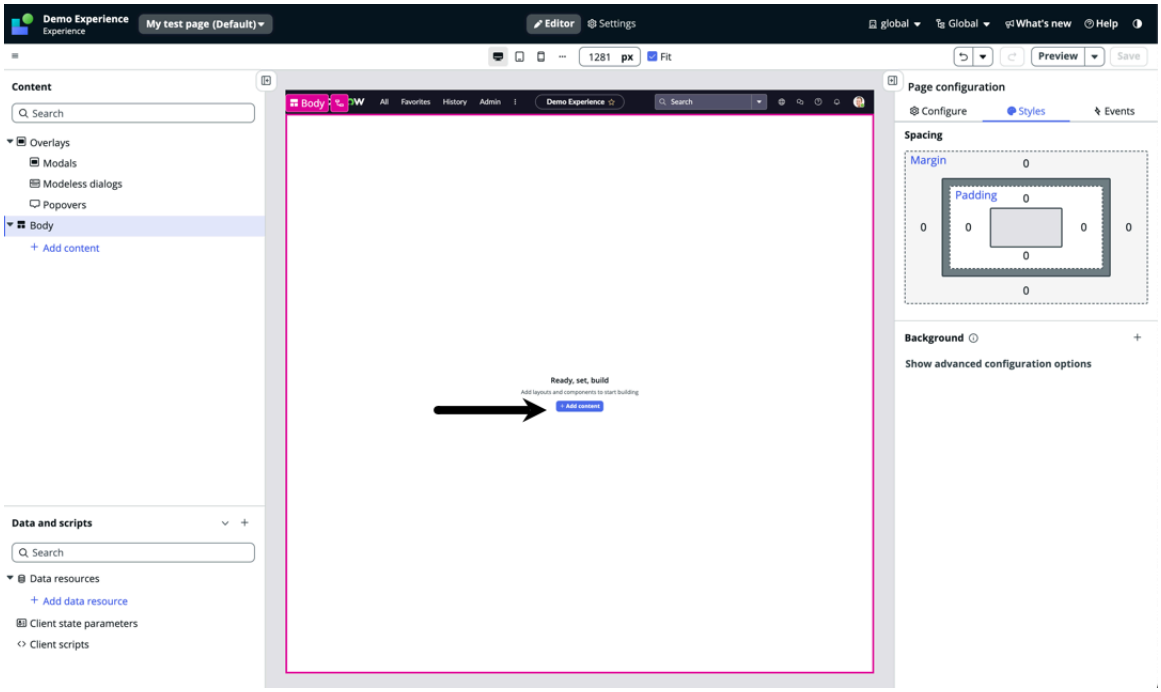
About this task

A page collection is a group of prebuilt globally available pages. Use page collection tabs to render each page in a page collection as a tab. You can select an existing page collection or can create your own. A controller is required to add a page collection. For more information about page collections, see #unique_350.

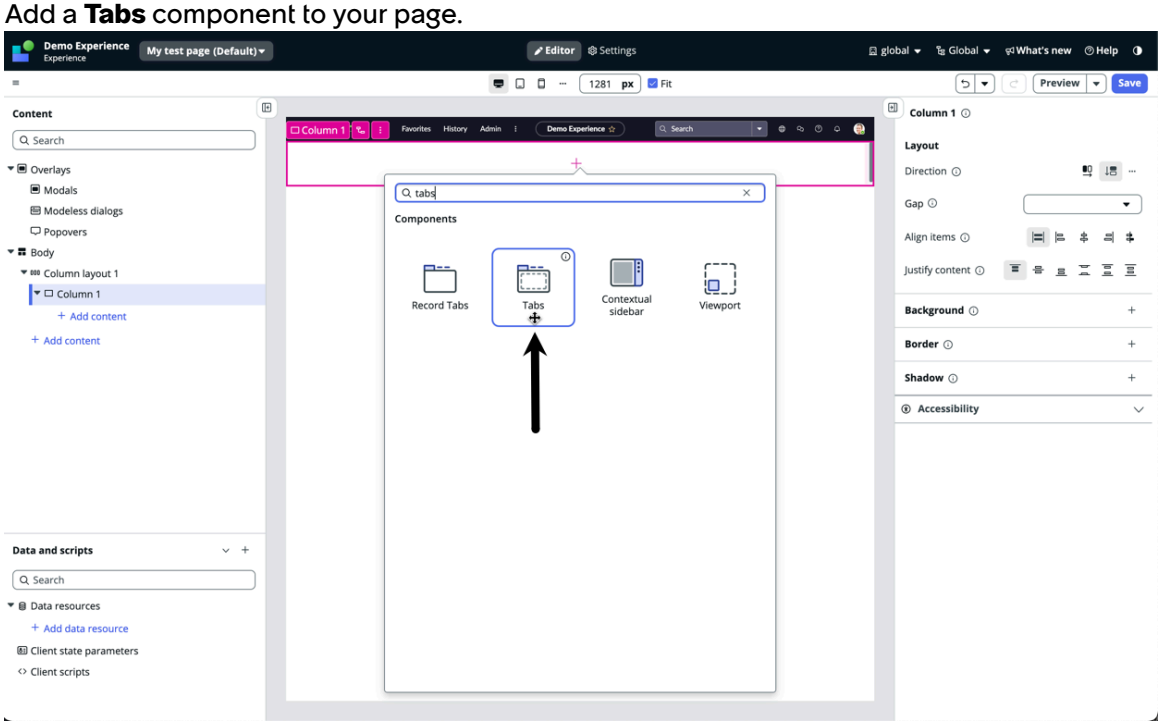
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create or open a page or page variant.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. Select the **+ Add content** button on the stage to open the toolbox.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



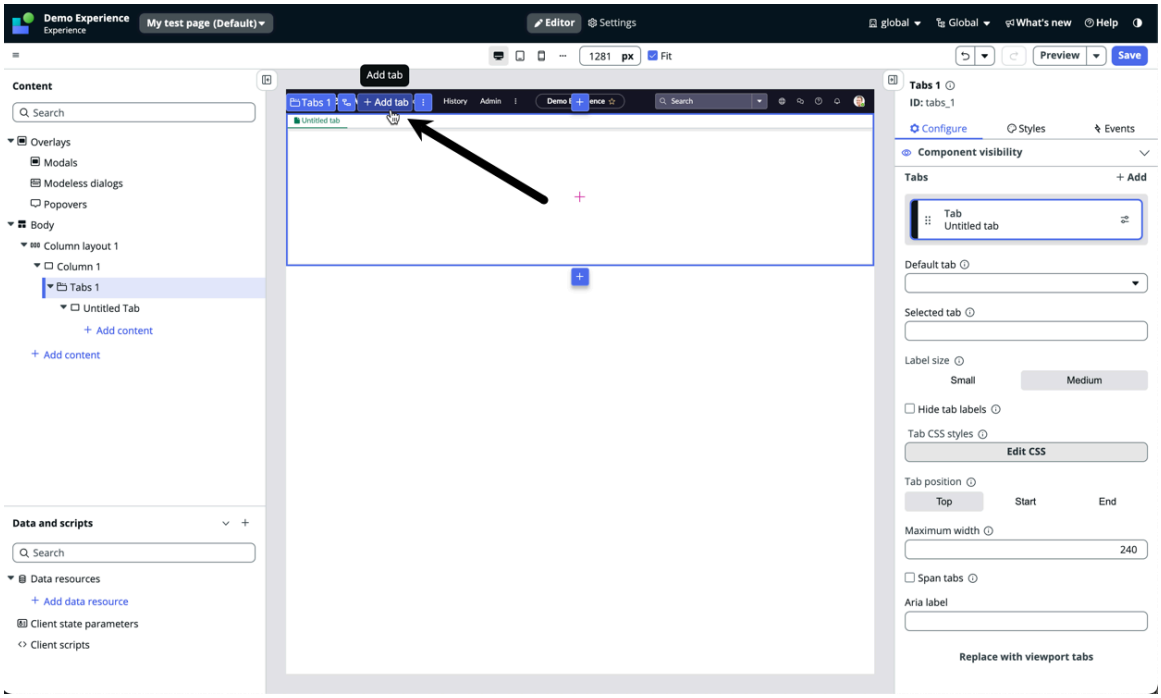
5. Select a **Single column** layout.
6. Next, select the **+** icon in the column to open the toolbox.
7. Add a **Tabs** component to your page.



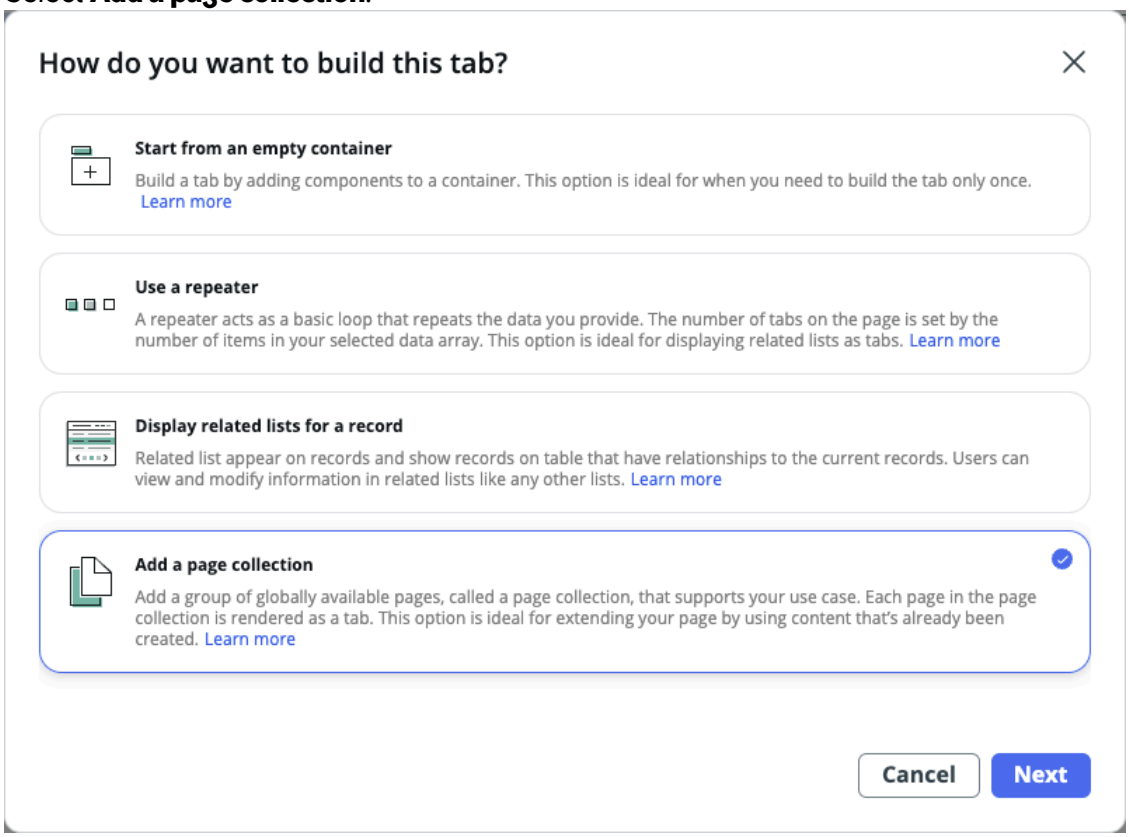
For more information on how to add a component to a page, see Add and configure components.

8. Select **+ Add tab** on the stage.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

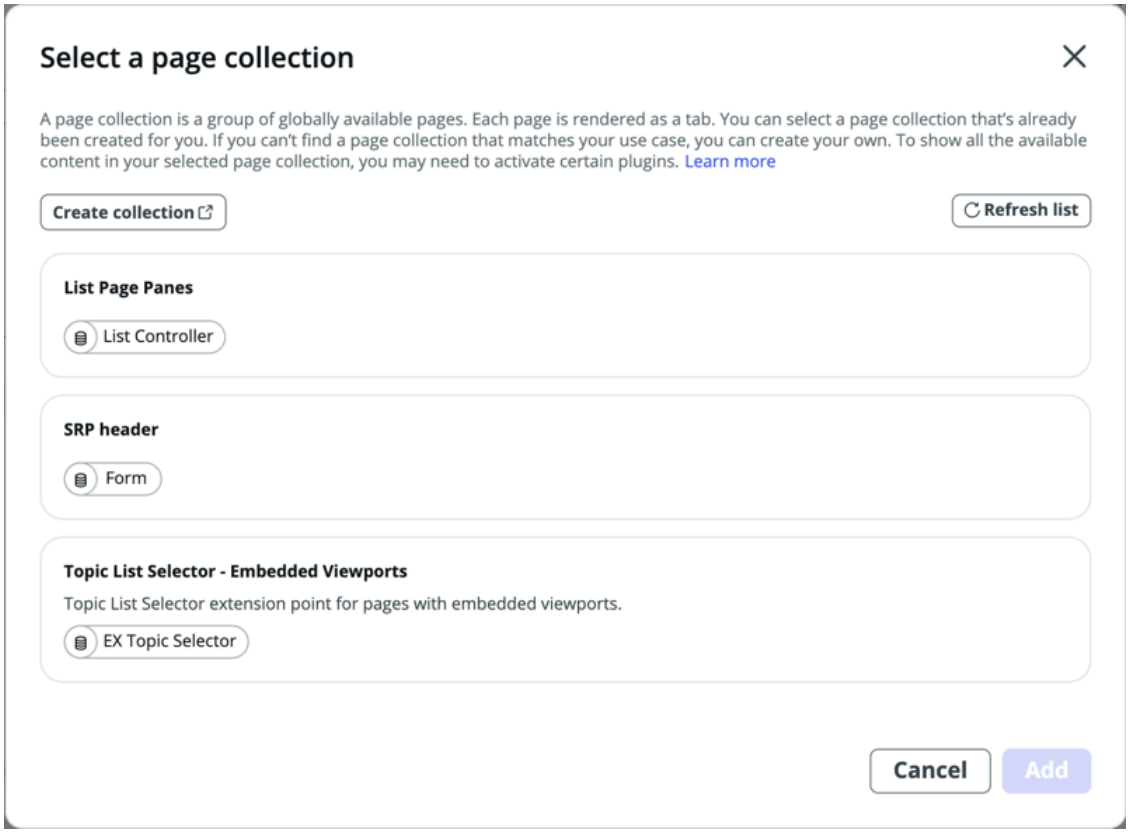


9. Select **Add a page collection**.

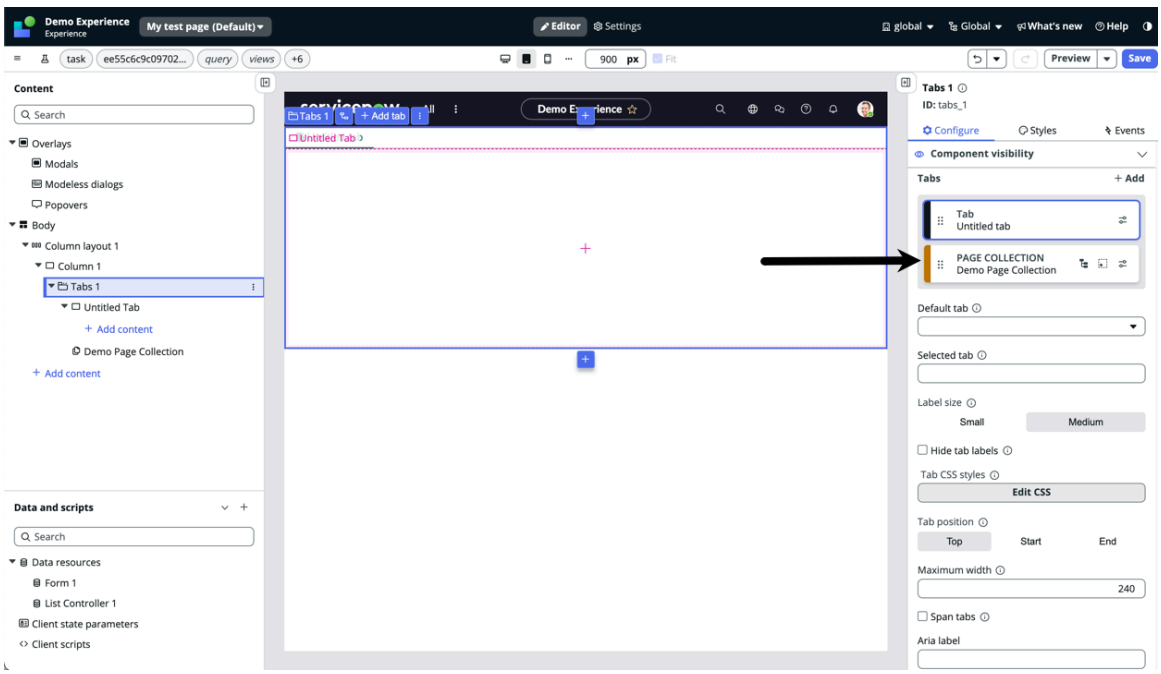


10. Select **Next**.
11. Select a page collection from the list or create a collection by selecting **+ Create collection**. To create your own page collection, see Create a page collection across multiple UI pages.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



12. Select **Add**.
The new page collection tab displays in the **Tabs** section of the configuration panel. Only one placeholder tab displays in the page preview.



13. Select **Save**.

14. View and test your page by selecting Preview.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add a contextual sidebar

Add a contextual sidebar to a page with UI Builder to display related content using a vertical tab structure.

Before you begin

Role required: ui_builder_admin

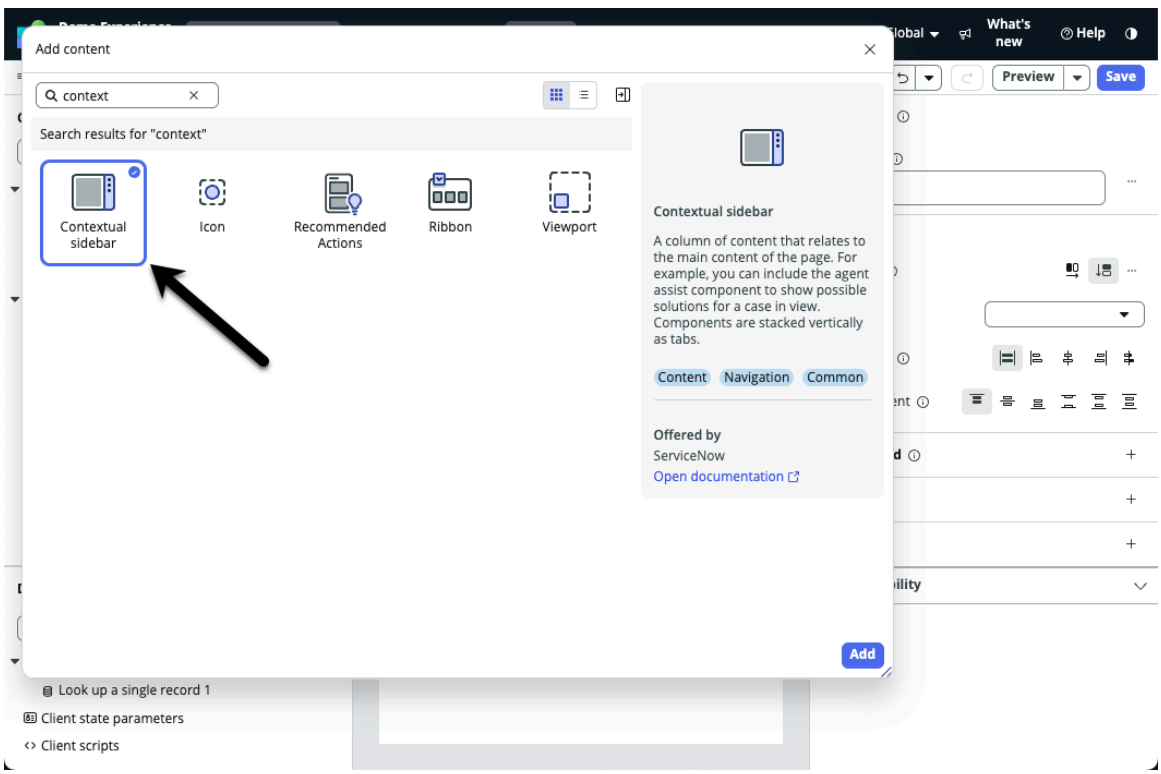
About this task

You can configure the contextual sidebar in the same way you would configure a tab component. Each vertical tab in the contextual sidebar renders custom content. You can place components in each tab for quick access directly from the page.

For example, a document creation page might have the attachments component on one tab, enabling you to attach files related to the document that you're working on. On another tab, there could be a nested comments component, enabling you to add and view comments related to the document.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create or open a page or page variant.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder. For information about creating a page variant, see Create a page variant
4. On the stage, open the toolbox by selecting the **+ Add content** button.
5. Select the **Single column** layout tile.
6. In the column, open the toolbox by selecting the **+** icon.
7. Select the **Contextual sidebar** component to add it to your page.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The contextual sidebar appears on the stage, and the Tabs section appears in the configuration panel.

8. In the configuration panel under the Tabs section, select the settings icon (⚙) to configure the tab.

   a. On the form, fill in the fields.

   Tab settings form

Field	Description
Tab label	Label that displays on the tab for your page.
Icon	Icon that appears next to the tab label. **Note:** Icons aren't required.
Hide tab	Option to hide or display the tab.

   b. Select **Save**.

9. On the stage in the contextual sidebar, open the toolbox by selecting the **+** icon.
10. Select the component that you want to add to the tab.
    The component displays under the tab in the content tree. For more information, see Add and configure components.
11. **Optional:** To add another tab, select **+ Add** under the Tabs section.

What to do next

For detailed information about the configuration of the contextual sidebar and its properties, see Tabs UIB Setup on the ServiceNow Developer Site.

Add forms to UI Builder pages

Use the Form component to add one or more forms to UI Builder pages.

Add functionality to your UI Builder pages by including forms. You define the fields on the form and their properties, such as making them required. Then, collect data as the form is completed and submitted.

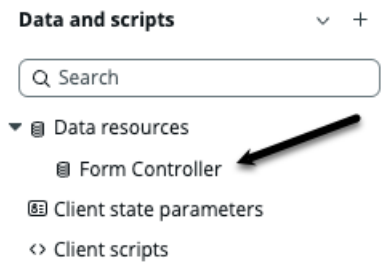
You can add more than one form to a single page. You can also add a form to a page that already contains a component with a nested form. Sample use cases include:

- Extend record pages by adding an inline tab with a form using its own form controller instance.
- Add modals with a form on a record page.

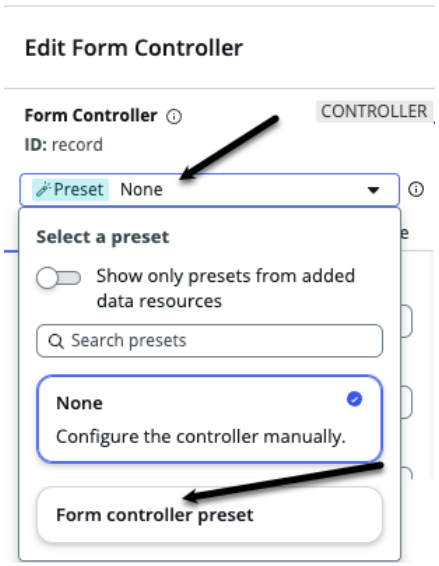
For existing pages with forms created in a pre-Xanadu ServiceNow release, you must apply a preset to the original form before adding another form to the page. Applying the preset is a prerequisite to adding multiple forms to a page and enables multiple forms to work as expected on a page. The form controller preset should be applied onto all form controllers.

1. Open the page containing an existing form.
2. In the data drawer, expand the **Data resources** list and select the original form controller.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



3. Select the **Preset** field.
4. Select **Form controller preset**.

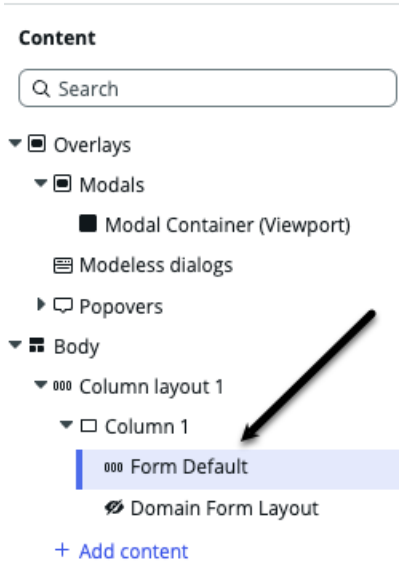


5. Select **Apply**.
6. Select the **X** to close the **Edit Form Controller** pop-up.

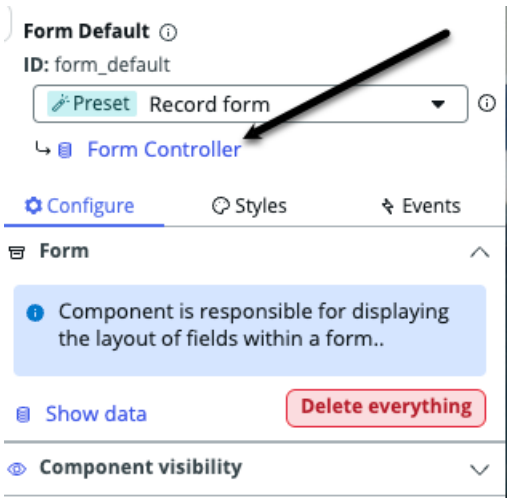
   Exactly one of your form controllers should have the **Is mapped to app Shell** property set to true. This property is used to specify the primary form on the page. The primary form is responsible for handling global events. You shouldn't set the property to true for more than one form controller or have zero form controllers with the property set to true.

7. Open the page containing one or more forms.
8. In the content tree, select a form.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

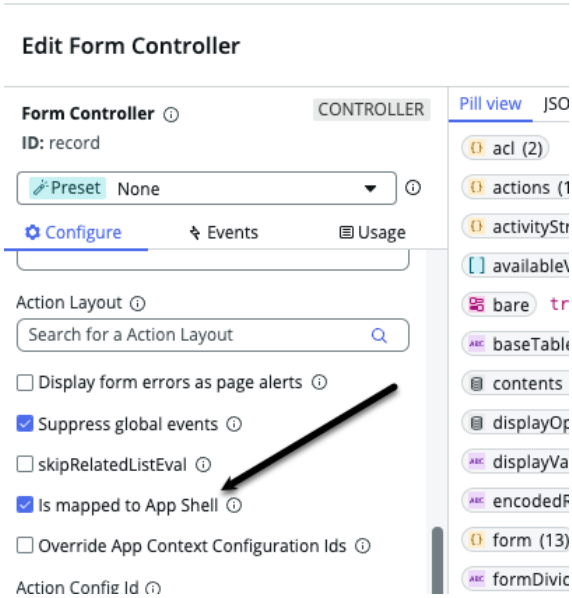


9. In the configuration panel, on the **Configure** tab, select **Form Controller**.



10. On the **Edit Form Controller** pop-up, scroll down in the **Form Controller** list to find the **Is mapped to App Shell** option.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



11. Select or clear the option for each form component on the page to confirm that exactly one form controller is mapped to the app shell.

Advanced form event handling

Experienced developers with knowledge of conflict event handling may find the following details useful.

If isMapped to app shell is set to true, the form handles these events automatically:

Screen Status Changed

- Description: Action to indicate that a form is dirty.
- Output: `CTRL_RECORD#SCREEN_STATUS_CHANGED`

Update Configuration Menu Requested

- Description: Action to set record configuration menu items on the avatar menu.
- Output: `CTRL_RECORD#UPDATE_CONFIGURATION_MENU_REQUEST`

Phone Requested

- Description: Action to make a call when the CTI plugin is enabled.
- Output: `CTRL_RECORD#PHONE_REQUESTED`

Form Loading State Changed

- Description: Action to show a loading spinning when that form is loading data.
- Output: `CTRL_RECORD#FORM_LOADING_STATE_CHANGED`

For detailed information about the Form component and its properties, see Form Overview on the ServiceNow Developer Site.

Create modals in UI Builder

Use modals in UI Builder with components to provide alerts or calls to action for a user. UI Builder comes with modals to save time and effort.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

What Modals are

Modals, also known as dialogs, are windows that overlay another content window. Modals take control of the user experience. Users cannot interact with the overlaid window until the modal is closed. Modals can contain different types of content such as:

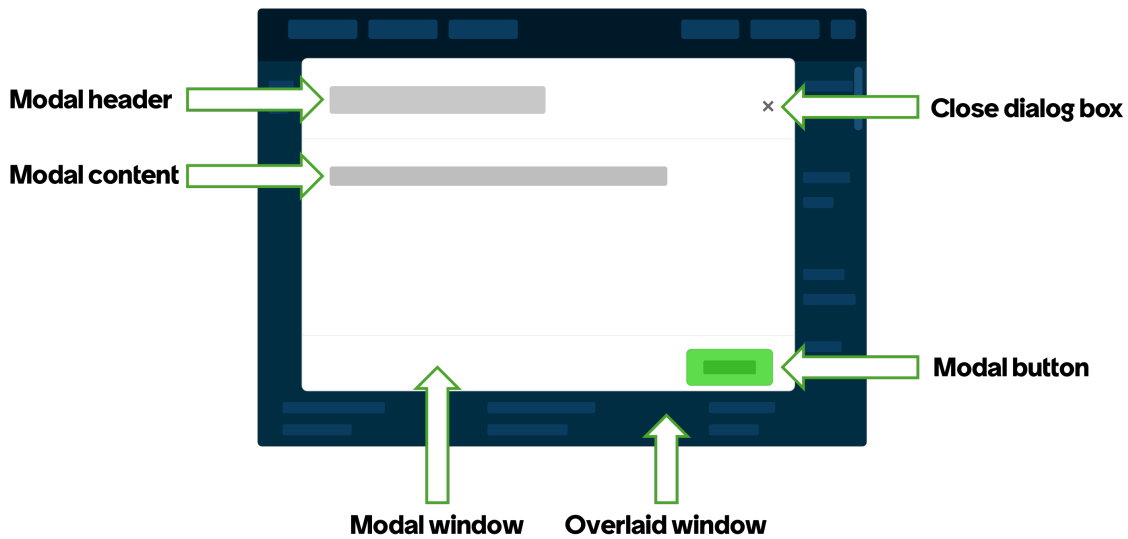
- Static text
- Dynamic text
- Forms
- Images
- Buttons

UI Builder has preconfigured modals available. You can add a modal to your component. Then, configure the content of the modal, and how it displays on the screen. Add an event handler to the modal to perform an action when a user selects it. The action can alert a user about something, or ask a user to confirm an action. A modal is a way to ensure that a user knows what is happening. For example, a modal may ask a user to confirm a selection before continuing whatever action they are performing on the main page.

Modal Anatomy

Modals in UI Builder can have:

- Modal header
- Modal content
- Close dialog button (no action taken by the modal)
- At least one Modal button (action can be taken by the modal)



Modal types

Different types of modals are available in UI Builder, as shown in the following table.

Modal type	Description
Alert	An Alert modal provides information relating to the component action. For example, when a

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Modal type	Description
	user presses a delete button, you could have an alert pop-up that lets the user know they cannot undo a delete action.
Confirm	A Confirm modal asks a user to confirm the component action. For example, when a user presses a delete button, the user would have to confirm the deletion of data. You can choose the confirm options from the primary and secondary button label fields, such as Yes/Cancel.
Confirm and destroy	A Confirm and destroy modal is more directive, usually relating to deleting or erasing content. It lets the user know the seriousness of an action, and asks them whether they want to proceed with the action.
Custom	Custom modals address scenarios that are not handled using the standard modals. Custom modals can be thought of as a container component on a modal. You can add a custom layout, components, events, and data resources just like you do on a page. The custom modal uses layouts to let you fully design what information you want in the modal. Layouts also decide where the information sits within the modal screen. You can use Cascading Style Sheets (CSS) styling to change the visual look of the modal.
iframe	Use iframe to bring content into your modal from existing iframe content from a URL and data.
Modal viewport	Dynamically pass content into your viewport modal through an event binding using a client script.

Event handlers and modals

Expose events to modals to handle call-to-action events. For example, a primary action, secondary action, and so on. You configure the data by adding an event handler and invoking a data resource. It is as simple as adding a new event handler for the component that has a modal. Or you can add an event handler to the modal itself. You select the event that you want associated with the component or modal and add it. See Add modal to component for detailed instructions.

Add modal to component

Learn how to add a modal in UI Builder. A modal is a window that appears when you click a component. For example, a modal might display when a delete button component is clicked, prompting the user to confirm deleting a record.

Before you begin

Role required: ui_builder_admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

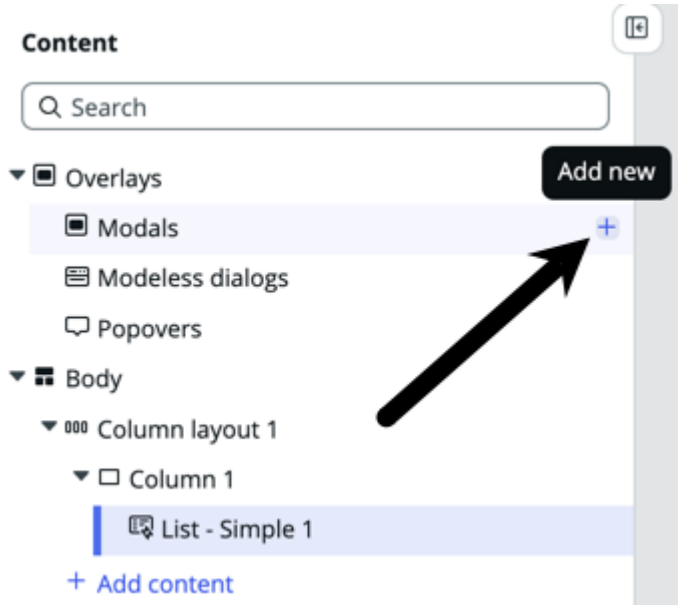
About this task

A modal is a screen that appears when an event handler is triggered by an event such as a button being clicked. The following procedure shows an example of how to add a Confirm modal and an associated event handler to a button component.

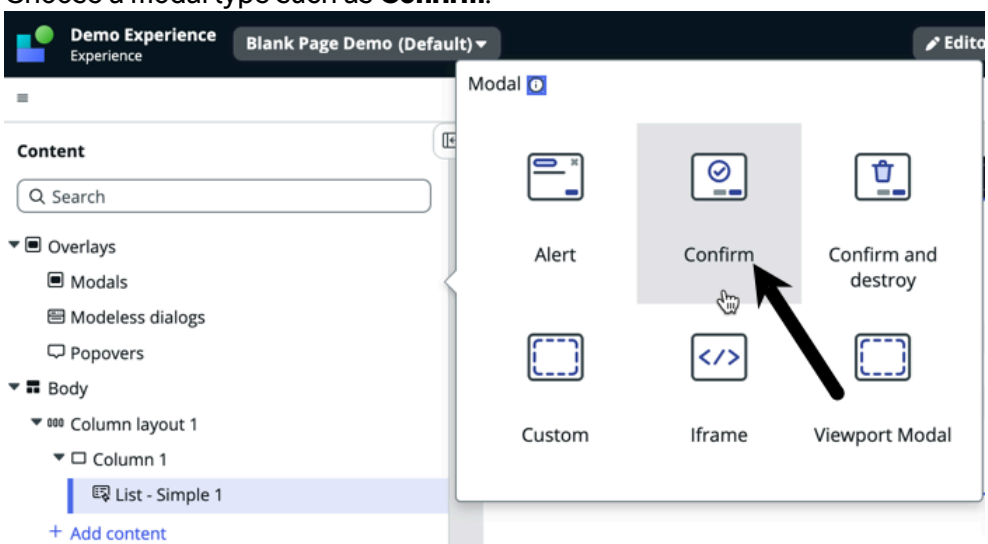
Procedure

1. Navigate to **All** > **Now experience framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Open or create a page or page variant.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. Add a modal to the page.

   a. Select the **+** icon in the content tree next to **Modals and popovers**.



   b. Choose a modal type such as **Confirm**.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

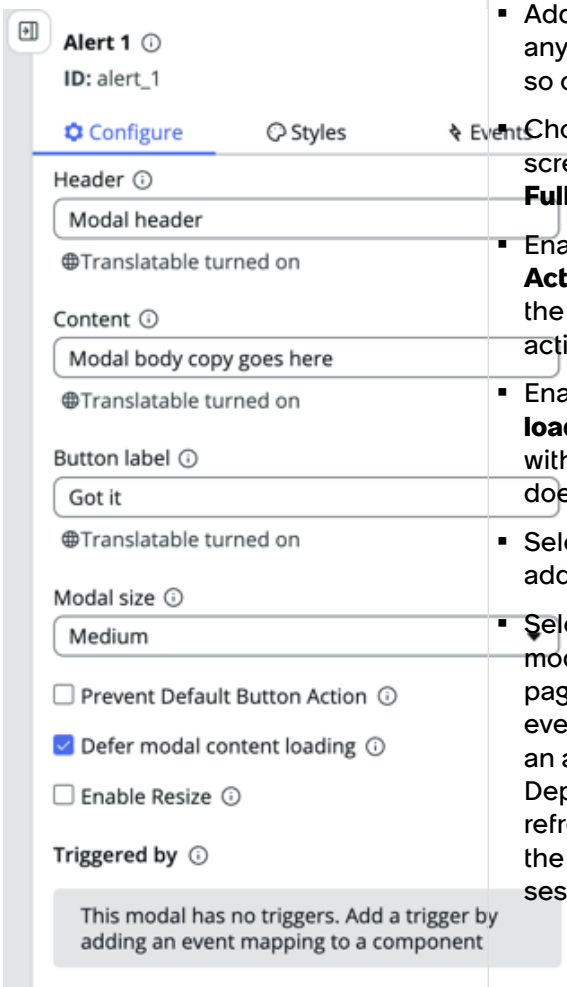
Modal type	Description
Alert	An Alert modal provides information relating to the component action. For example, when a user presses a delete button, you could have an alert pop-up that lets the user know they cannot undo a delete action.
Confirm	A Confirm modal asks a user to confirm the component action. For example, when a user presses a delete button, the user would have to confirm the deletion of data. You can choose the confirm options from the primary and secondary button label fields, such as Yes/Cancel.
Confirm and destroy	A Confirm and destroy modal is more directive, usually relating to deleting or erasing content. It lets the user know the seriousness of an action, and asks them whether they want to proceed with the action.
Custom	The Custom modal uses layouts to let you fully design what information you want in the modal. Layouts also decide where the information sits within the modal screen. You can use Cascading Style Sheets (CSS) styling to change the visual look of the modal, such as background color.
iframe	Use iframe to bring content into your modal using existing iframe content from a URL and data.
Modal viewport	Dynamically pass content into your viewport modal through an event binding using a client script. See Bind an event to a component for more information on binding an event to a component.

c. Configure the modals as shown in the table.

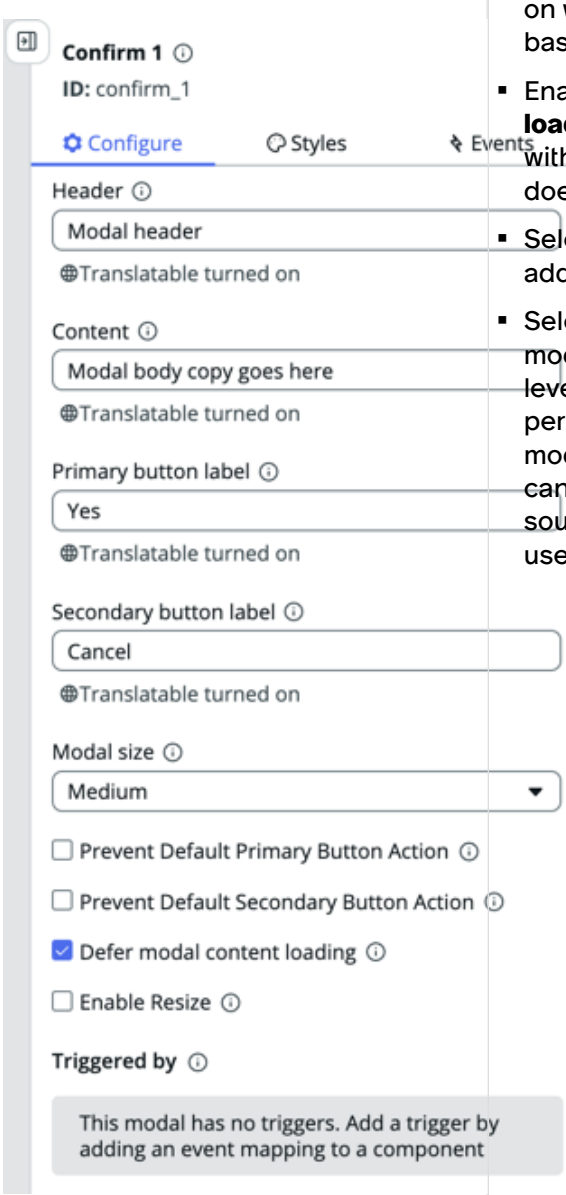
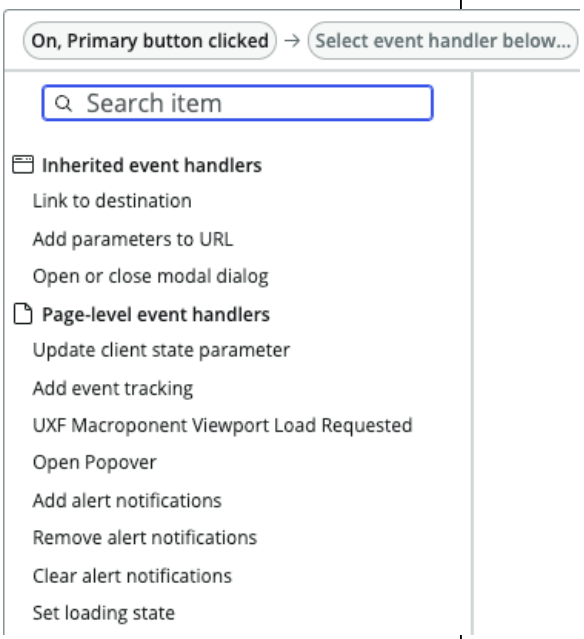
Configure each modal differently, depending on what the modal requires. Change what information goes into the modal, the size of the modal, and how it looks. You can add an event handler to the modal that performs the action for the modal, such as opening or closing the modal.

Modal	Configuration
Alert	▪ Add a header, which is the title of the modal. ▪ Write the content that you want displayed in the modal. The content tells the user what the alert is.

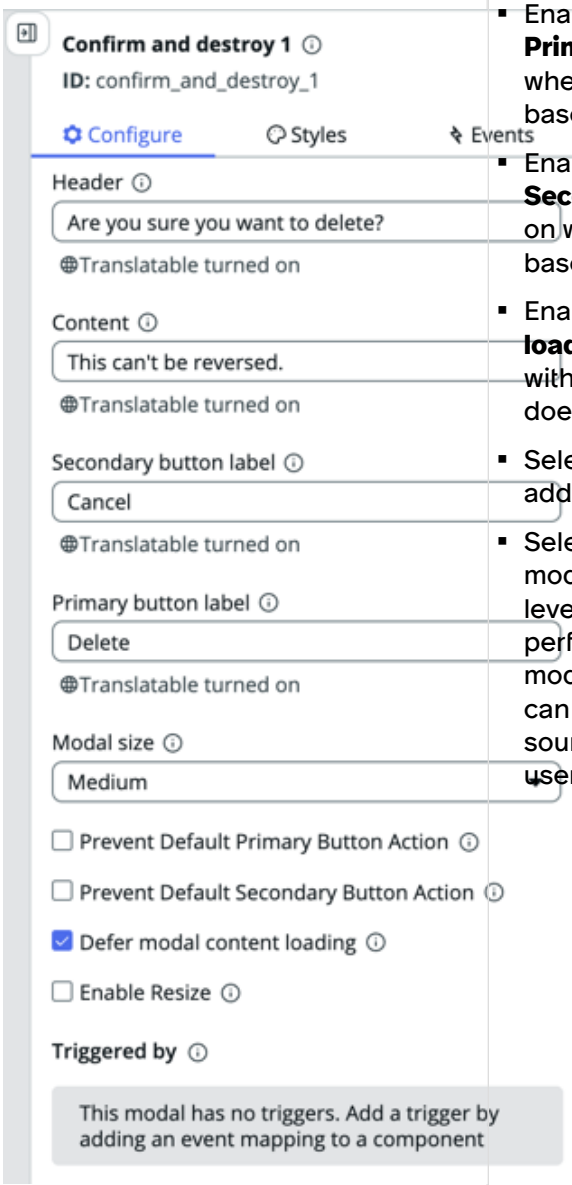
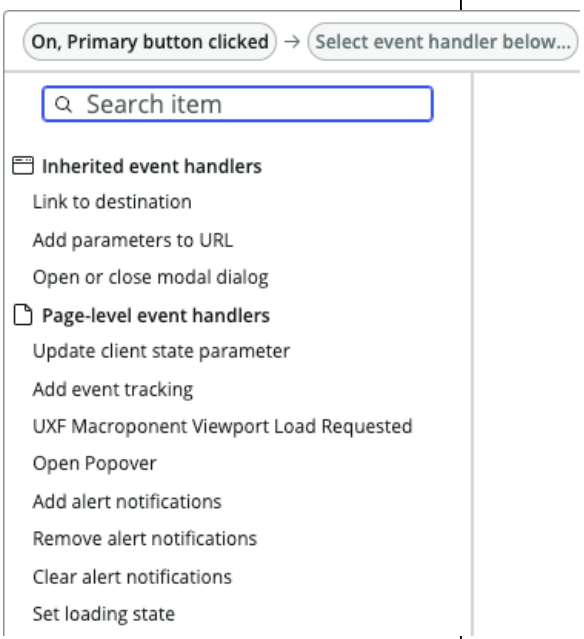
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Modal	Configuration
	▪ Add text for the button label. It can be anything you want, such as **OK**, **Yes**, and so on. ▪ Choose the size of the modal on the screen. Select **Small**, **Medium**, **Large**, or **Fullscreen**. ▪ Enable or disable **Prevent Default Button Action**, depending on whether you want the modal to close based on the default action. ▪ Enable or disable **Defer modal content loading**. If you disable it, the modal loads with the page. If you enable it, the modal doesn't load when the page loads. ▪ Select **Events** > **Add event mapping** to add an event handler to the modal. ▪ Select an event handler to apply to the modal, then select **Add** to add it to the page. Choose from inherited or page-level event handlers. Event handlers perform an action such as open or close a modal. Depending on the modal type, you can refresh data for the App Shell data source, the user session for GraphQL, or a user session for Transform.
Confirm	▪ Add a header, which is the title of the modal. ▪ Write the content that you want displayed in the modal. The content tells the user what the alert is. ▪ Add text for the primary button. The primary button is the main action button for users, such as **Yes**, **Add**, and so on. ▪ Add text for the secondary button. The secondary button is usually the no choice for users, such as **Cancel**, **No**, and so on. ▪ Choose the size of the modal on the screen. Select **Small**, **Medium**, **Large**, or **Fullscreen**. ▪ Enable or disable **Prevent Default Primary Button Action**, depending on whether you want the modal to close based on the default action. ▪ Enable or disable **Prevent Default Secondary Button Action**, depending

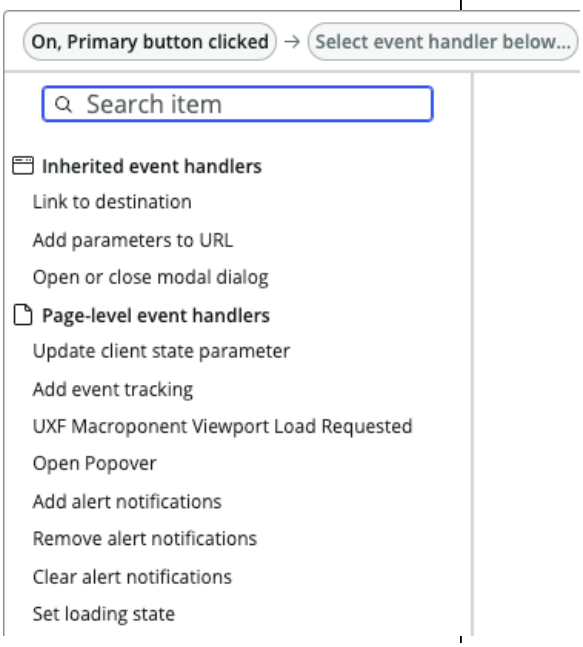
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Modal	Configuration
	on whether you want the modal to close based on the default action. ▪ Enable or disable **Defer modal content loading**. If you disable it, the modal loads with the page. If you enable it, the modal doesn't load when the page loads. ▪ Select **Events** > **Add event mapping** to add an event handler to the modal. ▪ Select an event handler to apply to the modal. Choose from inherited or page-level event handlers. Event handlers perform an action such as open or close a modal. Depending on the modal type, you can refresh data for the App Shell data source, the user session for GraphQL, or a user session for Transform. 
Confirm or destroy	▪ Add a header, which is the title of the modal. ▪ Write the content that you want displayed in the modal. The content tells the user what the alert is. ▪ Add text for the primary button. Primary is the main action button for users, such as **Delete** or **Erase**. ▪ Add text for the secondary button. The secondary button is usually the no choice for users, such as **Cancel**, **No**, and so on. ▪ Choose the size of the modal on the screen. Select **Small**, **Medium**, **Large**, or **Fullscreen**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

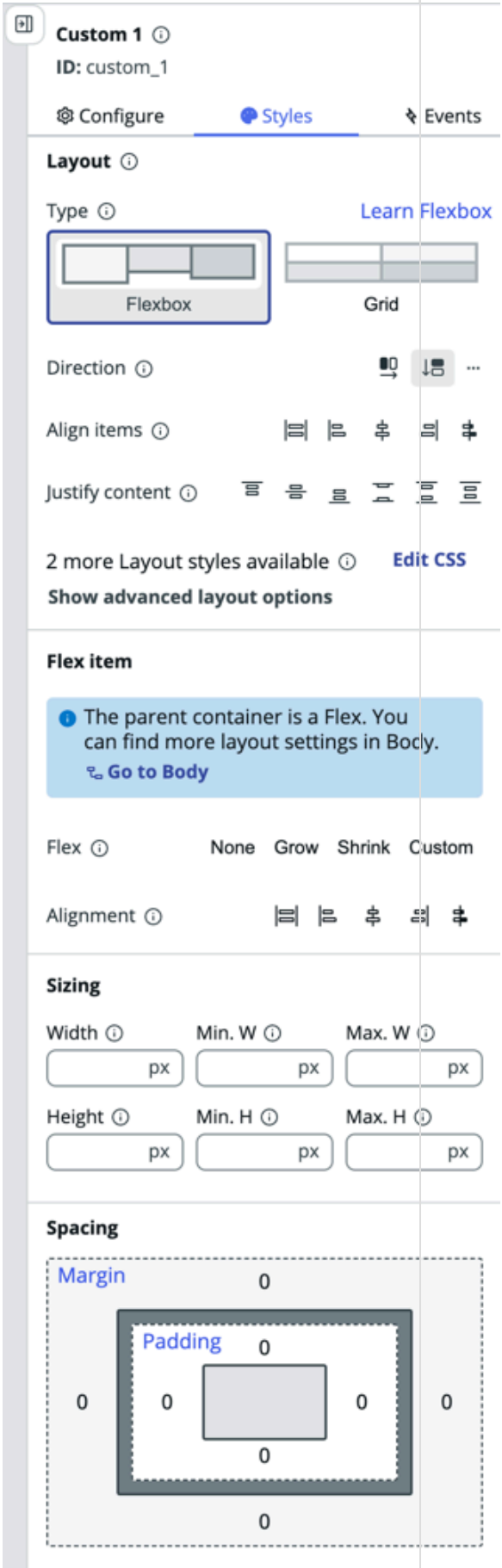
Modal	Configuration
	▪ Enable or disable **Prevent Default Primary Button Action**, depending on whether you want the modal to close based on the default action. ▪ Enable or disable **Prevent Default Secondary Button Action**, depending on whether you want the modal to close based on the default action. ▪ Enable or disable **Defer modal content loading**. If you disable it, the modal loads with the page. If you enable it, the modal doesn't load when the page loads. ▪ Select **Events** > **Add event mapping** to add an event handler to the modal. ▪ Select an event handler to apply to the modal. Choose from inherited or page-level event handlers. Event handlers perform an action such as open or close a modal. Depending on the modal type, you can refresh data for the App Shell data source, the user session for GraphQL, or a user session for Transform. 
Custom	▪ Choose a layout for your modal. You can use a flexbox or CSS grid layout. These layouts let you add content in your modal however you want. ▪ Use styling options to change how your modal looks. You can apply any standard CSS styling to your modal, such as background color and padding. For more information about styling, see Change the default appearance of components.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

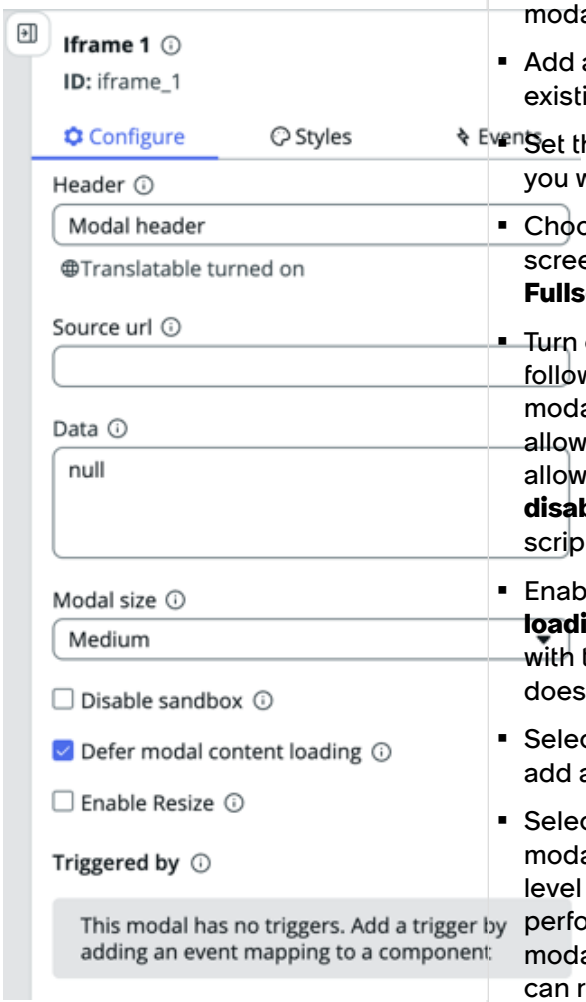
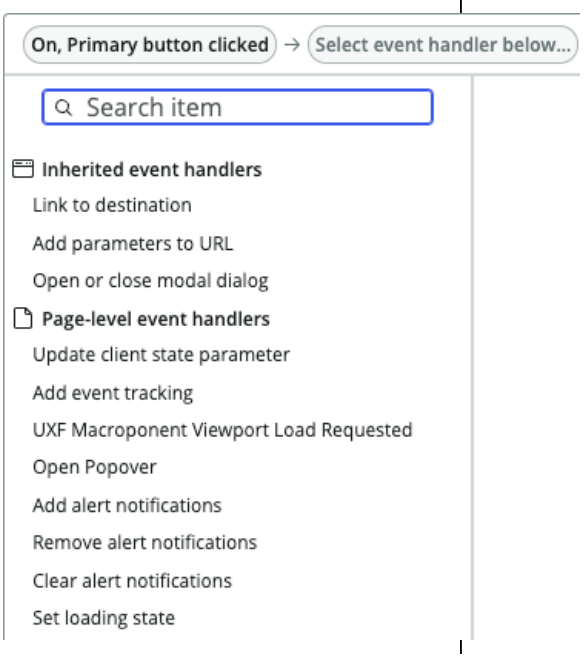
Modal	Configuration
	▪ Add components to the containers within the custom modal. ▪ Select **Events** > **Add event mapping** to add an event handler to the modal. ▪ Select an event handler to apply to the modal. Choose from inherited or page-level event handlers. Event handlers perform an action such as open or close a modal. Depending on the modal type, you can refresh data for the App Shell data source, the user session for GraphQL, or a user session for Transform. 

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

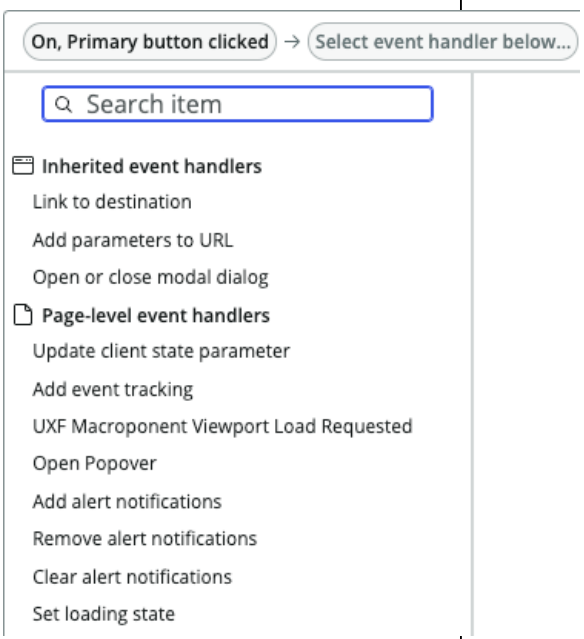
Modal	Configuration



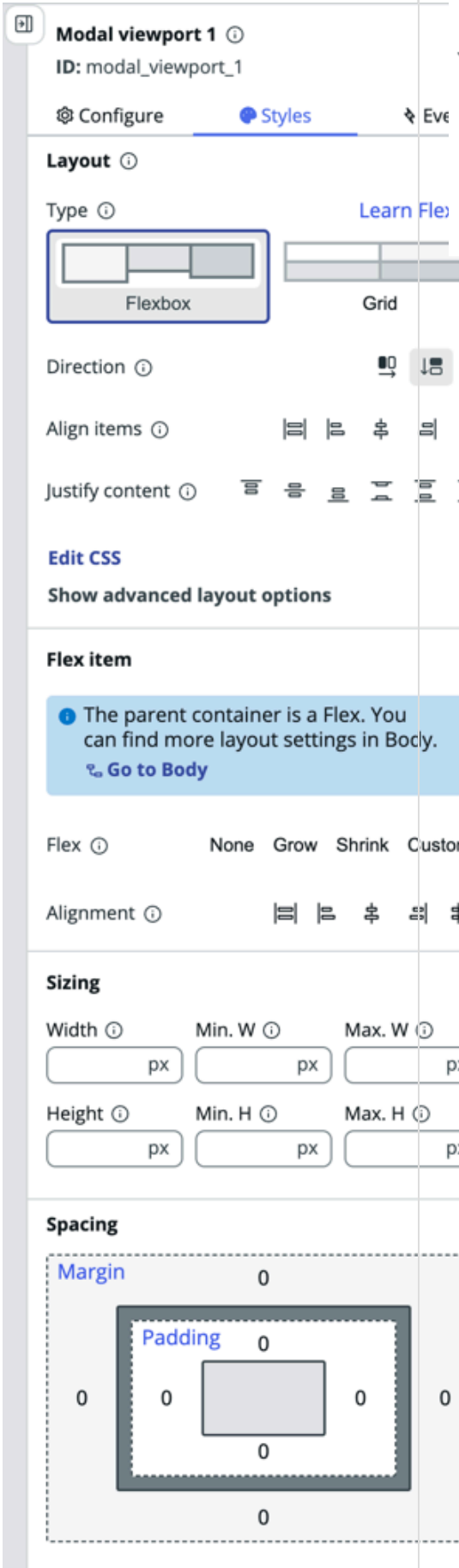
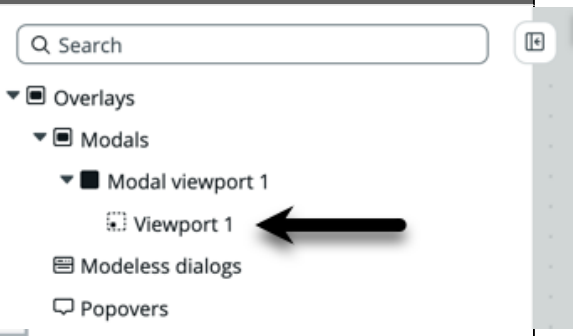
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Modal	Configuration
iframe 	▪ Add a header, which is the title of the modal. ▪ Add a source URL that points to your existing iframe content. ▪ Set the parameters and initial data that you want to iframe. ▪ Choose the size of the modal on the screen. Select **Small**, **Medium**, **Large**, or **Fullscreen**. ▪ Turn on **disable sandbox** to lift the following restrictions: allow-forms, allow-modals, allow-popups, allow-presentation, allow-same-origin, allow-scripts, and allow-downloads options. Turn off the **disable sandbox** to only lift the allow-scripts option. ▪ Enable or disable **Defer modal content loading**. If you disable it, the modal loads with the page. If you enable it, the modal doesn't load when the page loads. ▪ Select **Events** > **Add event mapping** to add an event handler to the modal. ▪ Select an event handler to apply to the modal. Choose from inherited or page-level event handlers. Event handlers perform an action such as open or close a modal. Depending on the modal type, you can refresh data for the App Shell data source, the user session for GraphQL, or a user session for Transform. 

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Modal	Configuration
Modal viewport	▪ Choose the size of the modal on the screen. Select **Small**, **Medium**, **Large**, or **Fullscreen**. ▪ Enable or disable **Hide Padding**. ▪ Enable or disable **Hide Close Button**. If you enable it, the close button does not display in the modal. ▪ Enable or disable **Defer modal content loading**. If you disable it, the modal loads with the page. If you enable it, the modal doesn't load when the page loads. ▪ Select **Events** > **Add event mapping** to add an event handler to the modal. ▪ Select an event handler to apply to the modal. Choose from inherited or page-level event handlers. Event handlers perform an action such as open or close a modal. Depending on the modal type, you can refresh data for the App Shell data source, the user session for GraphQL, or a user session for Transform.  ▪ Select the viewport component within the modal viewport.

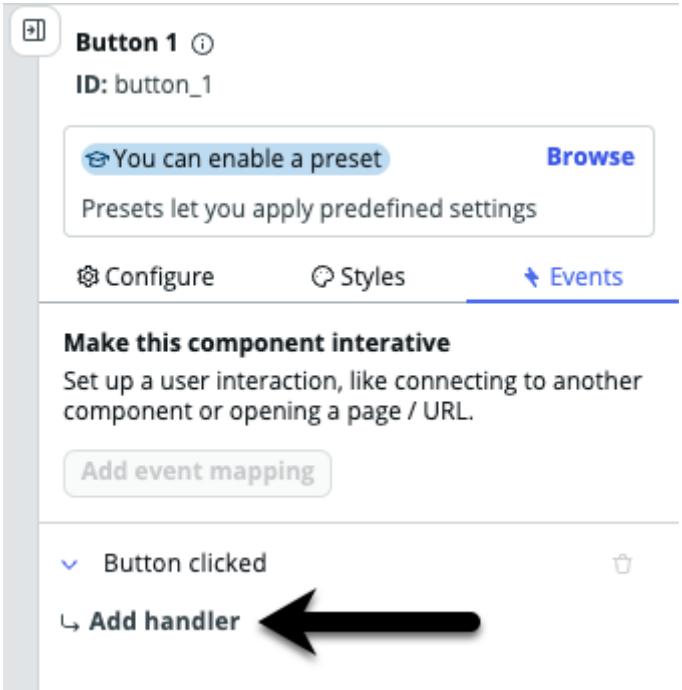
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Modal	Configuration
	

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

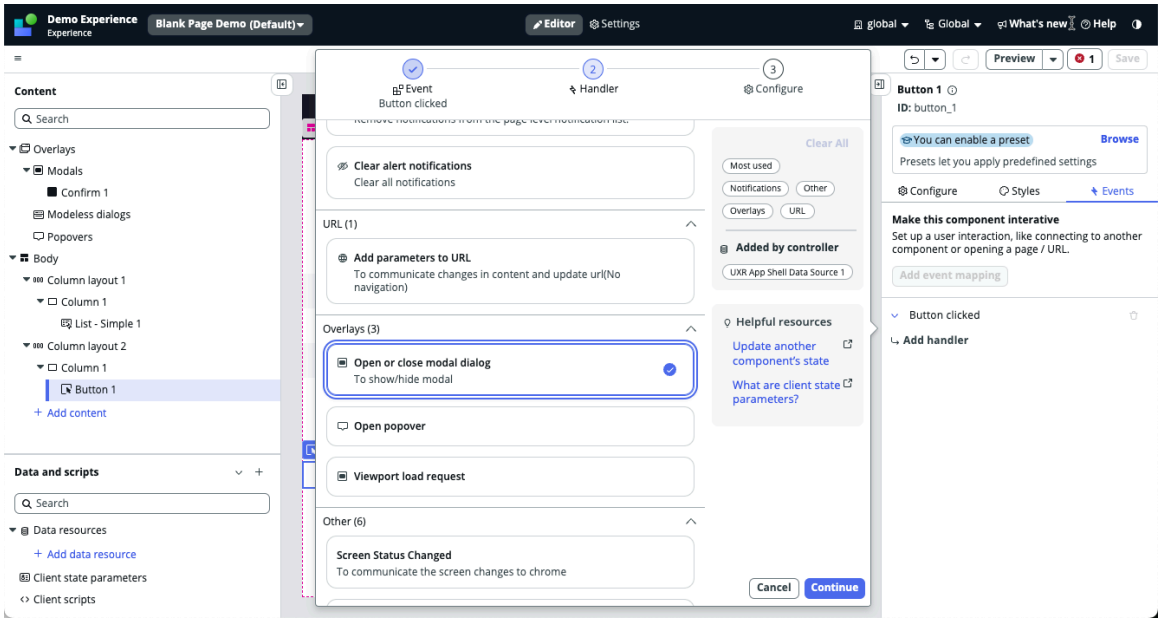
Modal	Configuration
	▪ Select **+ Add** in the Configure tab to add a page collection to the viewport. ▪ Select a page collection from the list and click **Add**.

5. Add a component to your page to trigger the modal you just added, such as a button component.
   See Add and configure components for more information.
6. Select the **Events** tab in the configuration panel.
7. Select **+ Add event handler**.



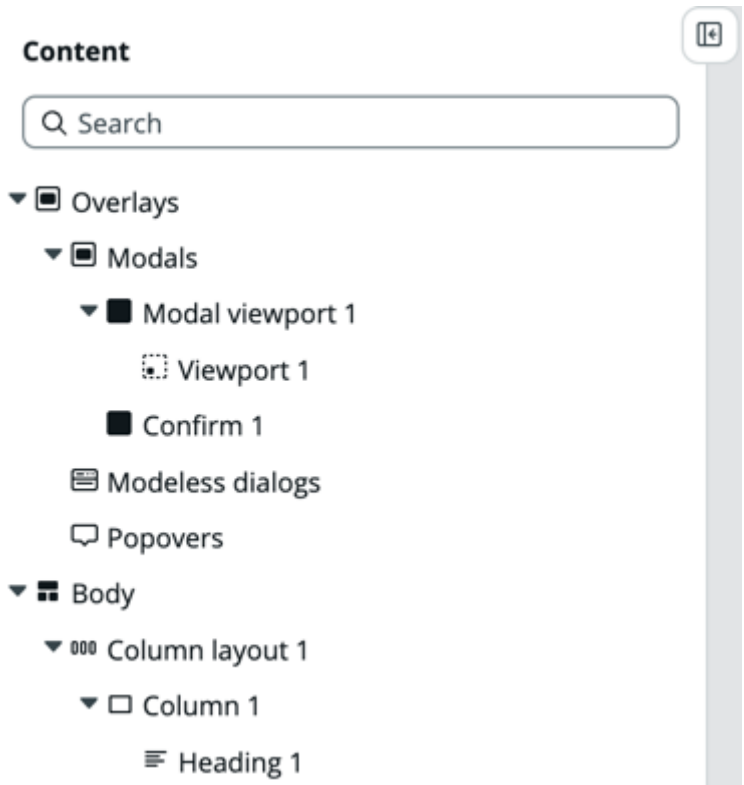
8. Select **Open or close modal dialog**.
9. Select the modal you created in the previous steps using the **Modal** drop-down.
10. Click **Add**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



11. When you finish configuring the modal, close it.

Notice in the content tree that the modals you create sit above the body of your page structure.



12. Click **Save**.

Extend your UI experience with viewport components

Viewports are specialized components that enable you to extend your experience without needing to own the parent page in UI Builder. You can work with viewports in three ways. You can add a viewport component or a viewport-enabled tab to a page, or add a viewport to the Contextual sidebar component.

Add a viewport component to your page to create separate content even if you do not own the content of the page itself. Create subpages, specify audiences, or customize content with

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

different data, routes, and screens. For more information about viewport components, see Add a viewport component to your page.

You can replace Tabs components with Viewport-enabled tabs as a way to display third-party custom data, assign audiences, and create variants. For more information about viewport-enabled tabs, see Replace a tab with a viewport-enabled tab.

Use the **Edit content** mode of UI Builder to add viewports or tab sets to a page or Contextual sidebar component. Edit tabs in a Contextual sidebar component as you would on a page using the **Config** panel. The main difference is, when you are in the **Edit content** mode of a tab, the page management area changes to **Tab management**.

Select the menu icon ⋮ to edit your tab settings, or change the required or optional parameters.

Add viewport modals to your experience to embed subpages or other experiences within a modal in your parent page/experience. For more information about adding viewport modals to your experience, see Add a viewport modal to your experience.

Add a viewport component to your page

Add a viewport component to your page and create a subpage to create separate content on the page.

Before you begin

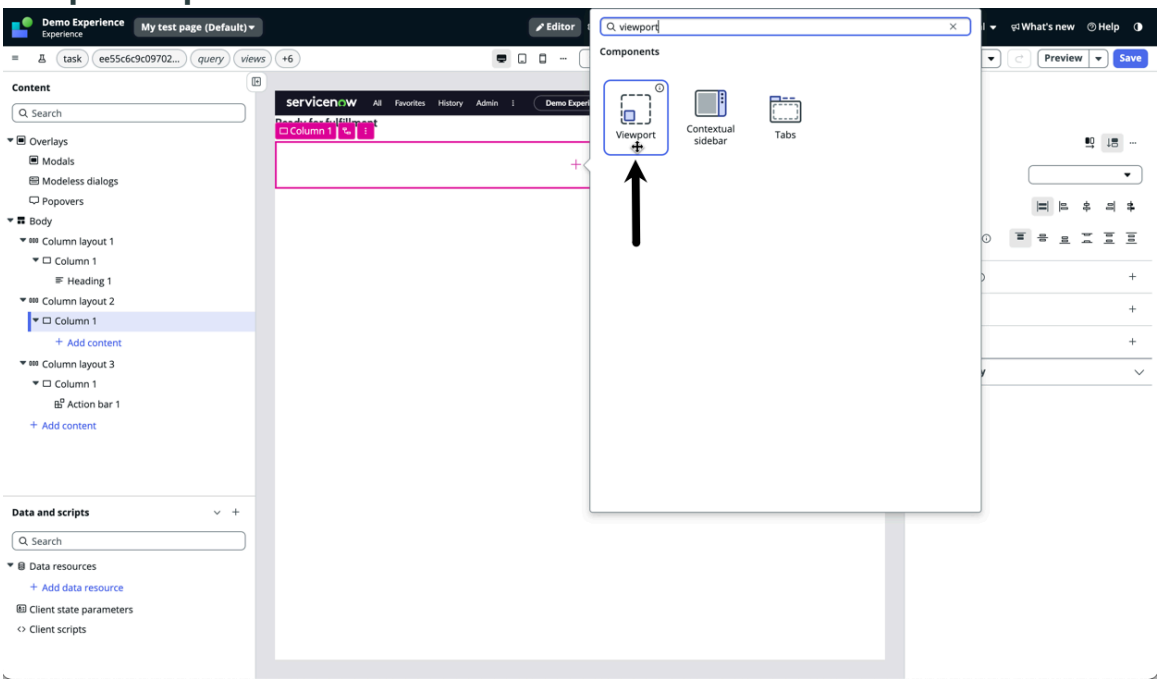
Role required: ui_builder_admin

Procedure

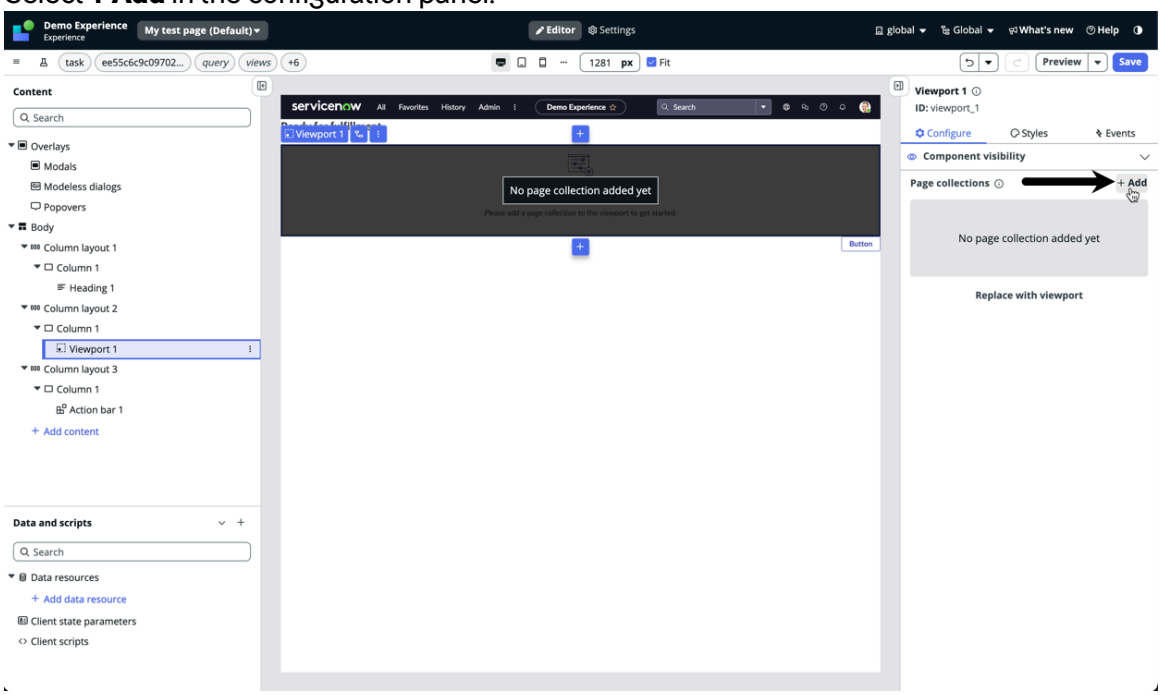
1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Open the editor for the page variant that you want to add the viewport to. If you haven't created a page for your experience, follow the steps to Create a page in UI Builder.
4. Select **+ Add content** in the content tree.
5. Select a **Single column** layout.
6. Next, select the **+** icon in the column to open the toolbox.
7. Enter `viewport` in the search box.
8. Select **Viewport**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Viewport component

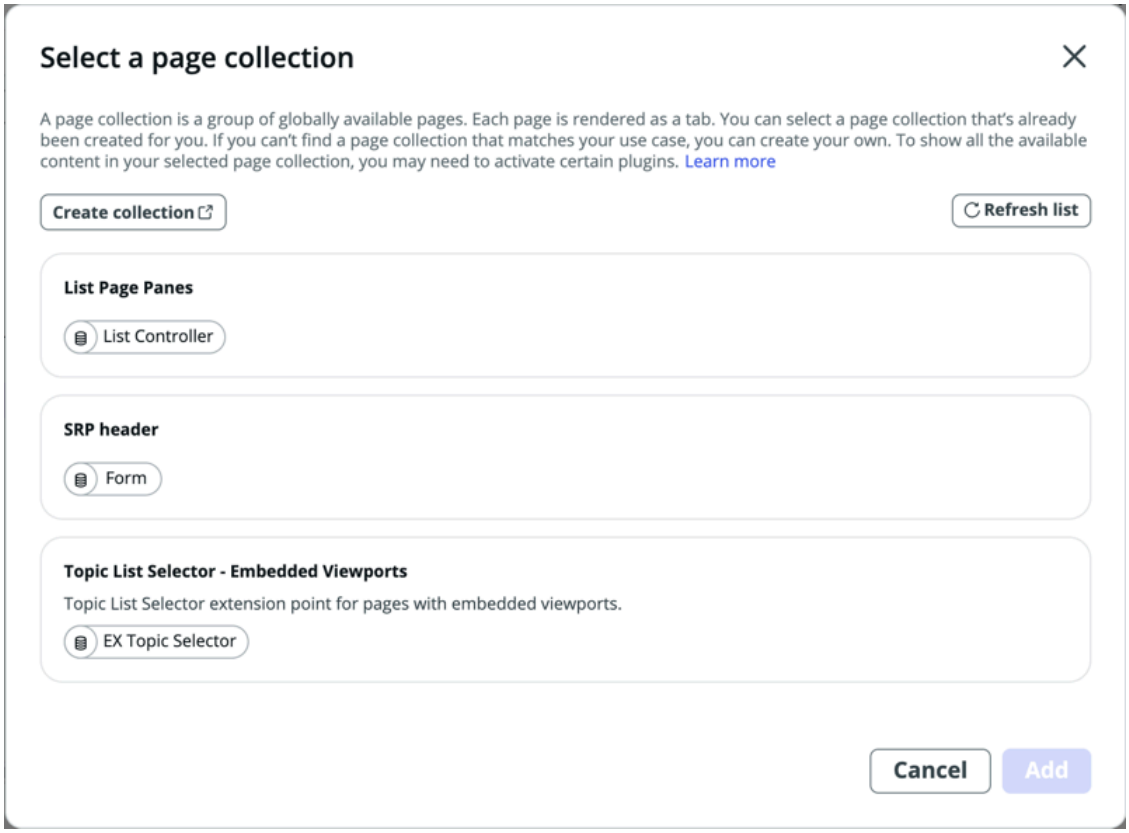


9. Select the viewport component in the content tree.
10. Select **+ Add** in the configuration panel.



11. Select a page collection from the list or create a collection by selecting **+ Create collection**. For more information on creating your own page collection, see Create a page collection across multiple UI pages.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



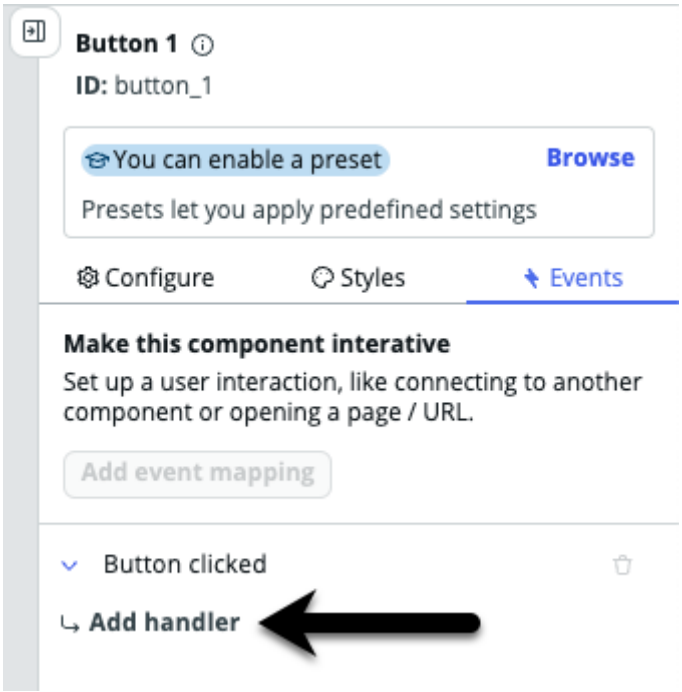
12. Select **Add**.
    The page collection displays in the **Page collections** section of the configuration panel. Only one page of the page collection displays in the staging area.
13. Select **Save**.
14. Locate the viewport ID and route.

    You can see the ID and route in the Config panel as shown in the following figure.

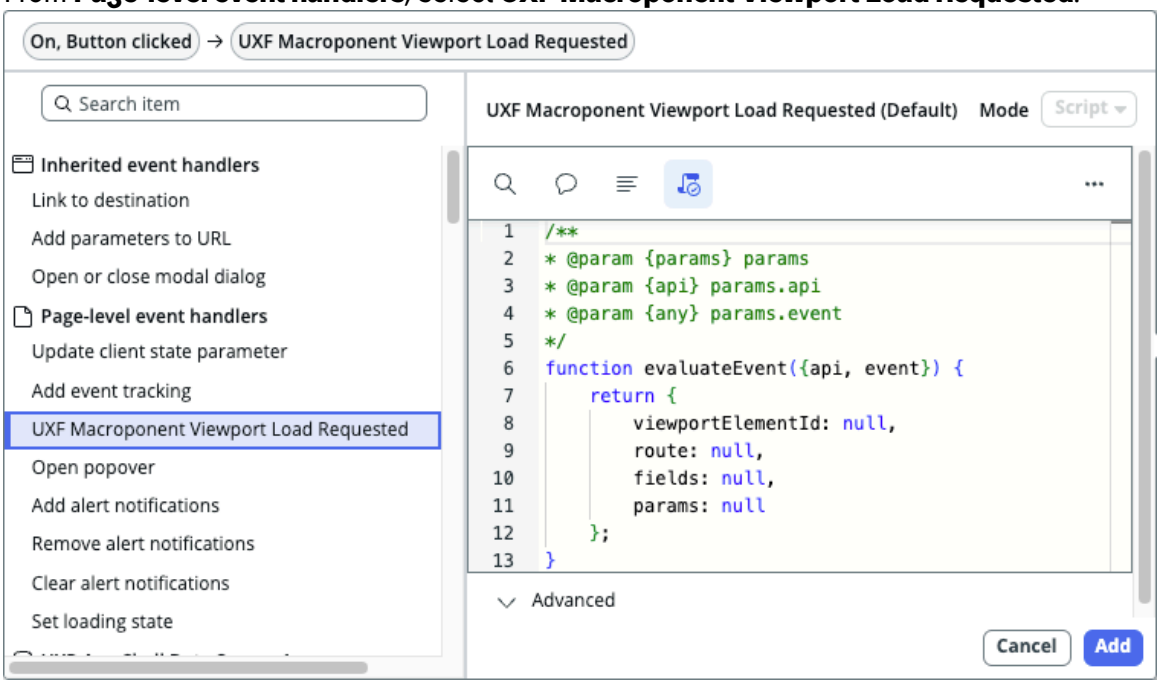


15. Add a component to your page to open the viewport you just added, such as a button component.
    For more information, see Add and configure components.
16. Select the **Events** tab in the configuration panel.
17. Select **+ Add event handler** to view the viewport.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



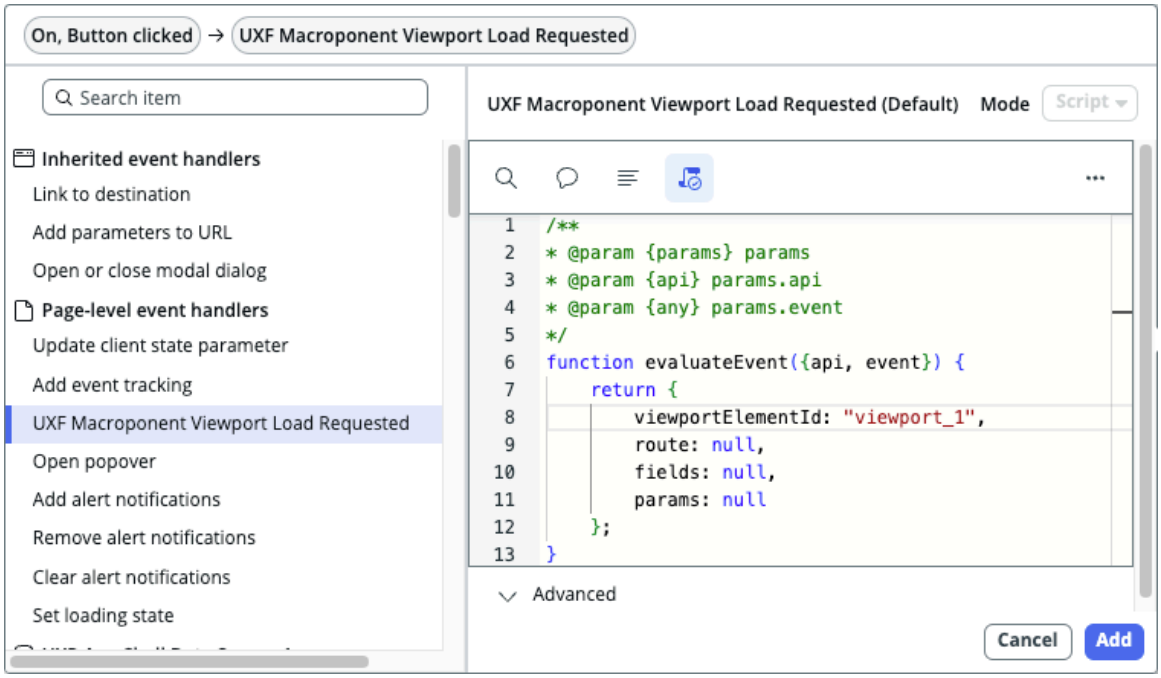
18. From **Page-level event handlers**, select **UXF Macroponent Viewport Load Requested**.



19. In the **viewportElementID** element, replace the `null` value with the viewport ID that you located in a previous step.
In this example, it is, `"viewport_1"`.

20. Select **Add**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



21. Select **Save**.
22. View and test your page by selecting Preview.

Add a viewport modal to your experience

Add a viewport within a modal in your experience.

Before you begin

Role required: ui_builder_admin

About this task

Use viewport modals to embed subpages or other experiences within a modal in your parent page or experience. Viewport modals can be opened via events or scripts. Viewport modals are limited to one subroute per viewport modal. You can create additional viewport modals for extra routes.

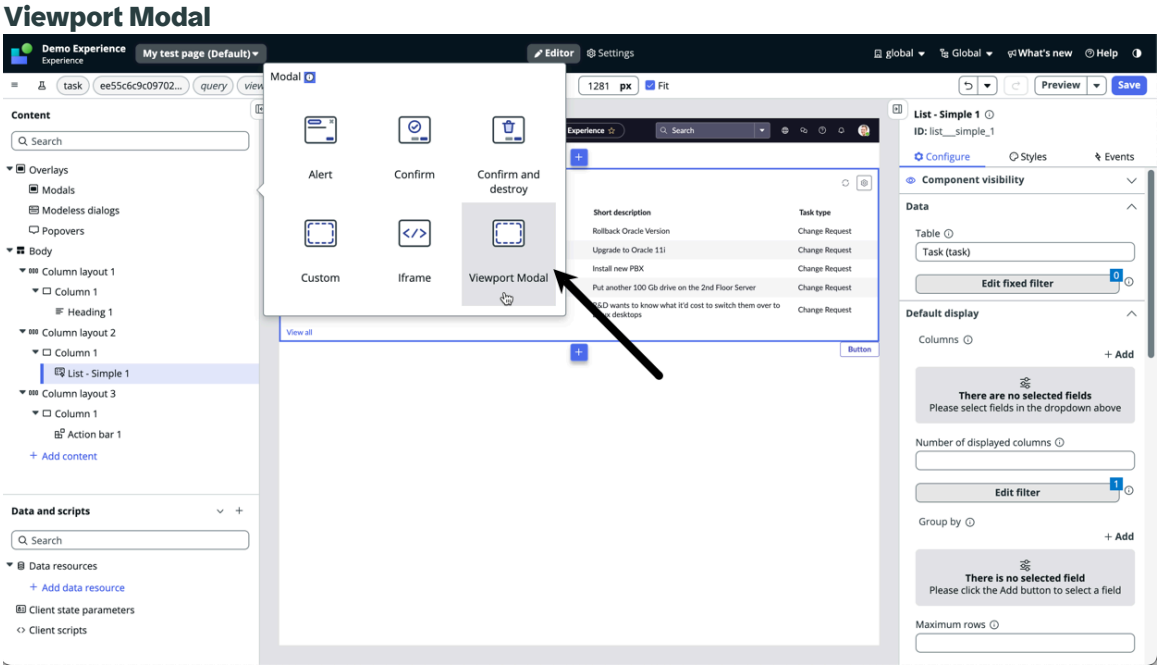
Note: Legacy viewport modals cannot be upgraded and must be recreated to take advantage of the new functionality.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. See Configure how users interact with your applications in UI Builder for more information.
3. Open or create a page.
   If you open an existing page, ensure you are in the same scope as the original page. If not, change the scope before you start editing the page. Application scoping protects applications by identifying and restricting access to application files and data. Administrators set the scope to specify what parts of an application are accessible to other applications. Application scope protects data and application files. See Learn about security and roles for more information on application scope.
4. Select **+ Add content** in the content tree.
5. Select a **Single column** layout.
6. Select **+** next to **Modals** in the content tree.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

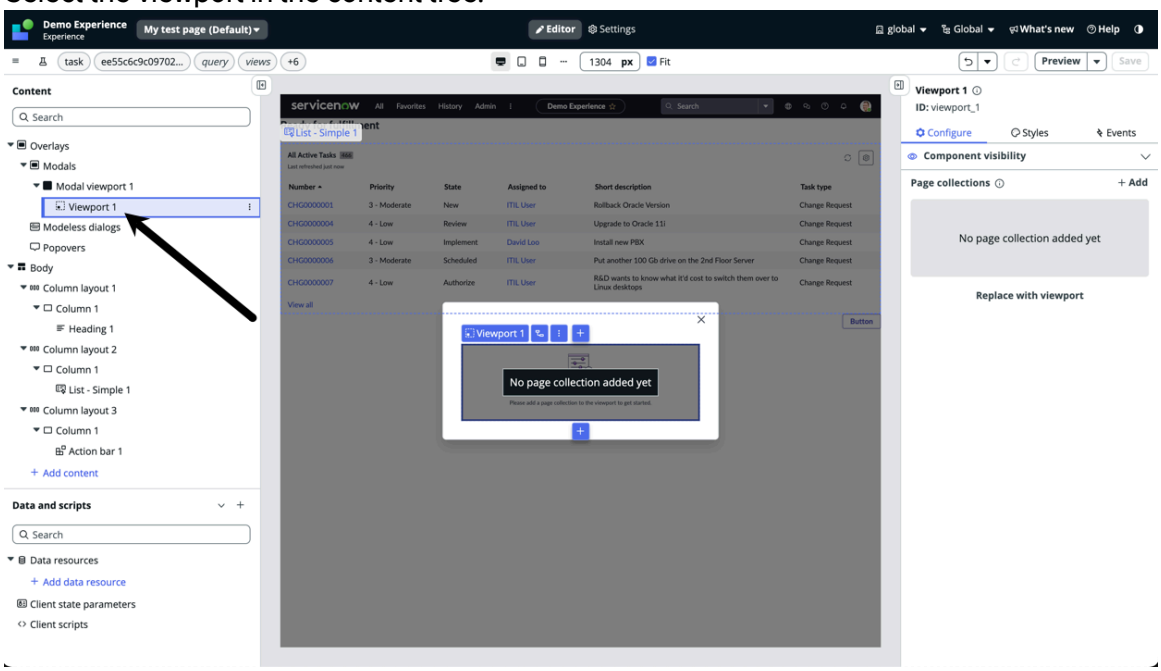
7. Select **Viewport Modal** in the list.



A viewport model appears above your page.

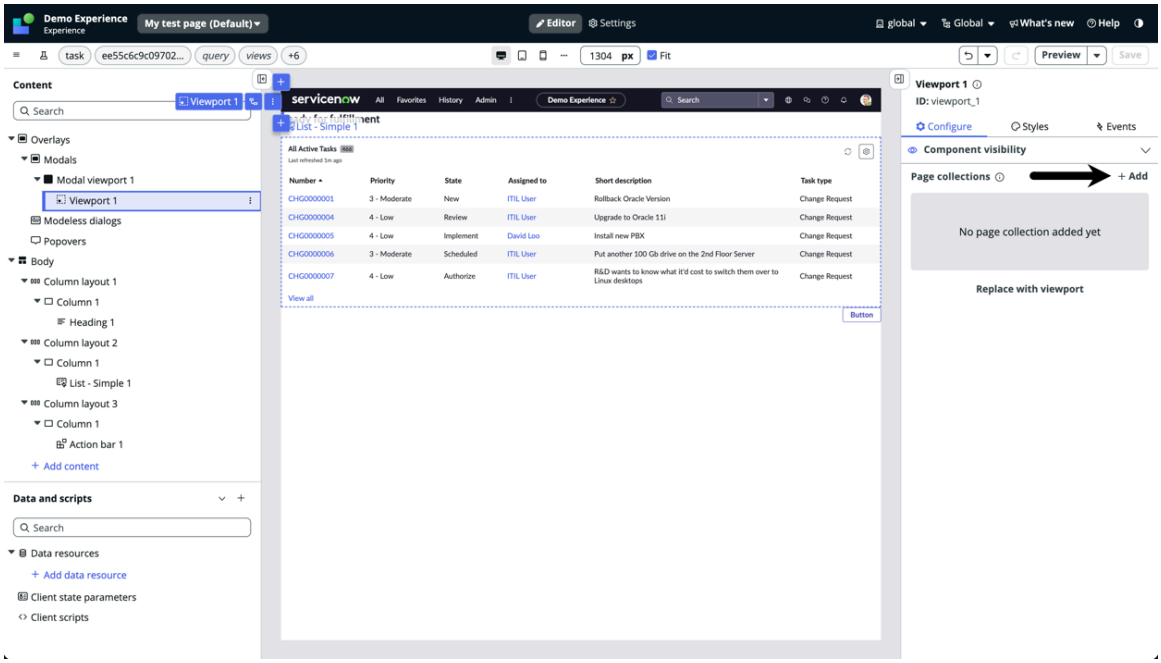
8. Click **Save**.

9. Select the viewport in the content tree.

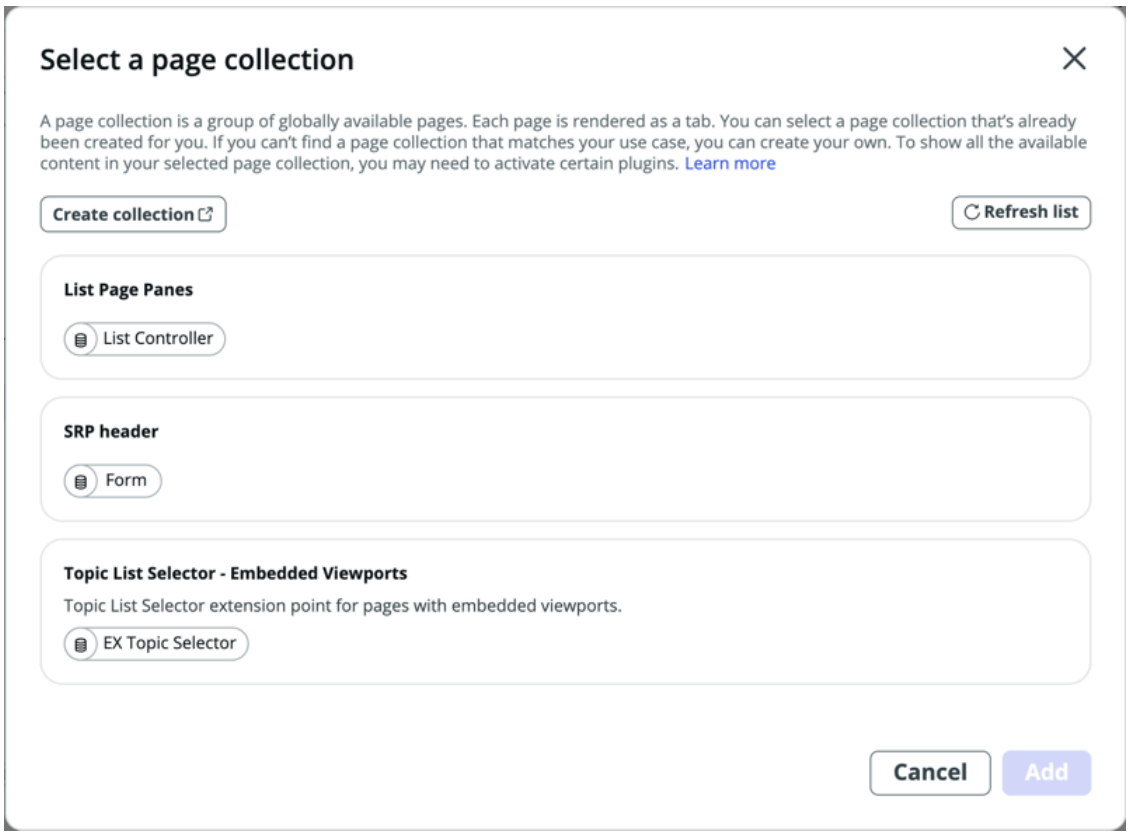


10. Select **+ Add** next to **Page collections** in the configure tab.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



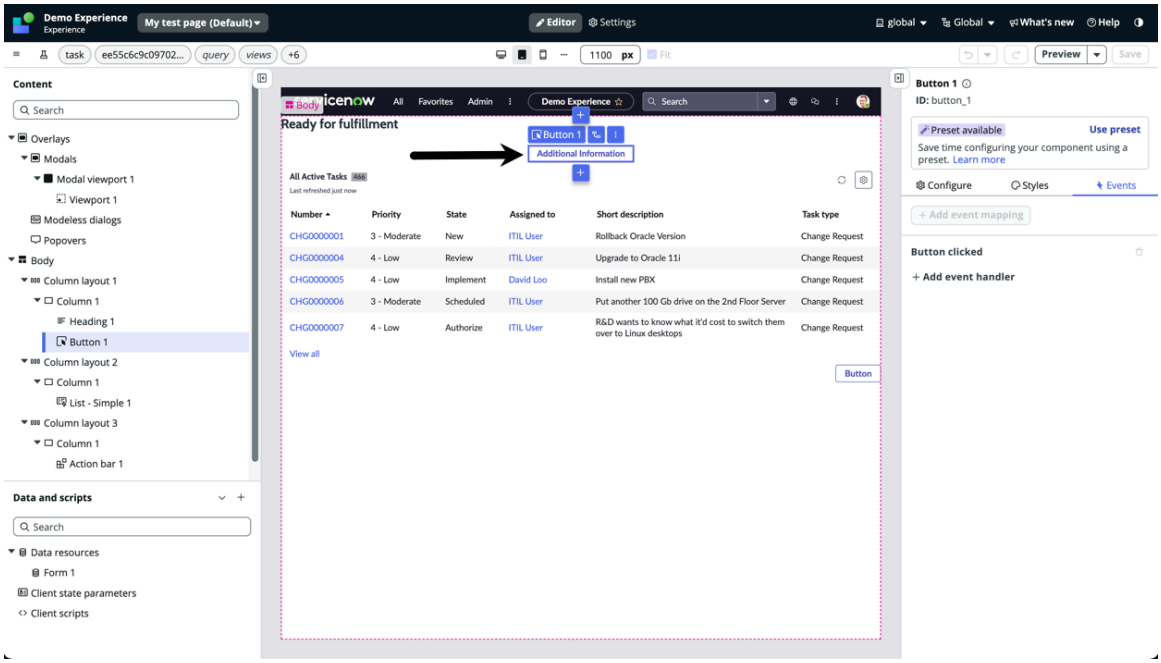
11. Select a page collection or create a new one.
    For more information, see Create a page collection across multiple UI pages.



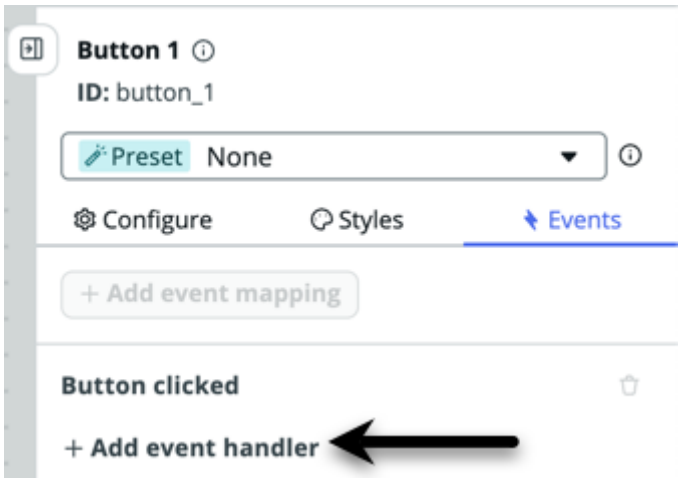
12. Click **Add**.
13. Click **Save**.
14. Add a component that opens the viewport modal.

    The following example uses a button to open the viewport modal.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

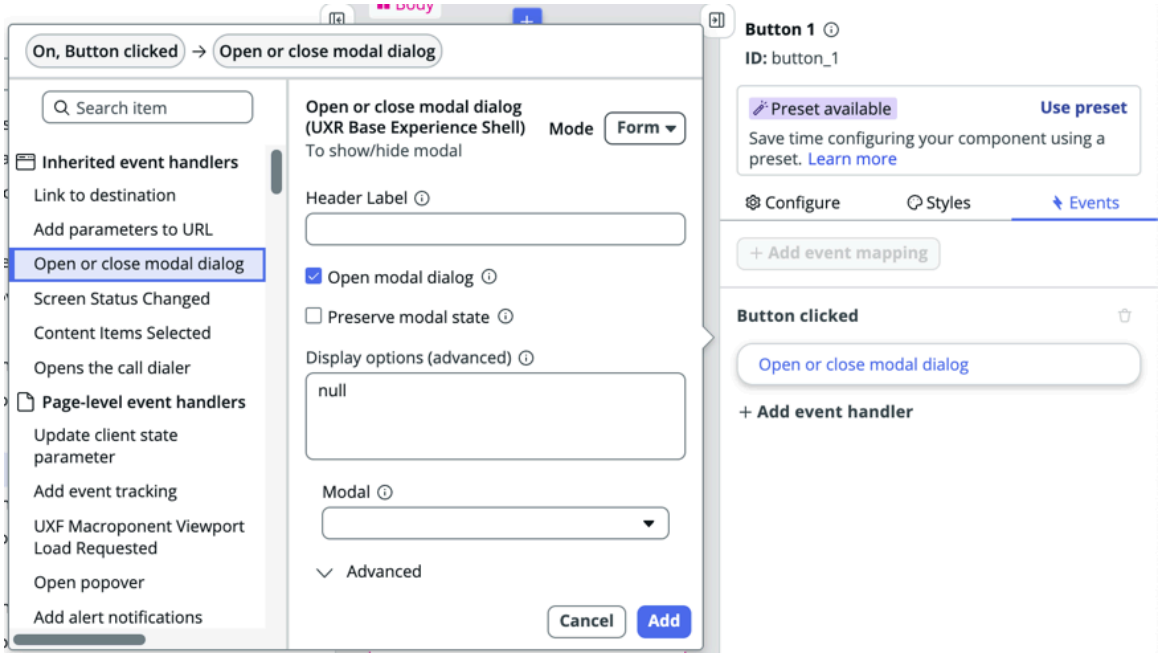


15. Select the **Events** tab.
16. Select **+ Add event handler**.



17. Select **Open or close modal dialog**.
18. Enable **Open modal dialog**.
19. Select the viewport modal that you created in the **Modal** dropdown.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



The **Viewport ID** auto populates.

20. Select **Add**.
21. Select **Save**.
22. View and test your page by selecting Preview.

Replace a tab with a viewport-enabled tab

Convert a tab on a page to a viewport-enabled tab. Use viewport-enabled tabs to display third-party custom data, assign audiences, and create variants.

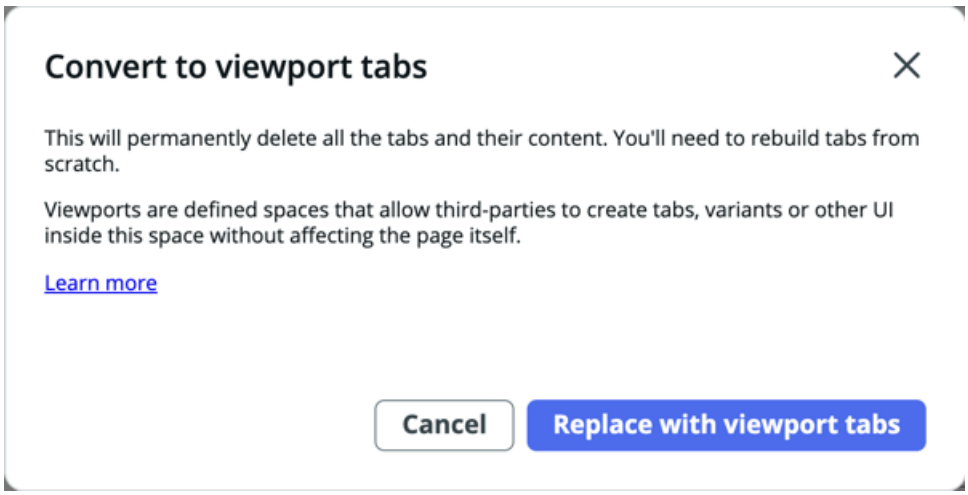
Before you begin

Role required: admin

About this task

Replace your tab or tabs with viewport-enabled tabs.

Note: Replacing a tab with a viewport-enabled tab is a permanent and one-way process. You lose any existing tab content and must recreate the viewport tab content.

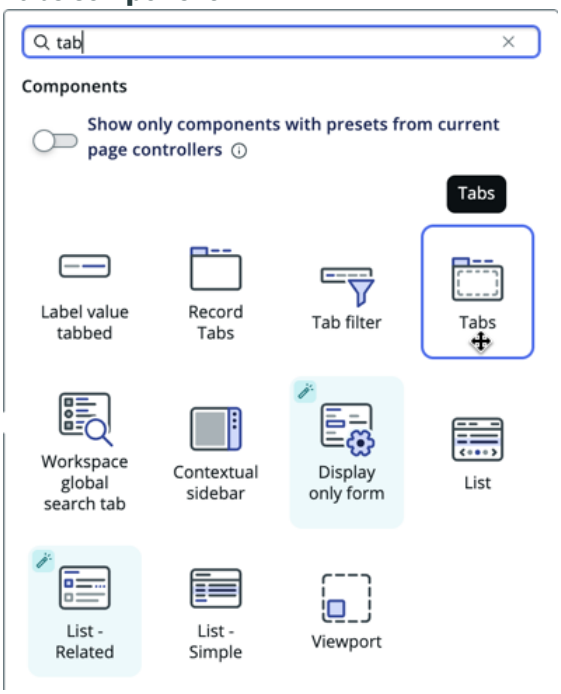


© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create or open a page with existing tabs, or add a **Tabs** component to your page.

Tabs component

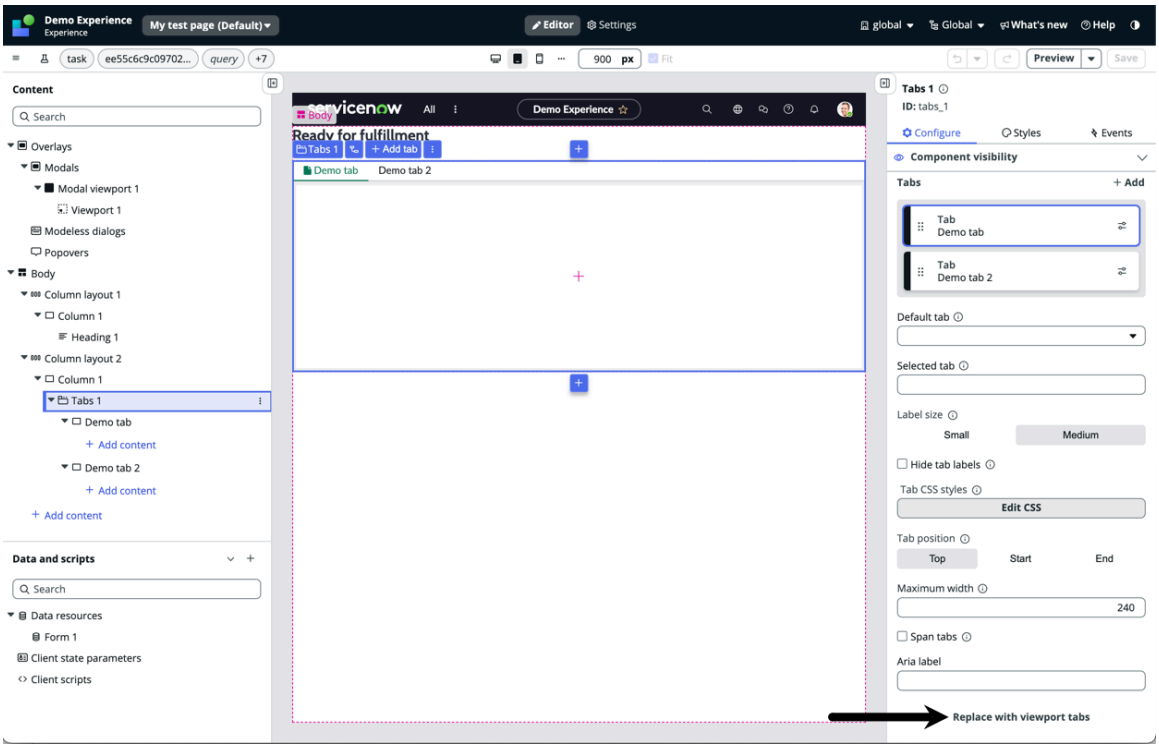


For more information on how to add a Tabs component to a page, see Add tabbed content to UI Builder pages.

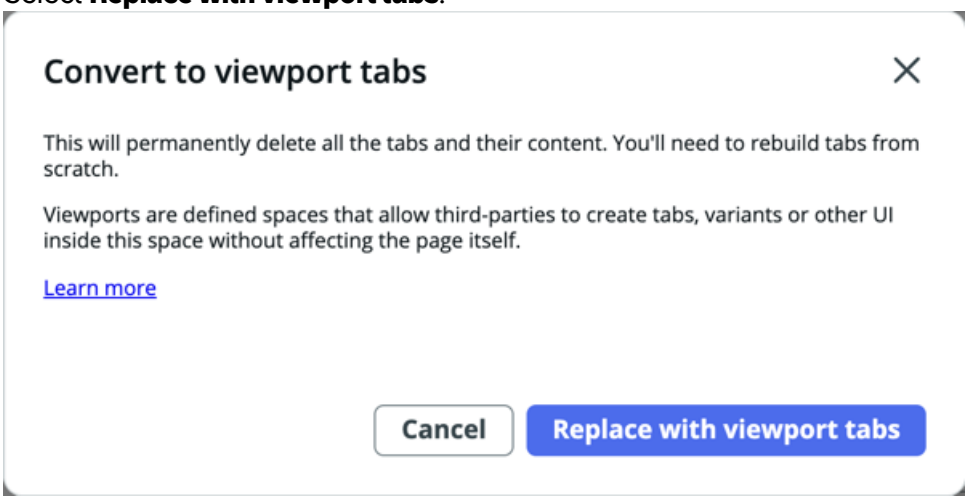
4. Select the tabs component that you want to replace with viewport tabs in the content tree.
5. Select **Replace with viewport tabs** in the configuration panel.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: Replacing a tab with a viewport-enabled tab is a permanent and one-way process. You lose any existing tab content and must recreate the viewport tab content.

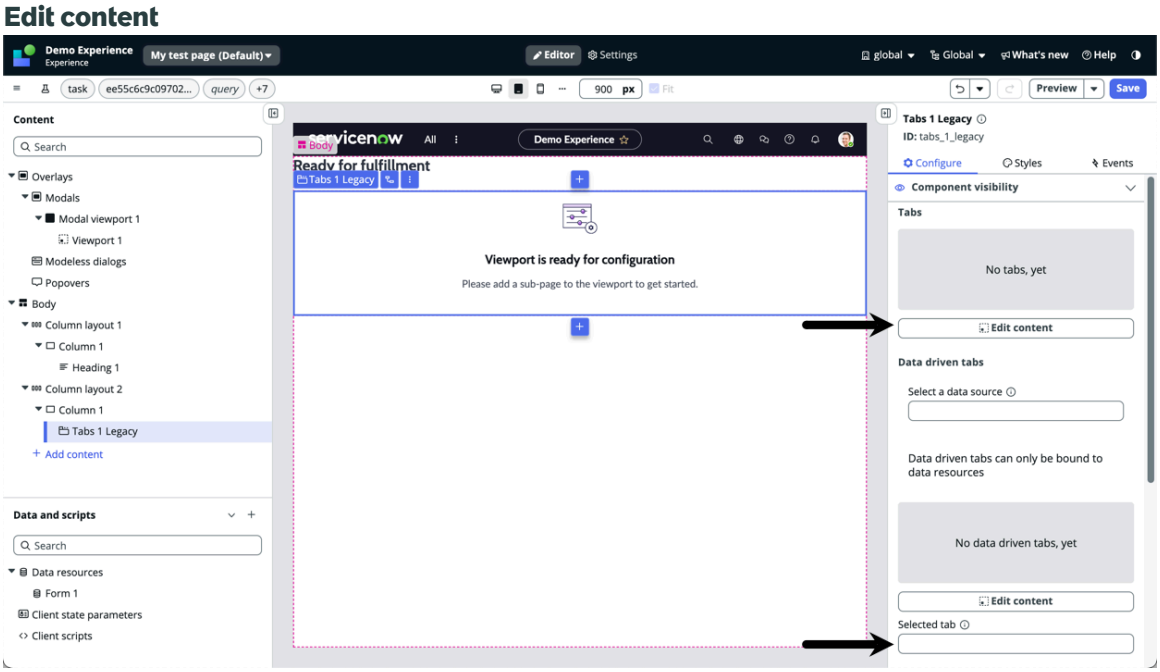


6. Select **Replace with viewport tabs**.



7. Click **Edit content** to add a viewport-enabled tab.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

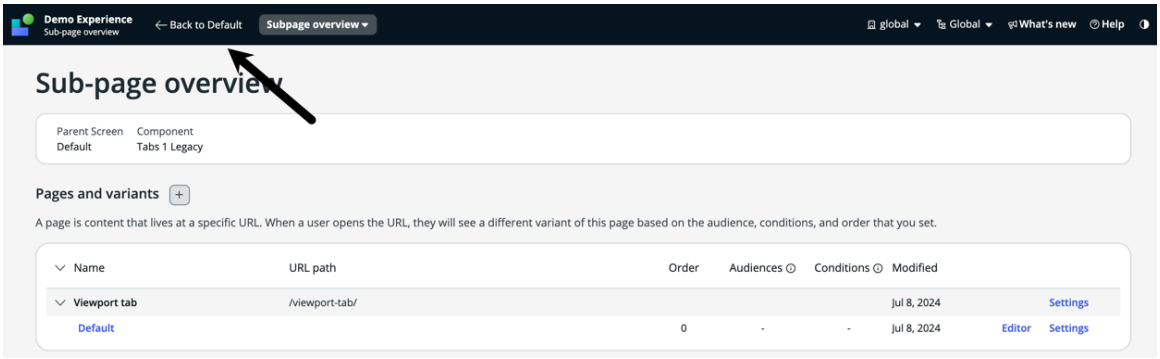


For more information on viewport components, see Add a viewport component to your page.

8. Select **Save and continue**.
   The sub-page overview page appears.
9. Select **Create new page.**
10. In the **Name** field, enter a name for your viewport-enabled tab.
11. Keep the default path, or change it to your preference.
12. Select **Continue**.
13. In the **Name** field, enter a name for the page variant or leave it as **Default**.
14. Add one or more audiences for this page.
    If an audience you need is not listed, you can choose the **View all available audiences** to create one.
15. **Optional:** Declare conditions for when to display the page or tab by either using the provided dropdown menus or writing an encoded query string.

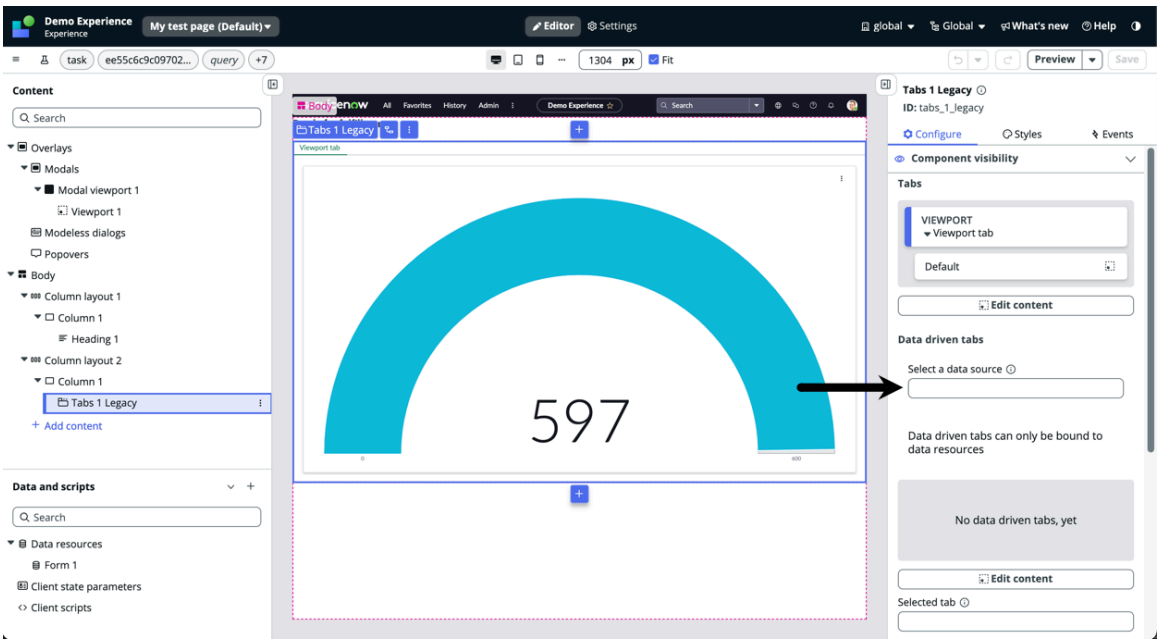
    (Optional) For more information on writing encoded queries, see Encoded query strings.
16. Select **Open in editor**.
17. Add components to your tab as you would to a page.
18. Add an event handler to any components to make them perform actions, such as loading page content.
    For more information about event handlers, see Manage actions in UI Builder pages.
19. Select **← Back to (name of page)** to return to your original page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

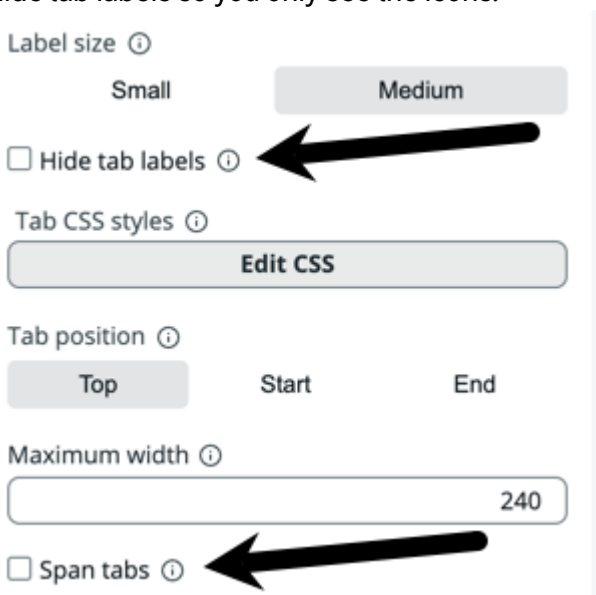


20. In **Data driven tabs**, select a data source to bind data to your viewport-enabled tab using data resources to dynamically expose your data from tables and records.

You then bind these data properties to components in your tab. For more information about using data resources, see Dynamically expose data in UI Builder pages (advanced feature).



21. Choose where the viewport-enabled tab labels appear on the page and decide whether to hide tab labels so you only see the icons.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Create popovers in UI Builder

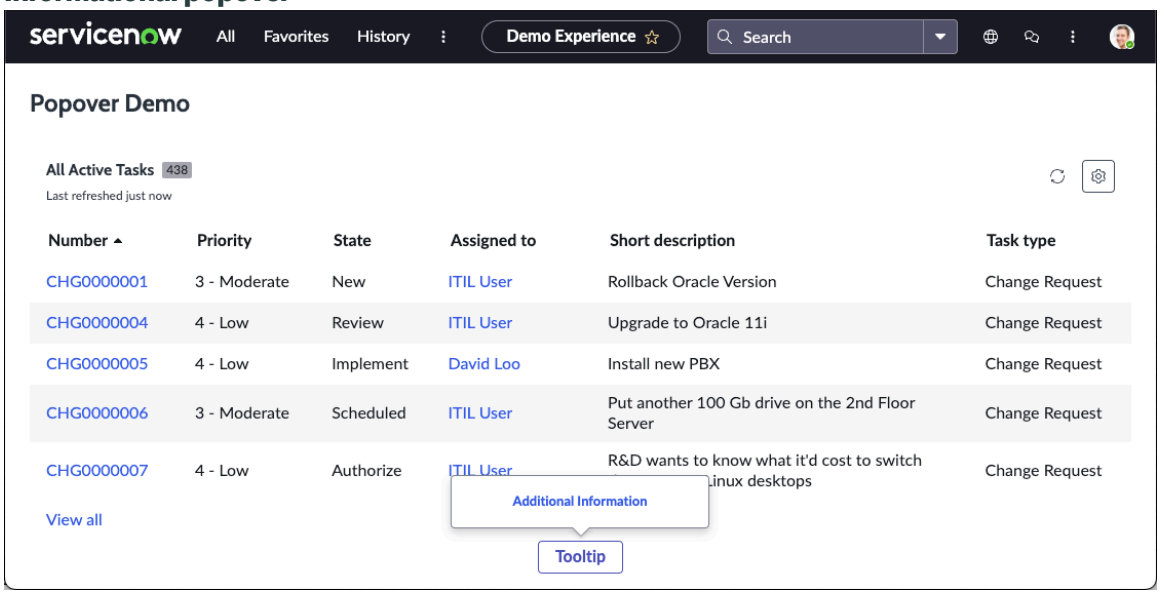
Use popovers on a UI Builder page to overlay contextual information or functionality to help users complete tasks.

A popover is a small window or dialog box that appears above a UI Builder page and contains additional information, options, or actions related to the content or task at hand. Add components to a popover in the same way you add modals to a page.

You can place popovers anywhere on a UI Builder page where you think additional information helps the users. Popovers are intended to provide small pieces of information or links to related content, so you should limit the amount of information or functionality within a popover because the popover only displays when a user is interacting with it.

You can make popovers visible or hidden with event mapping, such as triggering a popover to appear when selecting a button or pointing to a part of the page. For more information, see Define map events.

Informational popover



Add popover to a UI Builder page

Learn how to add a popover in UI Builder. A popover is a container that appears above a page when you click a component. For example, a popover might display contact information when selecting a persons name in a list.

Before you begin

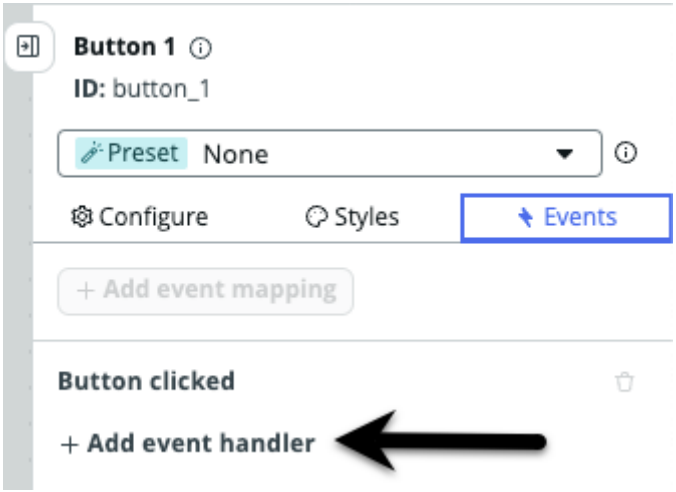
Role required: admin

Procedure

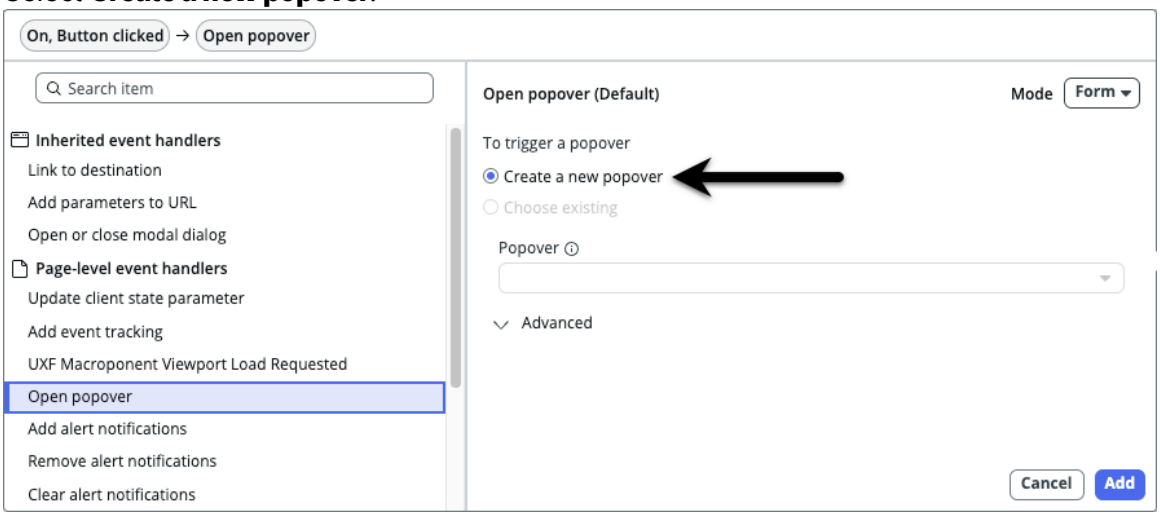
1. Navigate to **All** > **Now experience framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Open or create a page variant.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. Add a component to your page that you want to trigger a popover, such as a button component.
   See Add and configure components for more information.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

5. Select the **Events** tab in the configuration panel.
6. Select **+ Add event handler**.

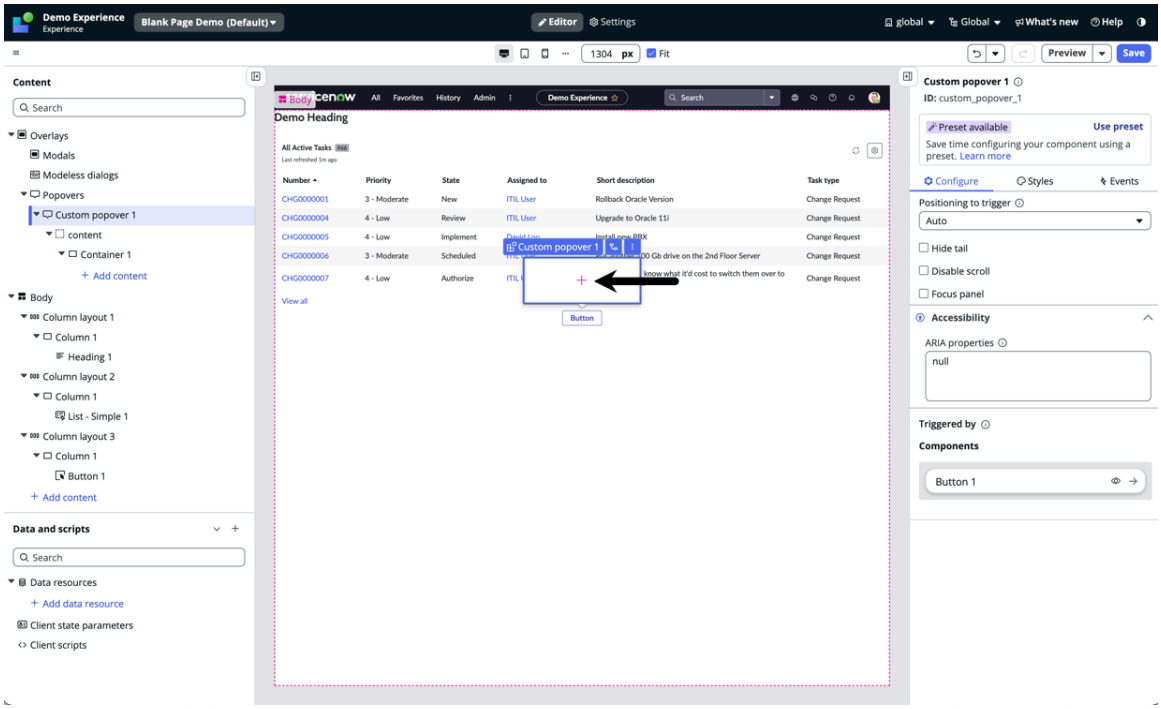


7. Select **Open Popover**.
8. Select **Create a new popover**.



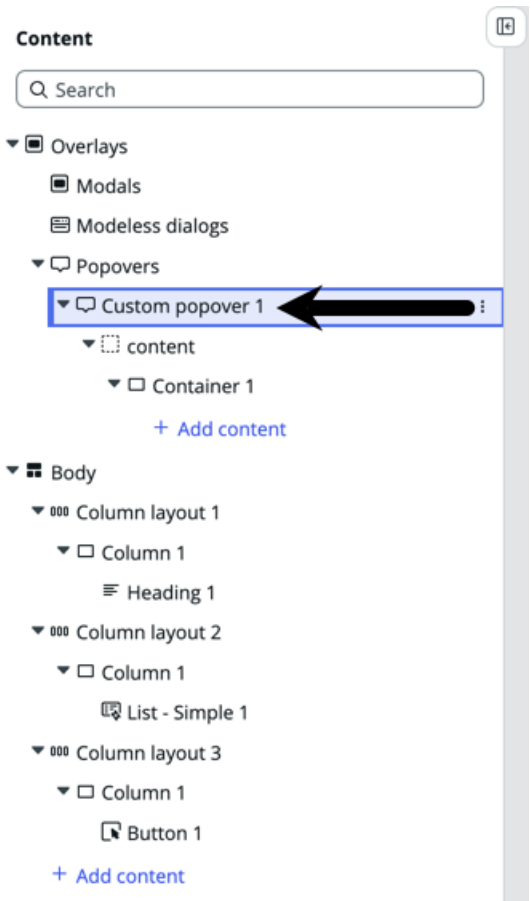
9. Select **Add**.
   The popover appears above the stage.
10. Add components to the popover by selecting the **+** icon.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



11. When you finish configuring the popover, close it.

Notice in the content tree that the popovers you create sit above the body of your page structure.



12. Click **Save**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

13. Select **Preview** in the UI Builder header.
14. Click on the button to test the popover.

Create modeless dialogs in UI Builder

Use modeless dialogs on a UI Builder page to add a floating window that enables you to interact with both the window content and the page content below.

The Modeless Dialog component is a floating window containing components and content that appears over a UI Builder page. You can do actions in the modeless dialog while still viewing and interacting with the page below. Multiple modeless dialog windows can be added to a single page. Modeless dialogs can be resized, moved to a new position on a page, and minimized.

Modeless dialogs are different than modals, which pop up but block interaction with the page until the modal is closed. Modeless dialogs are also different than popovers, which close after you click outside the popover.

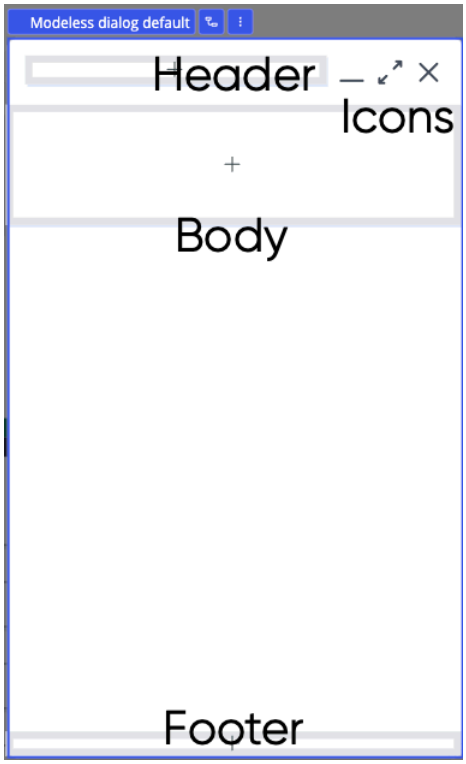
Common modeless dialog use cases include creating a record, composing an email, using chat, adding work notes or comments, reading help information, or adding/viewing attachments in the modeless dialog.

Anatomy of a modeless dialog

A modeless dialog consists of three parts:

- When you add a modeless dialog to a page, the header includes a preconfigured minimize icon, expand icon, and close icon. Other actions can be configured if needed, for example, configure an action in a button to open the modeless dialog in a new tab.
- Add any component from the UI Builder toolbox, for example, Email composer, Attachments, or Agent chat, into the body. You can apply layouts and configure the component as if you were adding the component to a page.
- If needed, provide additional buttons or content in the footer at the bottom of the modeless dialog.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Add modeless dialog to a UI Builder page

Learn how to add a modeless dialog in UI Builder. A modeless dialog is a floating window containing content above a page.

Before you begin

Role required: ui_builder_admin

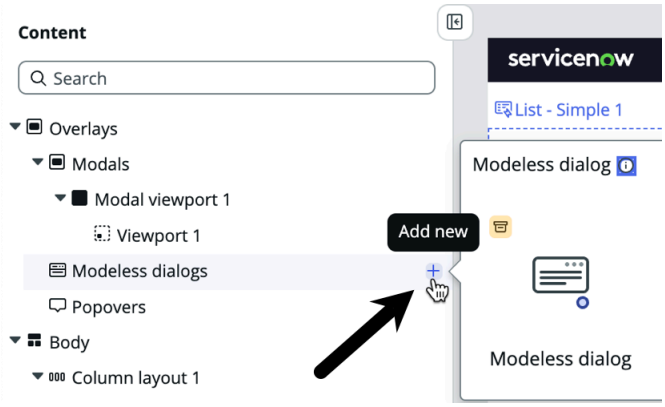
The following steps walk you through the process of configuring a button to open a modeless dialog containing the Attachments component.

Note: The procedure outlined here is just one example of how to use modeless dialogs. There are infinite possibilities. Add and configure modeless dialogs to meet your business needs.

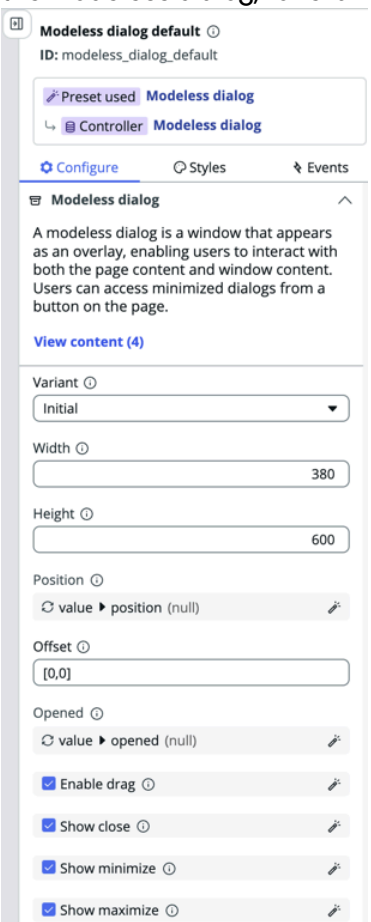
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Open or create a page variant.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. In the content tree, move your mouse to **Modeless dialogs**, select the + icon, and select the **Modeless dialog** component.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



5. If the modeless dialog doesn't appear on the stage automatically, select the new modeless dialog named **Modeless dialog default** in the content tree.
6. In the configuration panel **Configure** tab, view the presets that were added automatically with the modeless dialog, for example, the width, height, position, and enable drag setting.

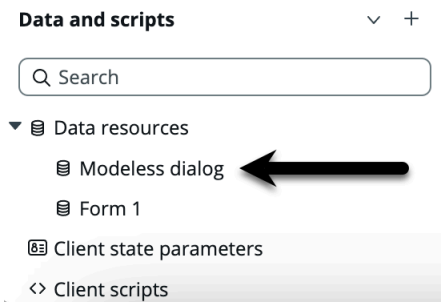


All of these preset properties can be edited, if necessary.

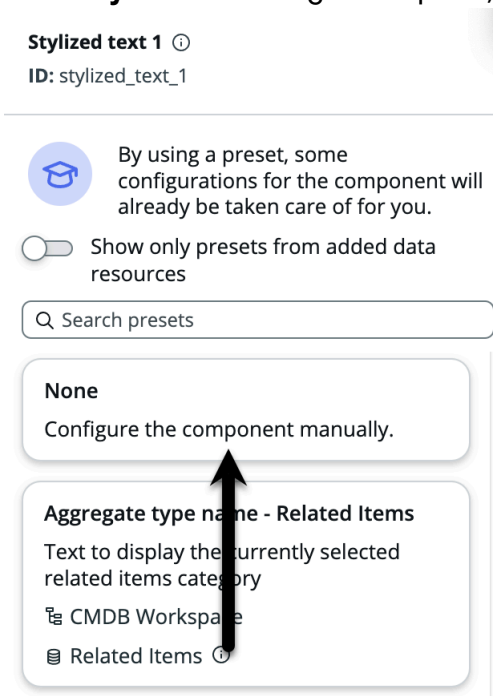
Note: For more information about Modeless Dialogs advanced properties, see the ServiceNow Developer site.

7. In the data resources drawer, the **Modeless dialog** controller was added automatically.

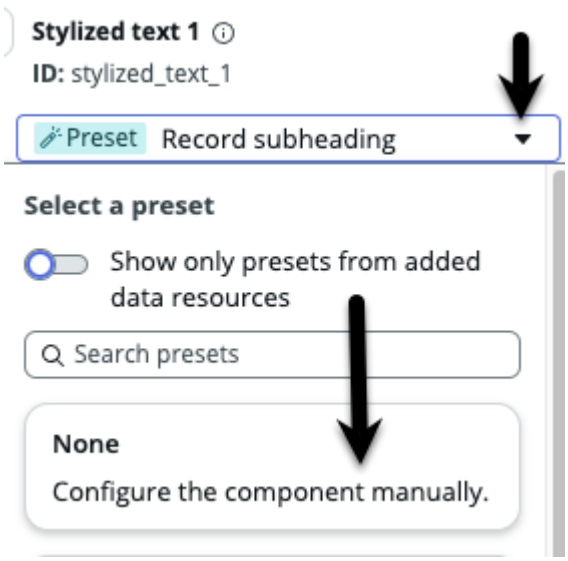
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



8. In the content tree, select **+ Add content** under **Modeless dialog default > actions** to add content to the modeless dialog header.
9. Search for and select the **Stylized text** component.
10. In the **Stylized text** configuration panel, select **None** to configure the component manually.

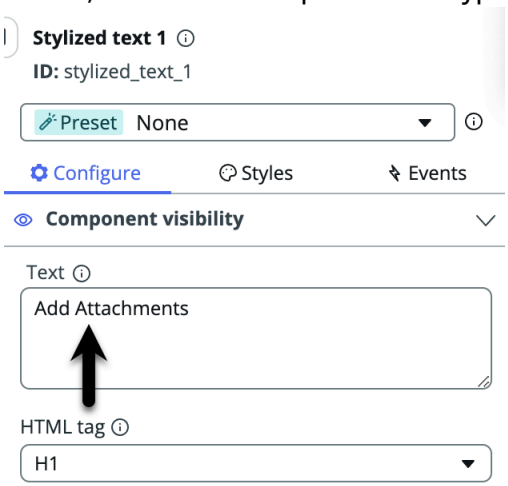


If the **Record subheading** preset was automatically added, select the drop-down arrow, select **None**, and select **Remove** on the **Configure** tab.

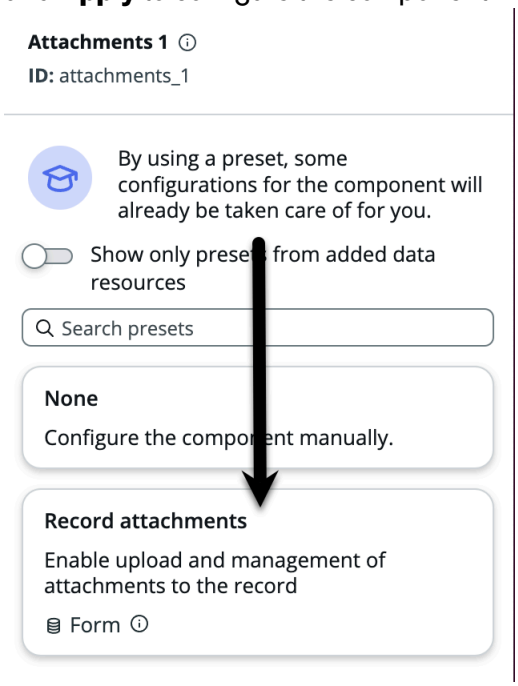


© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

11. If not open already, select **Configure** to open the configure tab.
12. In **Text**, remove the sample text and type `Add Attachments`.



13. Select **Save** in the UI Builder header.
14. In the content tree, select **+ Add content** under **Modeless dialog default > content > Container** to add content to the modeless dialog body.
15. Select the **Attachments** component.
16. In the **Attachments** configuration panel, if not selected already, select **Record attachments** and **Apply** to configure the component with a preset.

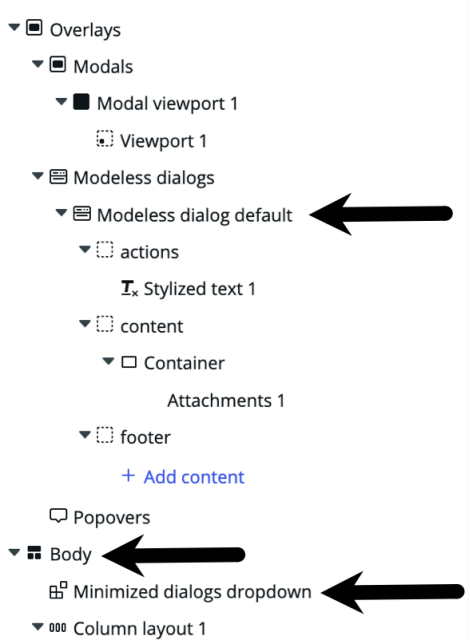


For more information about configuring the Attachments component, see the ServiceNow Developer site.

17. You could add content to the modeless dialog footer, but in this example leave the footer empty.
18. Select **Save**.
Notice in the content tree that the modeless dialog and all of its components are listed above the **Body** of your page structure. Also, the **Minimized dialogs dropdown** component is added

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

to the page automatically (and is listed under the **Body** in the content tree) to provide the functionality for the minimize icon in the modeless dialog header.

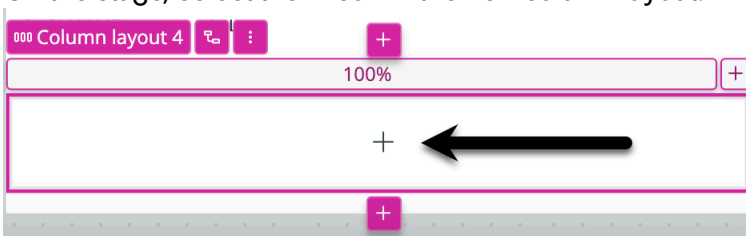


19. In the content tree, select **Body**.
20. Add a button component and configure it to open the modeless dialog.

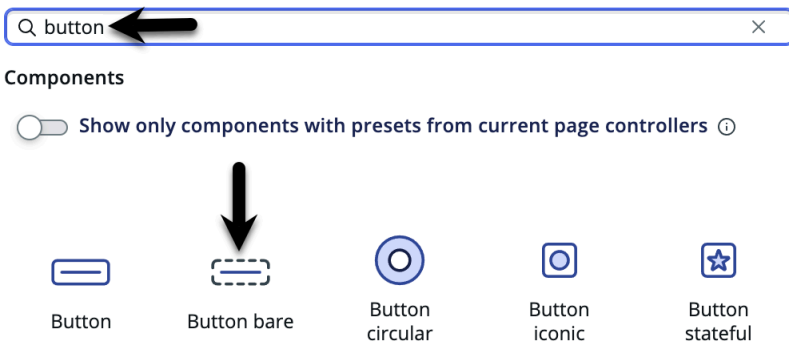
    a. In the content tree, select the **Menu** icon next to **Body** and select **Add content**.

    b. On the **Layouts** tab, select **Single column**.
    A new column layout containing a single column is added to the bottom of the page.

    c. On the stage, select the + icon in the new column layout.



    d. Search for and select the **Button bare** component.

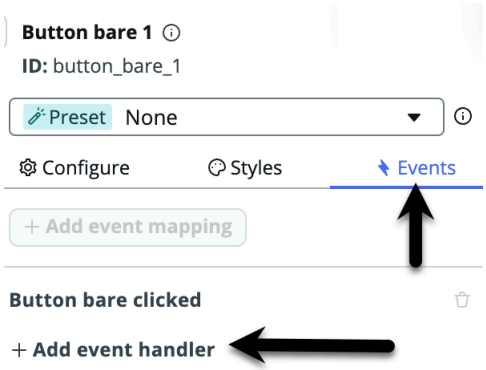


© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

e. In the configuration panel, on the **Configure** tab, select **None** to configure the component manually.

f. Select the **Events** tab in the configuration panel.

g. Select **+ Add event handler**.

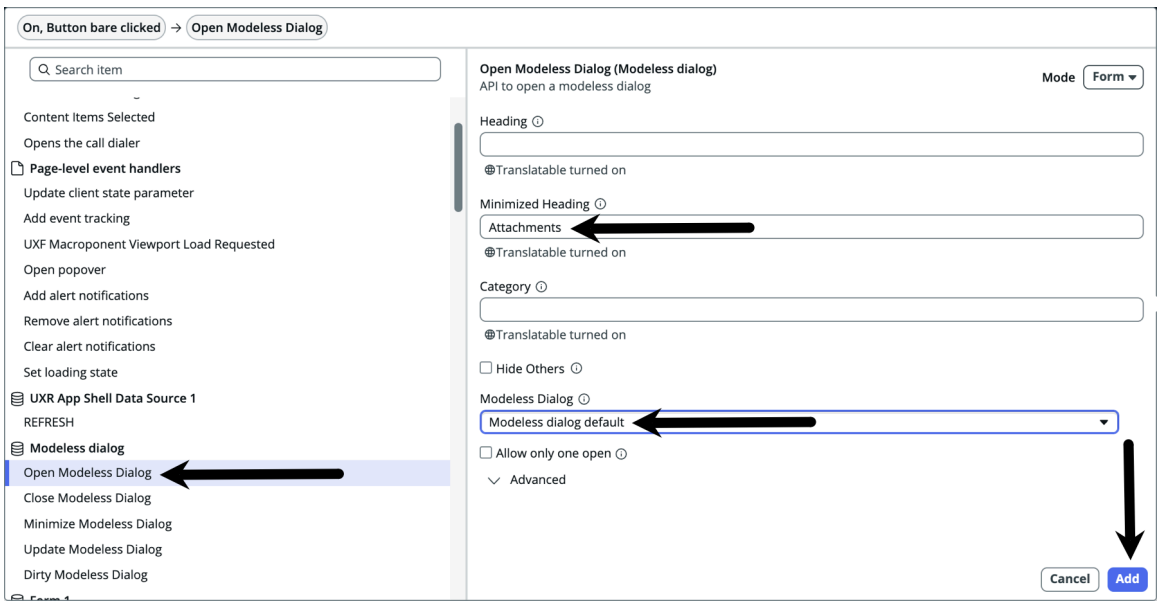


h. In the list at left, select **Open Modeless dialog** (you may need to scroll down in the list).

i. Type `Attachments` in **Minimized Heading**.

j. Select **Modeless dialog default** in **Modeless dialog**.
There are other options that can be configured here, including a heading and a category.

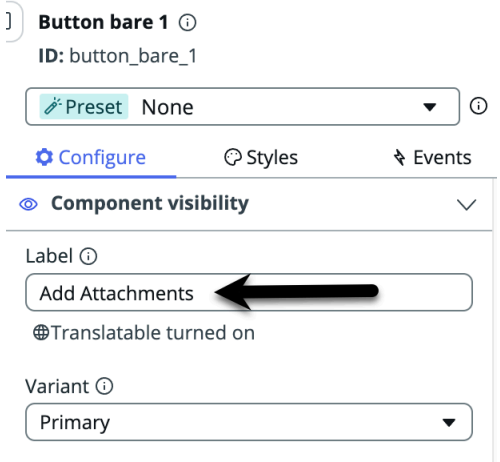
k. Select **Add**.



l. Select the **Configure** tab.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

m. Type `Add Attachments` in **Label**.



21. Select **Save**.
22. Select the drop-down arrow next to **Preview** in the UI Builder header and select **Open URL path** in the list.
23. Select the **Add Attachments** button to test the modeless dialog.
    The modeless dialog opens above the main page with the heading at the top and the Attachments component below.
24. To test the minimize functionality, select the minimize icon on the modeless dialog.



    The modeless dialog is minimized and can be accessed from the Minimized dialogs drop down.
25. Select the minimized dialogs icon and then select **Attachments** to open the modeless dialog window again.



Add modeless dialog event to a UI Builder page

Learn how to add and configure a modeless dialog event, such as open, close, or minimize in UI Builder. A modeless dialog is a floating window containing content above a page.

Before you begin

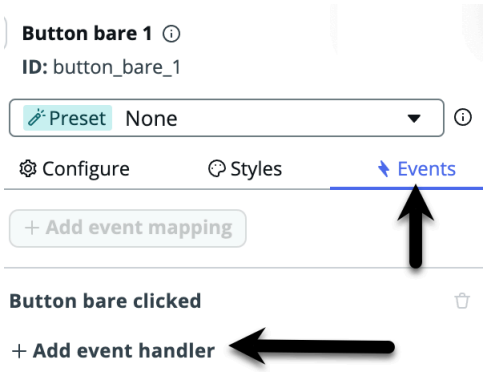
Role required: ui_builder_admin

Procedure

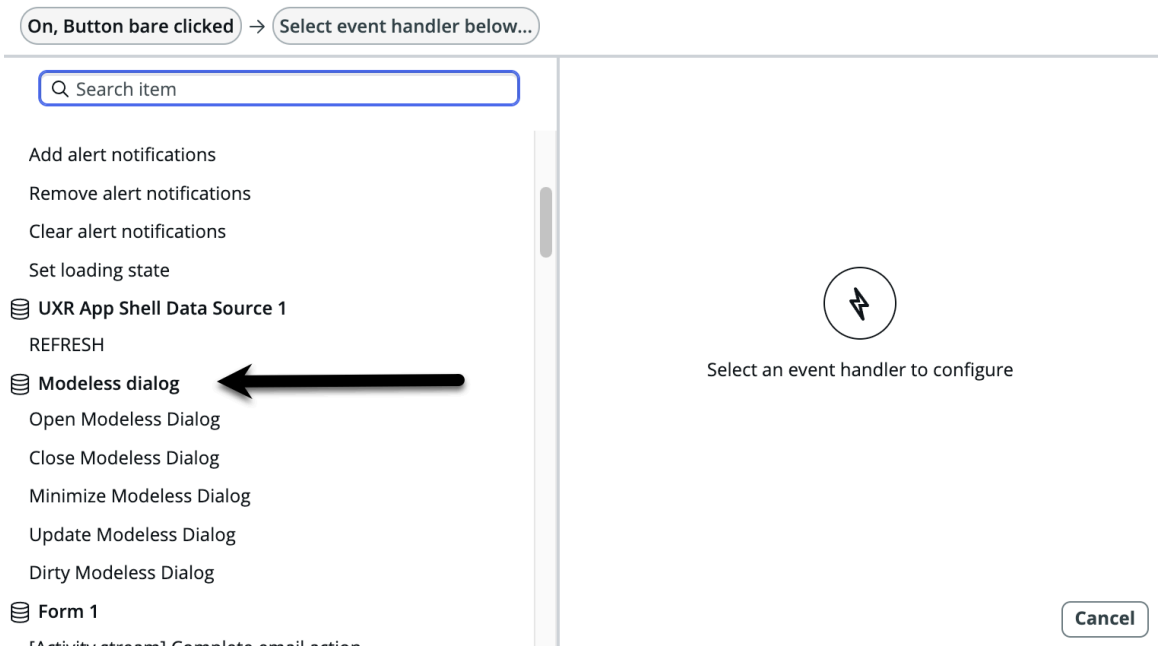
1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

3. Open or create a page variant.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. Add a modeless dialog to the page.
   For more information about how to create a modeless dialog, see Add modeless dialog to a UI Builder page.
5. Add a component to the page, such as a button, to take action on the modeless dialog.
   For more information about how to create a button that interacts with a modeless dialog, see Add modeless dialog to a UI Builder page.
6. Select the button component in the content tree or on the stage.
7. In the configuration panel, select the **Events** tab.
8. Select **+ Add event handler**.



9. From the Event handler preview window, select an action to assign to the button.
   There are five modeless dialog event handlers available.



Note: For any of the event handlers, select **Script** in the **Mode** drop-down if you prefer to work in the script editor.

To add a modeless dialog event handler	Do the following
Open Modeless Dialog	Opens a modeless dialog. For example, add a button labeled Compose email and configure

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

To add a modeless dialog event handler	Do the following
	the button to open a modeless dialog containing the Email composer (mini) component. ○ **Heading**: Type a heading for the modeless dialog. ○ **Minimized Heading**: Type a heading that is displayed when a user selects the minimized dialogs drop-down icon on the page. ○ **Category**: Type a category name to group the modeless dialog. ○ **Modeless Dialog**: Select the modeless dialog to open. ○ **Single instance**: Select this option to enable users to open only one instance of the modeless dialog. Leave this option unselected to enable users to open multiple instances of the modeless dialog. For example, if the button opens a modeless dialog containing the Email composer (mini) component and you don't select the single instance option, users can open multiple modeless dialog windows to compose email messages. ○ **Modeless Dialog Instance ID**: If you select the **Single instance** option, specify the modeless dialog instance ID that should be opened on click. Move your mouse over the field and select the **Bind data** icon. In **Data types** select **Modeless Dialog**. Drag the instanceID pill to the top section and select **Apply**. ○ **When to trigger** (Advanced option): Select **Always** to have the component (in this example, the button) always open the modeless dialog. Select **Conditionally** to add an event handler condition. For more information about event handler conditions, see Supported functions in the UI Builder component formula editor.
Close Modeless Dialog	Closes a modeless dialog. Use this option if you want to configure a component, such as a button, to close the modeless dialog from outside the modeless dialog window. ○ **Modeless Dialog Instance ID**: Specify the modeless dialog instance ID that should be closed on click. ○ **When to trigger** (Advanced option): Select **Always** to have the component (in this example, the button) always close the modeless dialog. Select **Conditionally** to add an event handler condition. For more informa

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

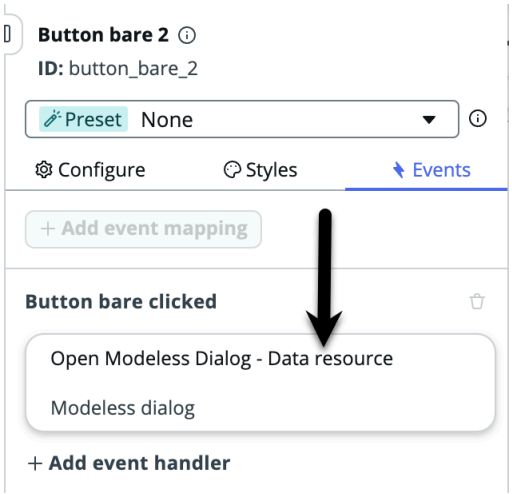
To add a modeless dialog event handler	Do the following
	tion about event handler conditions, see Supported functions in the UI Builder com ponent formula editor.
Minimize Modeless Dialog	Minimizes an open modeless dialog. Use this option if you want to configure a component, such as a button, to minimize the modeless dialog from outside the modeless dialog win dow. ○ **Modeless Dialog Instance ID**: Specify the modeless dialog instance ID that should be minimized on click. ○ **When to trigger** (Advanced option): Select **Always** to have the component (in this ex ample, the button) always minimize the modeless dialog. Select **Conditionally** to add an event handler condition. For more in formation about event handler conditions, see Supported functions in the UI Builder component formula editor.
Update Modeless Dialog	Updates the specified fields on a modeless dialog. For example, add a button labeled Up date Heading to the footer of a modeless di alog and configure the button to update the heading of the component in the modeless di alog. ○ **Heading**: Type new heading text for the modeless dialog that should be added when the button is selected. ○ **Minimized Heading**: Type new minimized heading text that should be added when the button is selected. ○ **Category**: Type a new category name that should be added when the button is select ed. ○ **Modeless Dialog**: Select the modeless dia log to update. ○ **Modeless Dialog Instance ID**: Specify the modeless dialog instance ID that should be updated on click. ○ **Is Dirty**: Select this option to mark the mod eless dialog as dirty. ○ **When to trigger** (Advanced option): Select **Always** to have the component (in this ex ample, the button) always update the mod eless dialog. Select **Conditionally** to add an event handler condition. For more infor mation about event handler conditions, see Supported functions in the UI Builder com ponent formula editor.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

To add a modeless dialog event handler	Do the following
Dirty Modeless Dialog	Marks a modeless dialog as dirty. ○ In **Modeless Dialog Instance ID**, specify the modeless dialog instance ID to mark as dirty. ○ **When to trigger** (Advanced option): Select **Always** to have the component (in this example, the button) always marked as dirty. Select **Conditionally** to add an event handler condition. For more information about event handler conditions, see Supported functions in the UI Builder component formula editor.

10. Select **Add**.

Result
The configured event handler displays in the configuration panel **Events** tab for the component.



Configure alerts to auto-dismiss

Set alerts to automatically clear after a specified time period.

Before you begin
Role required: ui_builder_admin

About this task
You can configure alerts to auto-dismiss in two ways:

- Experience settings: apply auto-dismiss rules to all alerts of a specific type
- Events: add an event to override experience settings for specific alerts

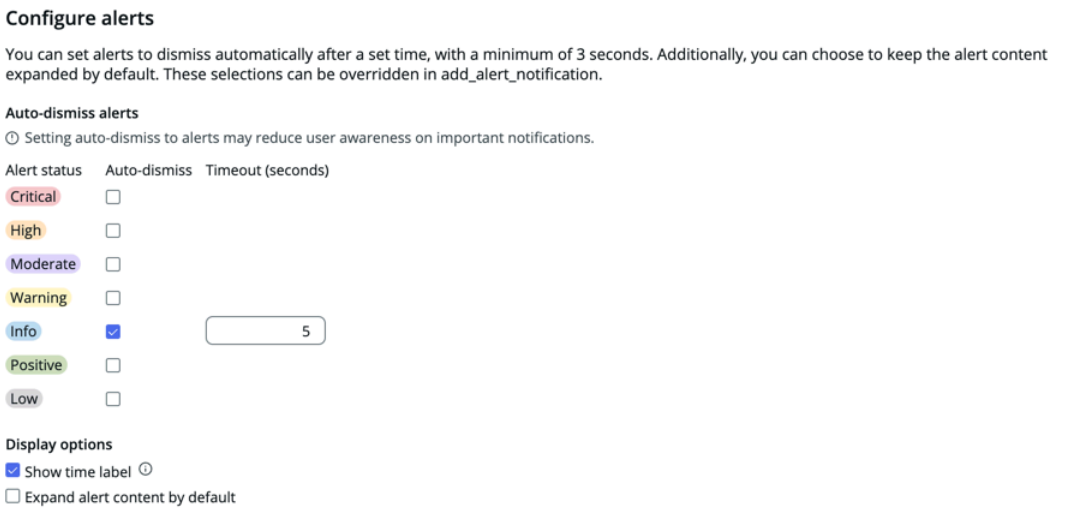
Important: You must first enable auto-dismiss at the experience settings level before you can configure it at the event level.

Configurations made through an event take priority over those made in the experience settings. For example, in the experience settings, you might set all **Positive** alerts to auto-dismiss after 5 seconds. Then, for a specific **Positive** alert, you could override this initial configuration by adding an event that sets it to auto-dismiss after 10 seconds, or to disable auto-dismiss entirely.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

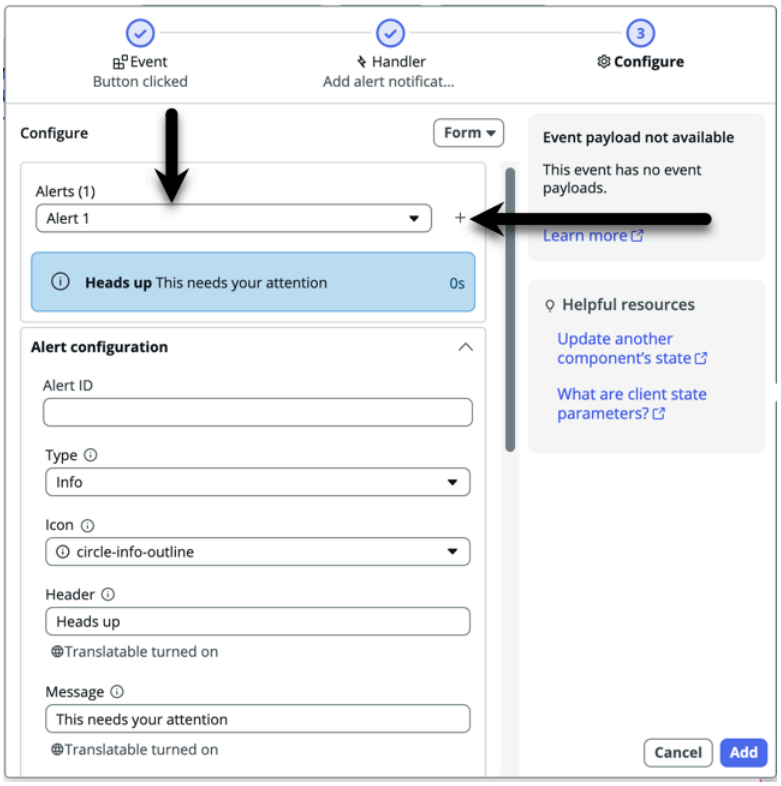
1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Select **View experience settings**.
4. Under **Configure alerts**, select the **Auto-dismiss** option next to the alert type you wish to configure.



   For this example, select the **Auto-dismiss** option next to the **Info** alert type and enter 5 in the **Timeout (seconds)** field. **Info** alerts disappear after 5 seconds.

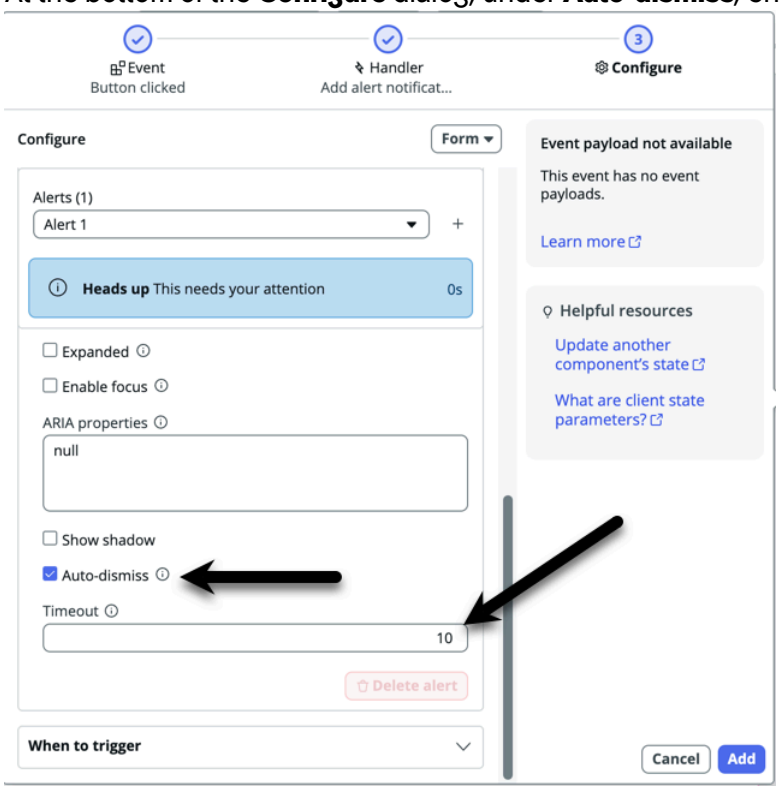
5. Select **Save**.
6. Return to the experience view.
7. Create or open a page.
8. In the content tree, select **+ Add content**.
9. Select a **Button** component and select **Add**.
10. In the configuration panel, select the **Events** tab.
11. Under **Button clicked**, select **Add handler**.
12. Select **Add alert notifications** and select **Continue**.
    The **Configure** dialog enables you to edit multiple alerts in one place and displays a preview of the alert being edited. You can select the **+** to add more alerts, then select the alert in the dropdown to configure it.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



For this example, leave the alert **Type** as **Info**.

13. At the bottom of the **Configure** dialog, under **Auto-dismiss**, enter 10 in the **Timeout** field.



This specific **Info** alert disappears after 10 seconds.

14. **Optional:** You can also clear the check box to turn off auto-dismissal for this individual alert.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: Other alerts of the same type retain the initial configuration in the experience settings until overridden through an event.

15. Select **Add**.

Result
Alerts of the selected type are now set to auto-dismiss after the specified time period.

Manage actions in UI Builder pages

Learn how to work with events so that you can add actions to components, pages, data resources, and declarative actions in UI Builder.

Actions in UI Builder

UI actions tell UI Builder what to do when an event is triggered. An event is an action a user takes or occurrence that happens on a page. Use UI actions to create interactive, user-friendly interfaces that help your users complete tasks. Each component has its own events associated with it. Events include:

- User clicks a data visualization
- Page successfully fetches data
- User selects a radio button
- Page loads

Events in UI Builder

Use an event to add actions to your components, pages, and data resources.

- A component event is an action that you set up for a component. You set up an event handler to configure that component action. For example, you can add a button component to your UI Builder page. Then, you can add an event handler to apply an action for that button, such as going to a web page.
- Page events perform actions for the entire page. You can configure the following page events:
  - Page event mappings. Add, remove, or clear alert notifications on your page.
  - Variant event mappings. Add, remove, or clear alert notifications on your page variant.
  - Dispatched events. Create dispatched events for your page to create relayed event mappings that model events after a parent event handler. Select a target parent event handler to model the payload fields after it.
  - Handled events. A handled event is an event that is exposed and available for use by other users. After you create a handled event, it is available under *`Page event mappings`* for other users to use. You can also set up payload fields that you create manually or choose a template for your handled event, such as an open or close a modal dialog.
- Data resources events map data resources to notify information about when data is fetched.
- Events for a page or component do nothing until the event is mapped to one or more event handlers.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Event mapping in UI Builder

Map actions to events such as clicking a button or filling in a field. An event mapping is the process that enables you to map an event's payload or contextual values to the object or handler that acts on that event.

For more information about event mapping, see Define map events.

Event handlers in UI Builder

An event handler is an action performed when an event occurs. By mapping an event handler to an event, you are specifying which action or actions to take when the event occurs. Use an event handler to configure an action for your UI Builder page or the components on the page. For example:

- Clicking a data visualization opens a list of records represented in the visualization
- Fetching data successfully for a list opens an alert that indicates the data fetch was successful
- Selecting a radio button adjusts the filter for a list on the page
- Loading a page opens a modal to confirm acceptance of cookies before proceeding

When you add an event handler to a UI Builder page or component, you can choose different types of event handlers. For example, a Button component can have the following types of event handlers:

- Inherited event handlers. An inherited event handler is exposed from the page that you are working in. If you are in the parent UI Builder page, an inherited event handler could be exposed from the app shell. The following table lists the different types of inherited event handlers that you can use and what you can do with them.

Inherited event handlers

Event handler	Description
Breadcrumb URL changed	
Link to destination	Navigate to a destination. ○ App routes: Link to another page within an app, like a home screen. ○ External URL: Link to a website or any external URL. Sample script `return { route: null, /* Page route, e.g. 'record' */ fields: null, /* Required params, e.g. {"table":"incident","sysId":"X"} */ params: null, /* Optional params, e.g. {"selectedIndex" : 1} */ ☑#redirect: null, /* ??? True/false? */`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Event handler	Description
	``☑#passiveNavigation: null, /* Load in background, e.g. 'false' */ title: null, multiInstField: null, ☑#targetRoute: null, /* ??? */ ☑#external: null /* ??? True/false? */ };```
Add parameters to URL	Add additional parameters to a URL. Sample script ```{ “selectedTabIndex” : 0 }```
Open or close modal dialog	After you create a modal, use *Open or close modal dialog* to trigger the modal.

- Page-level event handlers. This type of event handler is common to all pages, and you would use this handler type when you want to add or clear page-level alert notifications. The following table lists the different types of page-level event handlers that you can use and what you can do with them.

Page-level event handlers

Event handler	Description
Add alert notifications	Add a code snippet to send an alert notification. For example: ```return { items: [{"type" : "info", "message" : "Info message", "id" : "optionalID"}] /* Types: info, warning, error */ };```
Remove alert notification	Add code to call alert notification IDs that you want to dismiss. For example, click a button to remove a page load alert notification.
Clear alert notification	Add code to call all alert notification IDs that you want to dismiss. For example, click a button to remove all alert notifications.
Set loading state	Toggle loading on or off. For example, you can toggle loading on to load the page when you click a button or toggle loading off to not load the page when a button is clicked.
Update client state parameter	Declaratively set the client state parameter. Let's say that you have a client state parameter that you set up with a value. You can configure the *Update client*

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Event handler	Description
	`state parameter` event handler to update the client state parameter with a new value. For example, you have a client state parameter that is called **Greeting** that is set up with `Hello` as the initial value. You can add an *`Update client state parameter`* event handler, select the **Greeting** client state parameter, and then enter a new value like `Goodbye`. When you click the button, **Goodbye** replaces **Hello** on the page.
UXF macroponent viewport load requested	Add to a component, such as a button component, to open a viewport. For more information, see Add a viewport component to your page.

- Data resource handlers. This type of event handler triggers the fetching or writing of data to the server.

  You can refresh the app shell data source data on your UI Builder page by clicking a button. For example, with a data resource handler, you can do the following actions:

  - Bind data to the description of an incident record.
  - Change the value of the incident description.
  - Add a button component to your page.
  - Label the button as `Refresh incident`.
  - Add a *`Look Up Record`* event handler for the button.
  - Save your page.
  - If the description of the incident record changes, click **Refresh** to update the description on your page.
- Client scripts. Scripts that execute when an event is triggered on a component. You create these scripts in the Client scripts section in UI Builder. For more information, see Define and bind client scripts to components.

Binding events to components in UI Builder

Bind event handlers to the events on a component. When you add components to your UI Builder page, these components are not configured to perform any action on your page. For example, a button component is static and does not do anything until you bind an event action to it, such as deleting a record. Some components have several events where event handlers can be assigned. For example, on the list component, you can assign a navigation handler to the *`Row clicked`* event. You can also assign an open modal to the *`Data fetch failed`* event. For more information, see Bind an event to a component.

Binding events to UI Builder pages

Bind a page-level event to execute event handlers on the page. For example, use page-level events for page notifications, page load, or when a page property changes. The assignment of the event handler to the page-level event is similar to assigning handlers to events from components.

You can bind event mappings using the following methods:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Page event mappings. Add, remove, or clear alert notifications on your page.
- Variant event mappings. Add, remove, or clear alert notifications on your page variant.
- Dispatched events. Create dispatched events for your page to create relayed event mappings that model events after a parent event handler. Select a target parent event handler to model the payload fields after it.
- Handled events. Add a handled event for an event that is exposed and available for use by other users.

For more information, see Bind an event to a page.

Binding events to data resources in UI Builder

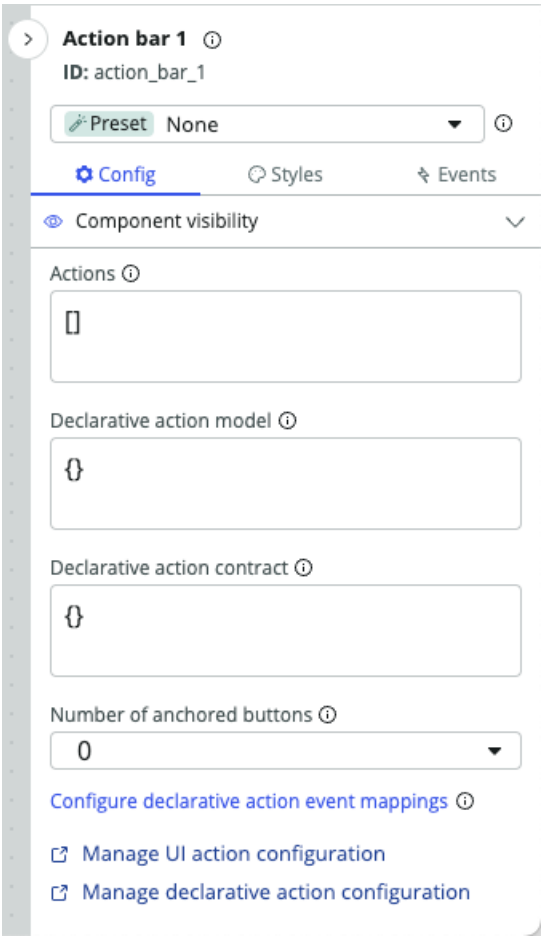
Bind event handlers to individual data resources on your UI Builder page. For example, when a data resource successfully fetches new data, it executes an event handler, like navigation, to take a user to the next step in a flow. When a data resource successfully adds a record to a table, it shows a success modal that uses the show modal event handler. For more information, see Bind an event to a data resource.

Binding events to declarative actions in UI Builder

Bind data elements to add event actions to a declarative action definition in **Actions & Components** in the ServiceNow AI Platform®. For example, you could bind a data element to add an event action to complete work on a table.

If you add a component to your UI Builder page that has a declarative action, you must bind it to a handled event. The handled event creates an action that is performed when a user selects the component. By selecting **Configure declarative action event mapping**, you add a new event handler to define what the declarative action does on the page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



For more information, see Bind an event to a declarative action.

Define map events

An event mapping in UI Builder is the process that enables you to map an event's payload or contextual values to the object or handler that acts on that event. The four event types are: component, page, data resource, and declarative action.

Event mapping is an important process within UI Builder. When you build pages with components, you need those components to perform actions for users. For example, if you add a button component to the page, a button-clicked event must be mapped to an event handler. The event handler performs a button-clicked action when it is selected by a user. An example is when you add a data resource, such as a form, and have an event handler notify the user when the form successfully loads.

Event types

The event types that are available are based on the component. For example, declarative action events are available for specific components, such as the Action bar and List components.

You choose a type of event based on what action you want to perform on your page. For example, if you want to bind an action to a component, such as a button loading a web page, you would use a component event. If you want an event to apply to your whole page, such as adding an alert notification to a page, you use a page event. The following table describes each event type that is available in UI Builder and provides some examples on how you can use the events.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Event types and descriptions

Event type	Description
Component events	Action that you set up for a component. You set up an event handler to configure that component action. For example, add an event handler to apply an action for a button, such as going to a web page. For more information on binding events to components, see Bind an event to a component.
Page events	Page event that performs an action for the entire page. You can configure the following page events: • Page event mappings. ○ These event mappings are saved on the page definition record, which can be found in the [sys_ux_macroponent] table. ○ The source events for these event mappings are Page ready and Page property. They are defined for your page. ○ The available handlers for these event mappings are: ▪ User session events ▪ Page-level events ▪ UXR App Shell Data Source ▪ Client scripts that are defined on the page ▪ Operations from local and inherited data resources • Variant event mappings. ○ These event mappings are saved on the variant record, which you can find in the [sys_ux_screen] table. ○ The source events for these event mappings are any dispatched events that are defined for your page. ○ The available handlers for these event mappings are canvas-level events, the UXR App Shell Data Source, and operations from the inherited data resources. ○ Variant event mappings are used as relays so that the events that are dispatched from components on your page can be relayed up to user session event handlers. ○ Mappings are created automatically on a page save when you have mapped a component's dispatched event to a user

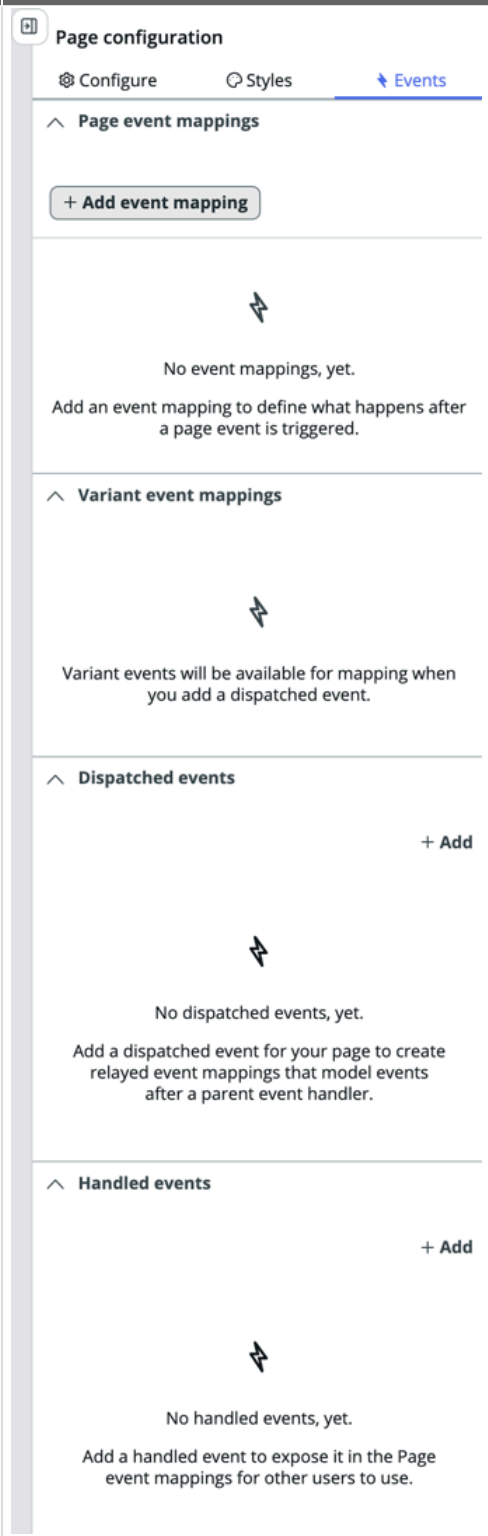
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Event types and descriptions (continued)

Event type	Description
	session handler. For example, by mapping a **Button clicked** event to the canvas-level **Link to destination** event, the event creates a dispatched relay event and a variant event mapping when the page is saved. • Dispatched events. ○ These events are saved in the [sys_ux_event] table. The page definition record contains references to the [sys_ux_event] record. ○ These events serve as source events for variant event mappings. ○ You can select **+Add** to create new dispatched events for your page. Configure the event label to auto-populate the event name and payload fields for your new `sys_ux_event`. ○ These events are used to create relay event mappings. You can select a **Target parent event handler** to model the payload fields after the selected event. You can also manually create payload fields for your dispatched event. • Handled events. ○ These events are saved in the [sys_ux_event] table. The page definition record contains references to the [sys_ux_event] record. ○ These events serve as source events for page event mappings. ○ You can select **+Add** to create new dispatched events for your page. Handled events are not modeled after parent event handlers. Payload fields for handled events are created manually. To use an existing handler's payload fields as a template, select a template and edit the fields as necessary.

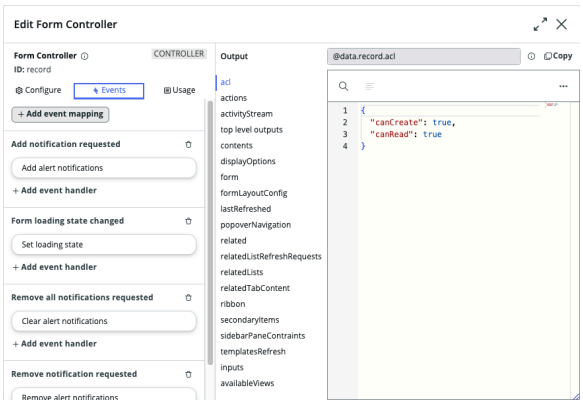
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Event types and descriptions (continued)

Event type	Description
	 For more information on binding an event to a page, see Bind an event to a page.
Data resource events	Events that are mapped to data resources to provide notifications about when data is fetched.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Event types and descriptions (continued)

Event type	Description
	• Data Fetch Initiated. When a data resource event is triggered, the event handler executes the data fetch process. • Data Fetch Succeeded. When a data resource event is triggered, the event handler executes the process to notify a user when the data fetch completed successfully. • Data Fetch Failed. When a data resource event is triggered, the event handler executes the process to notify a user if the data fetch was unsuccessful.  For more information on binding an event to a data resource, see Bind an event to a data resource.
Declarative action events	Bind data elements within UI Builder to add event actions to a declarative action. You configure a declarative action event mapping in the ServiceNow AI Platform® declarative action assignment table. For an example, navigate to **Filter navigator** > **sys_declarative_action_assignment.list** and then search for and open an existing declarative action. In UI Builder, you bind an event to the declarative action. For more information on how to use declarative action events, see Bind an event to a declarative action.

Event payloads in UI Builder

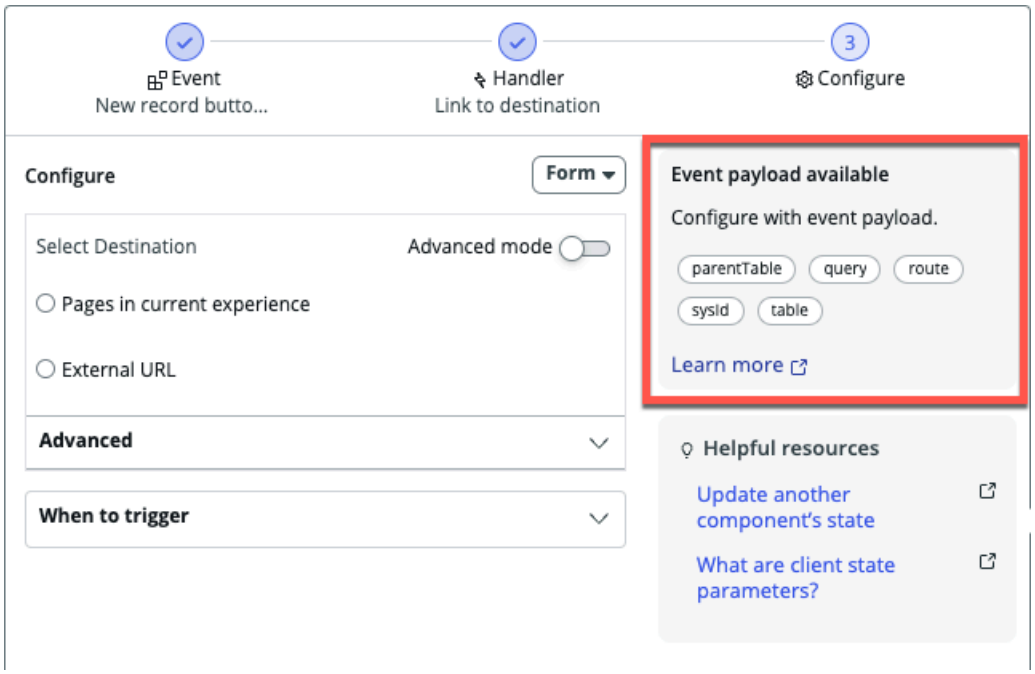
Use event payloads to link data to an action.

Event payloads are pieces of data sent by a component when a selected event is triggered. The data sent by an event can include the type of event, timestamps, user actions, or resource

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

data such as a SysID. You can use event payloads in UI Builder to share information between components on a page.

You can use this payload data when configuring an event handler so that the resulting interaction can be linked to the emitted data. For example, a SysID can be passed to other components on a page to display information related to a specified record.



Each component and event has a unique set of payload options. Event payloads may not be properly defined for each component. If that is the case, define a client script so that the console logs payloads such as `console.log(event.payload)`.

Create an event with payloads to share information between components

Use event payloads in UI Builder to share information between components on a page.

Before you begin

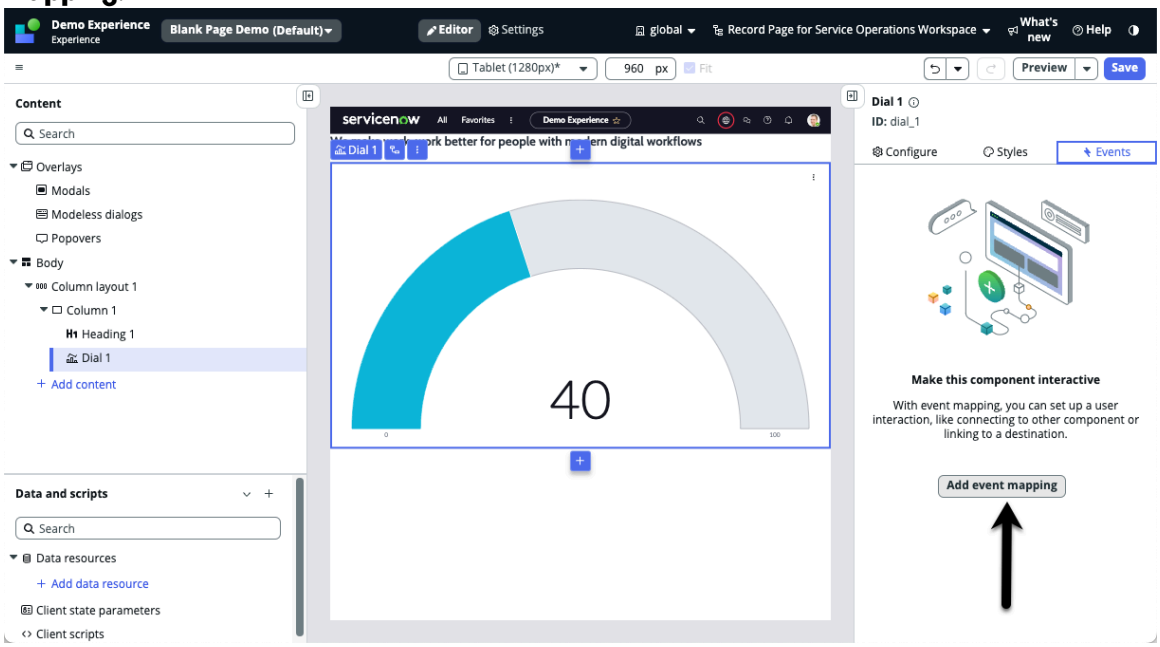
Role required: ui_builder_admin

Procedure

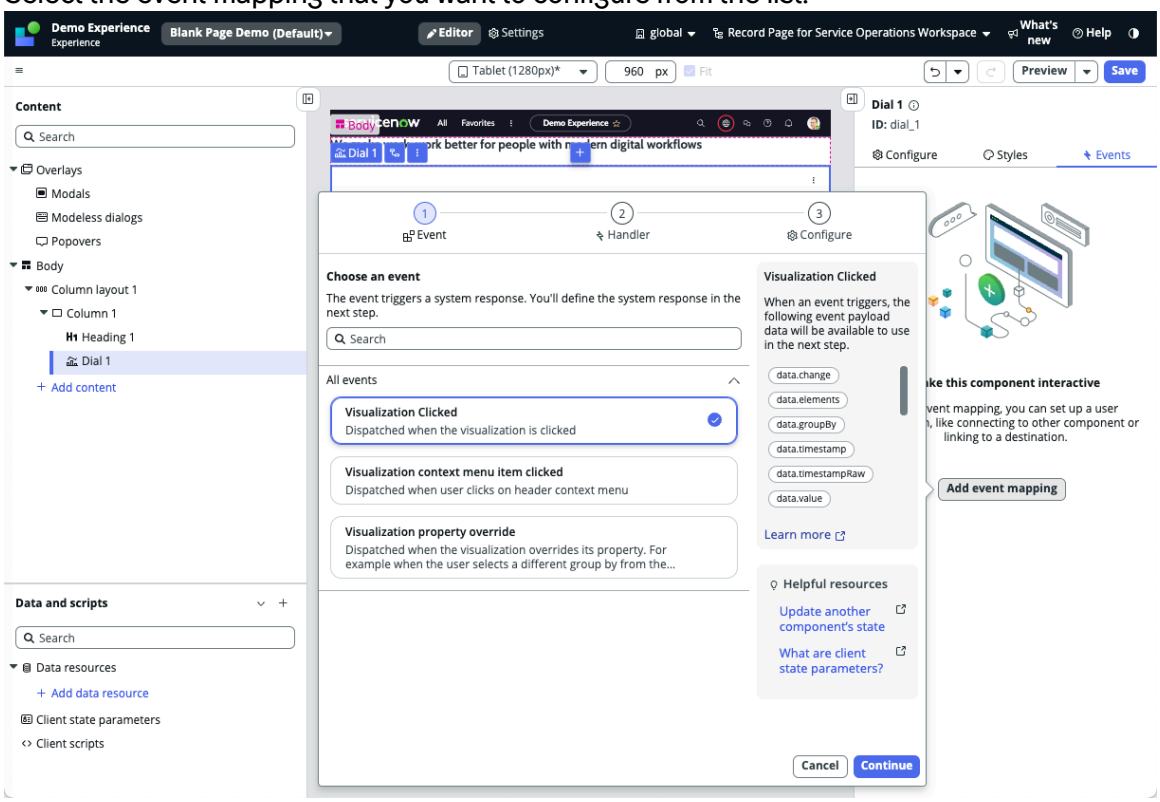
1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   For more information, see Configure how users interact with your applications in UI Builder.
3. Create or open a page.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. Add a component that has events, such as the data visualization component.
   For more information about adding components to a page, see Add and configure components.
5. To add an event handler to your component, select the **Events** tab.
   For more information on how to add event handlers to your component, see Manage actions in UI Builder pages.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. To start the process of setting up an event handler for your component, click **Add event mapping**.



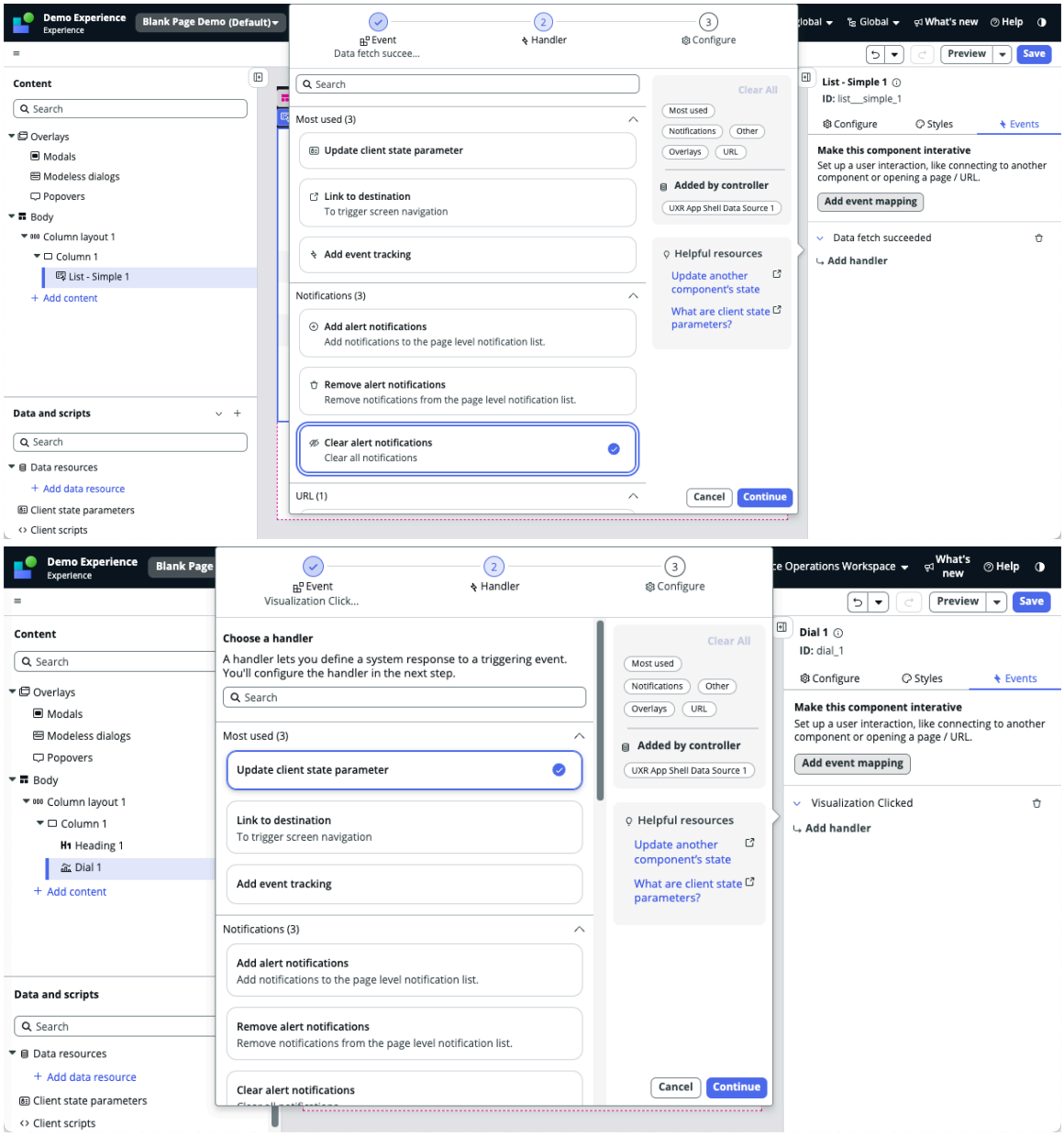
b. Select the event mapping that you want to configure from the list.



c. Select **Continue**.

d. From the Event handler preview section, select an event handler to bind to your component.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



e. Select **Continue**.

f. Select the **Create a new client state parameter** from the **Client State Parameter Name** drop-down.

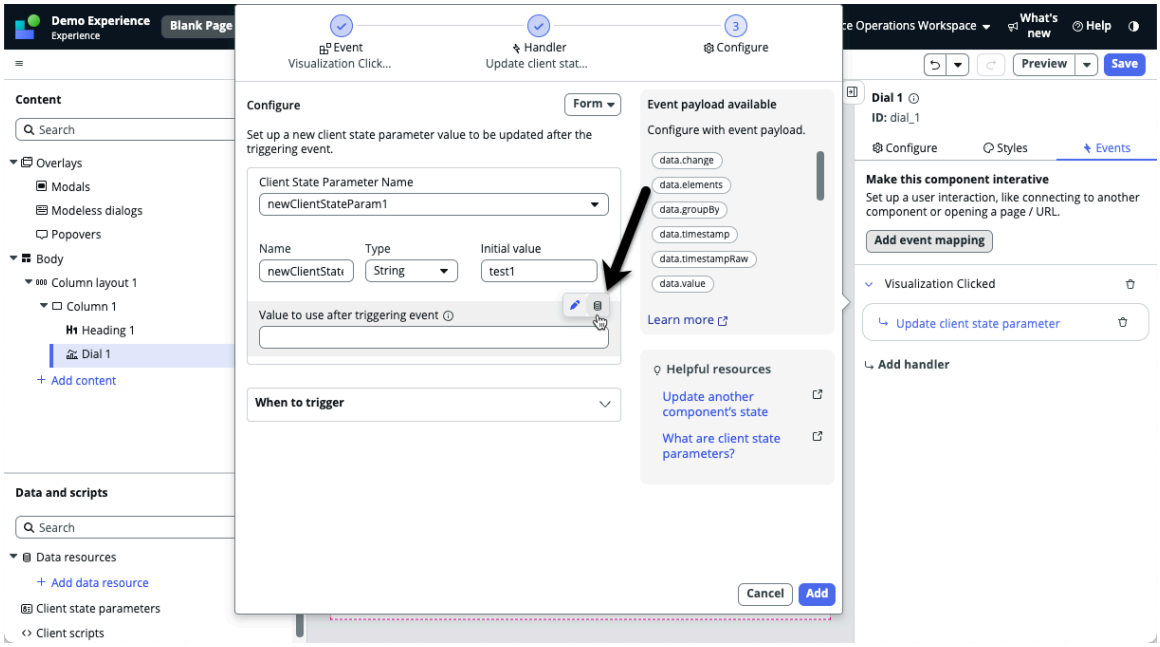
g. Enter a name in the **Name** field.

h. Select the type of client state parameter from the **Type** drop-down.

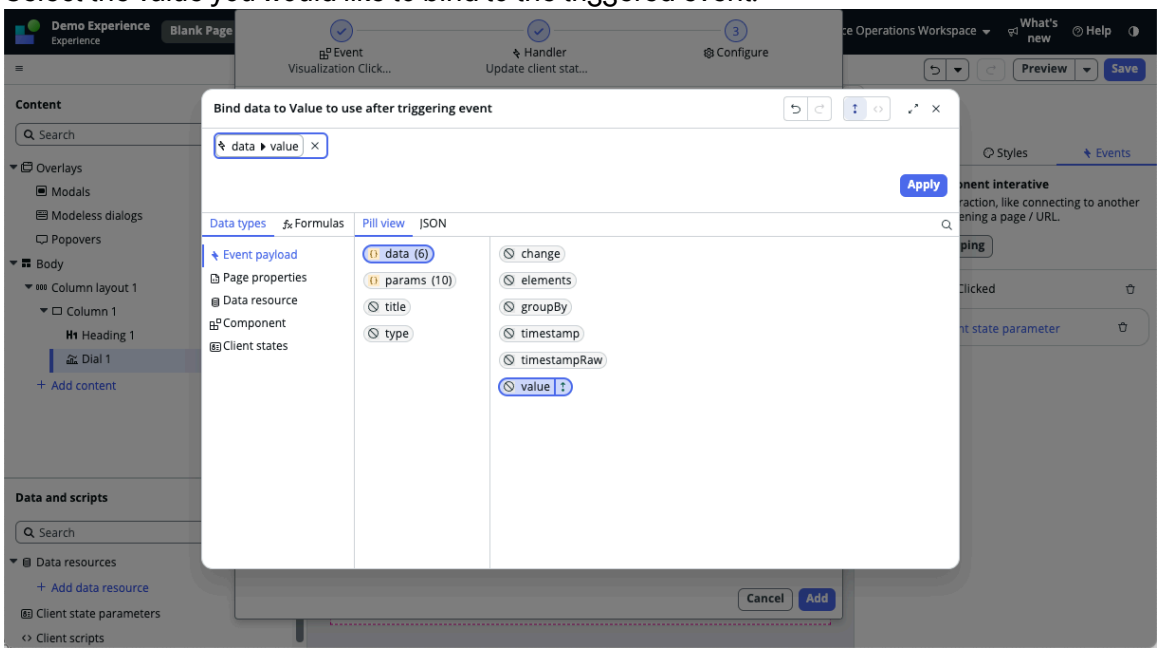
i. Enter a value in the **Initial value** field.

j. Select the bind data icon in the **Value to use after triggering event** field.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



k. Select the value you would like to bind to the triggered event.

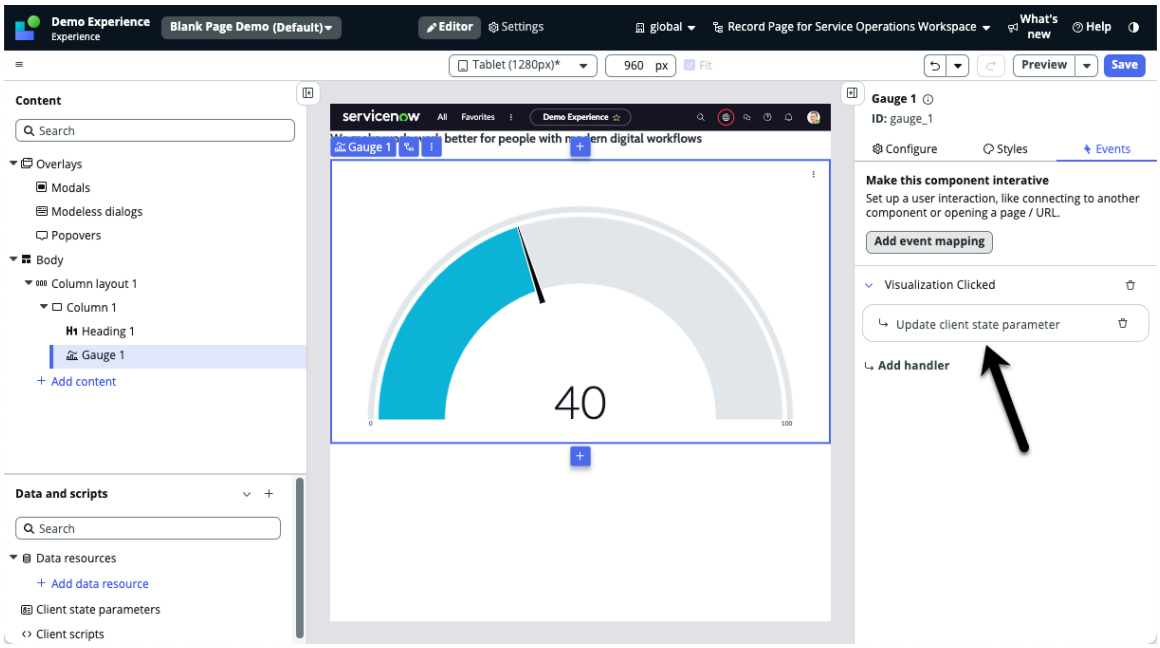


l. Select **Apply**.

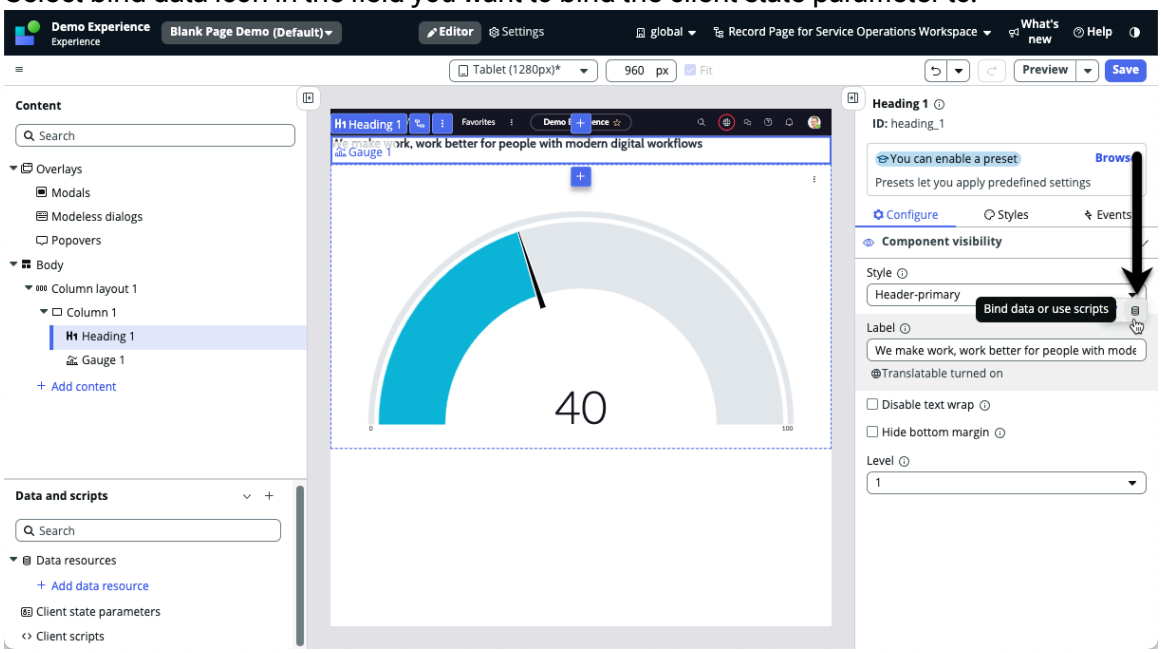
m. Select **Add**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The data handler will appear below the event mapping in the events tab.

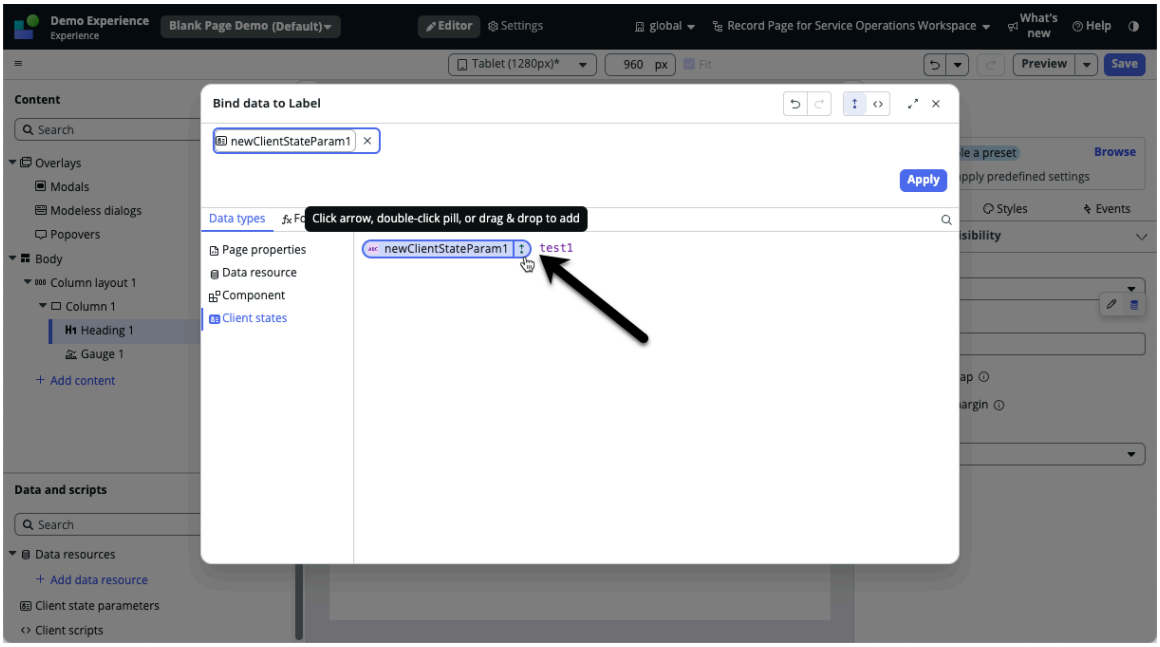


6. Select the component you want to bind the client state parameter to, in this example we will be using a heading component.
7. Select bind data icon in the field you want to bind the client state parameter to.



8. Select **Client states**.
9. Select the arrow next to the client state parameter you want to bind.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



10. Select **Apply**.
11. Select **Save**.
12. Select **Preview** to test your the event you just configured.

Configure an event handler manually

Add an event handler to a page, component, data resource, or declarative action within UI Builder so that your user can trigger an action.

Before you begin

Role required: ui_builder_admin

About this task

An event handler lets you configure an action, components, data resource, or declarative action on your page. For example, you can map an event to your page to add an alert notification when the page successfully loads or you can add an event handler for a button component to perform an action when a user clicks it. The event handler could also be a modal on your page that asks a user to verify that the user wants to delete the record. For more information, see Manage actions in UI Builder pages.

Note: You can also use Now Assist in UI Builder to configure event handlers. For more, information, see Configure an event handler with ServiceNow Otto

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   For more information, see Configure how users interact with your applications in UI Builder.
3. Open or create a page.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. Add a component to your page, such as a button.
   For more information about adding components to a page, see Add and configure components.
5. To add an event handler to your component's event, go to the configuration panel and select **Events**.

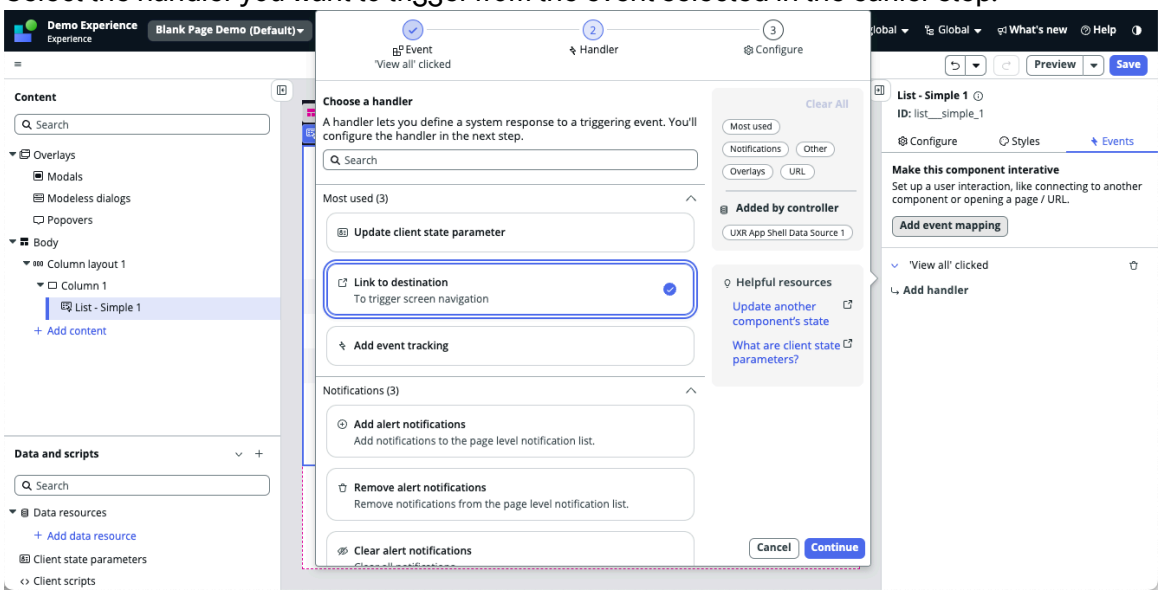
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

An event handler lets you assign an event to a component. For example, if you add a button component to your page, you want it to perform an action when a user clicks it.

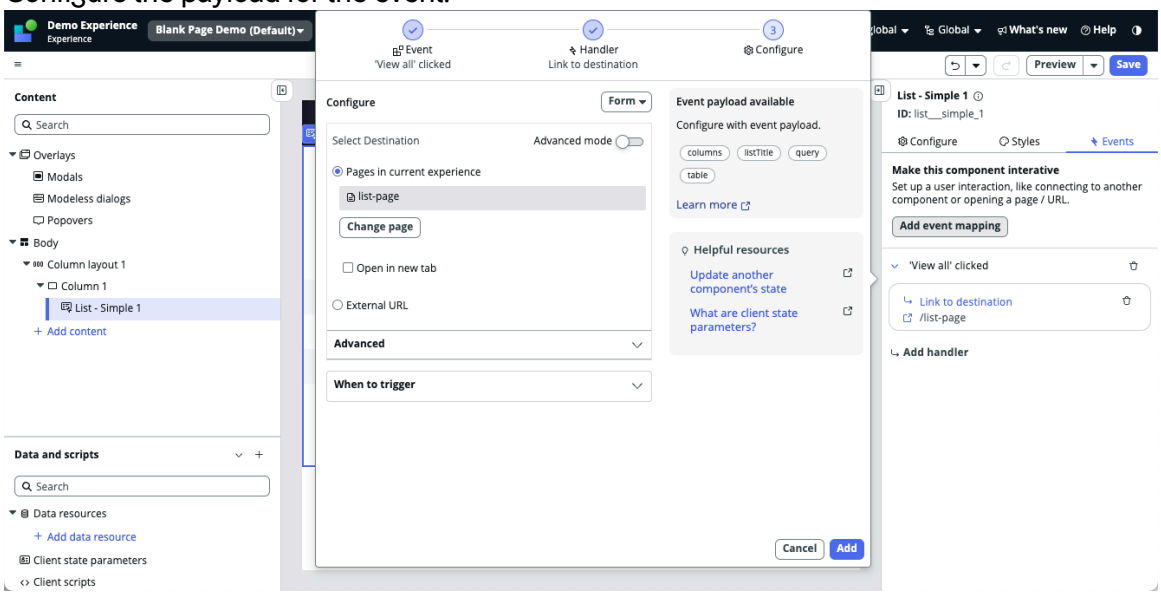
6. To start the process of assigning an event to a component select **Add event mapping**.

   Some components, such as the button component, only have one event mapping. Other components can have many events.

7. Select an event mapping that you want to configure from the list.
8. From the Event handler preview screen, select an action that you want assigned to the component.
9. Select **Continue**.
10. Select the handler you want to trigger from the event selected in the earlier step.



11. Select **Continue**.
12. Configure the payload for the event.



13. Select **Add**.
14. Select **Save**.
15. Test the event by selecting **Preview** in the header and trigger the action.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Result
The configured event handler displays in the events tab of the configuration panel.

Configure an event handler with Now Assist

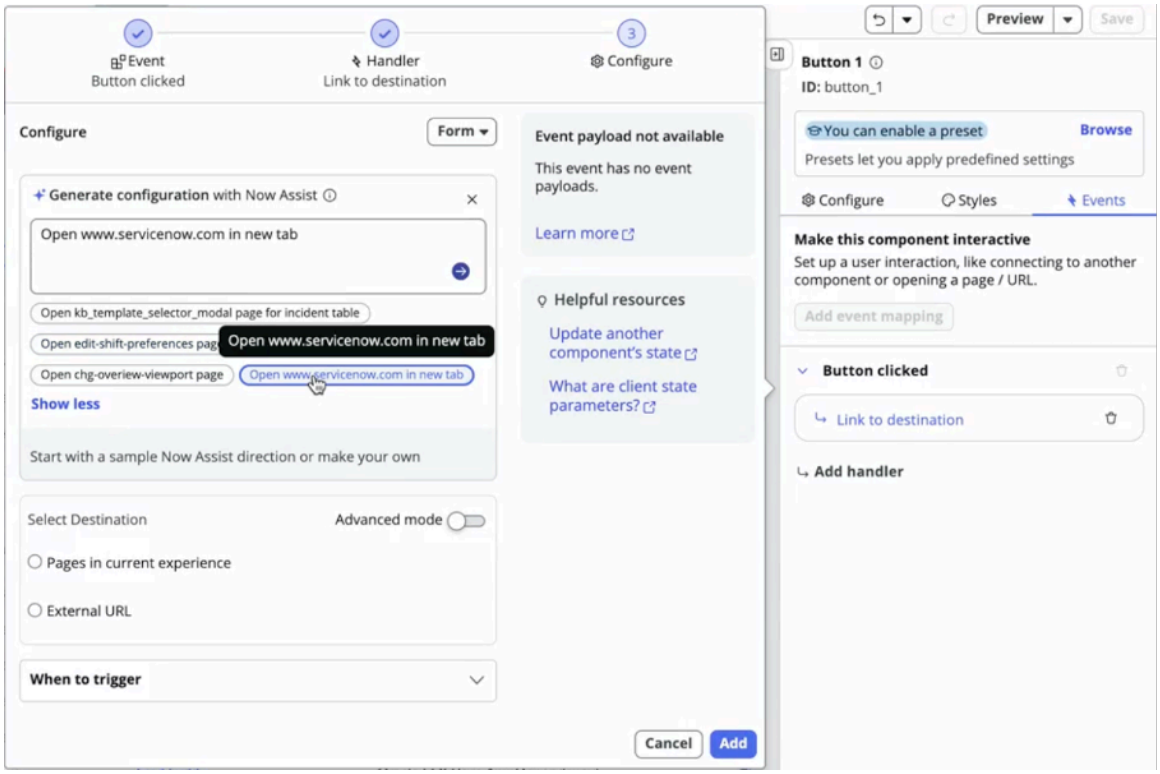
Use Now Assist in UI Builder to configure event handlers. At present, you can configure link to destination, open or close modal, and view load requested event handlers using Now Assist in UI Builder.

Before you begin
Role required: ui_builder_admin

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   For more information, see Configure how users interact with your applications in UI Builder.
3. Open or create a page.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. Add a component to your page, such as a button.
   For more information about adding components to a page, see Add and configure components.
5. To add an event handler to your component's event, go to the configuration panel and select **Events**.
   An event handler lets you assign an event to a component. For example, if you add a button component to your page, you want it to perform an action when a user clicks it.
6. In the event handler, select **Link to destination** and select **Continue**.
7. In the Configure section, select **Get Started**.
8. In the **Generate configuration with Now Assist** field:
   - Type the action you want to perform (For example, open www.servicenow.com) or select an option from the dynamic prompts.
   - You can reference data from page context.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



9. Select the right arrow icon.
   The configurations are set.
10. You can either reject, or accept and edit the configurations.
11. After you review the changes, select **Add**.
12. Test the event by selecting **Preview** in the header and trigger the action.

Result
The event handler configurations are updated.

Bind events to add actions

After you have created events, you can bind them to actions on your components, pages, data resources, and declarative actions.

Bind an event to a component

Bind data elements within UI Builder so that you can add event actions to your components.

Before you begin
Role required: ui_builder_admin

About this task
Each component has specific events that it binds to. For example, a button component has only a button-clicked event, while other components can have multiple events that are associated with them.

Some components do not have an event action that is applied to them. An example is the heading component. Many components require that you map an event to your component to make it perform an action, such as loading data.

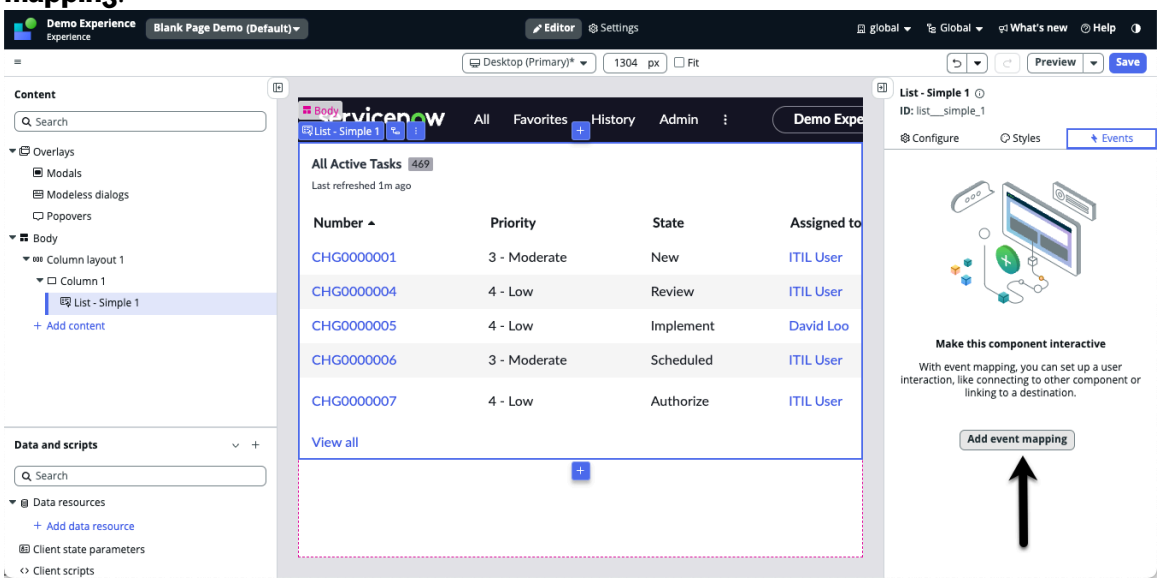
To add actions to components, pages, and data resources on your page, you can add an event handler. A button component that you add to a page is static. By binding an event action to the button, you can link it to a web page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

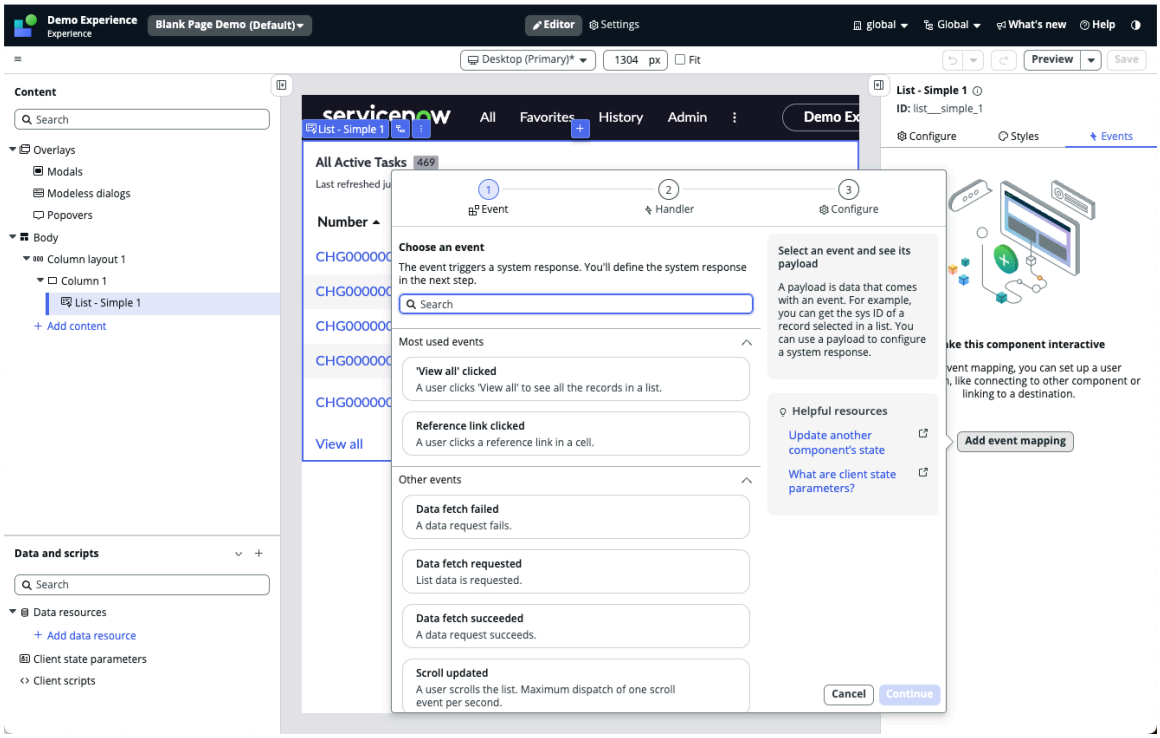
1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   For more information, see Configure how users interact with your applications in UI Builder.
3. Create or open a page.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. Add a component that has events to which it can bind to your page, such as a button.
   For more information about adding components to a page, see Add and configure components.
5. To add an event handler to your component, select the **Events** tab.
   For more information on how to add event handlers to your component, see Manage actions in UI Builder pages.

   a. To start the process of setting up an event handler for your component, click **Add event mapping**.



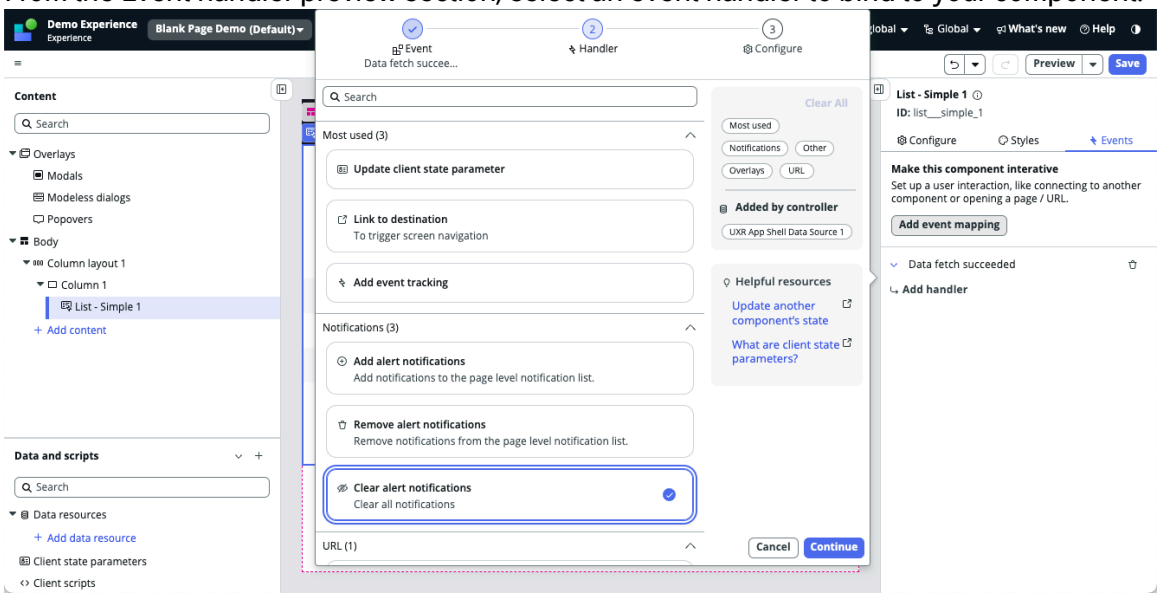
   b. Select the event mapping that you want to configure from the list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



c. Select **Continue**.

d. From the Event handler preview section, select an event handler to bind to your component.



e. Select **Continue**.

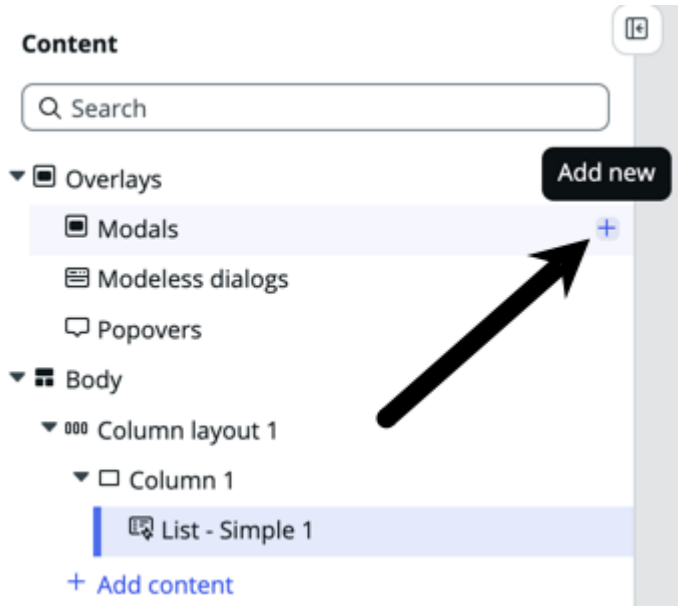
f. Configure the payload for the event.
You configure each event handler differently, depending on the action of the event. For example, if you add an event handler for a button component, you can choose what that button action performs.

g. Select **Add**.

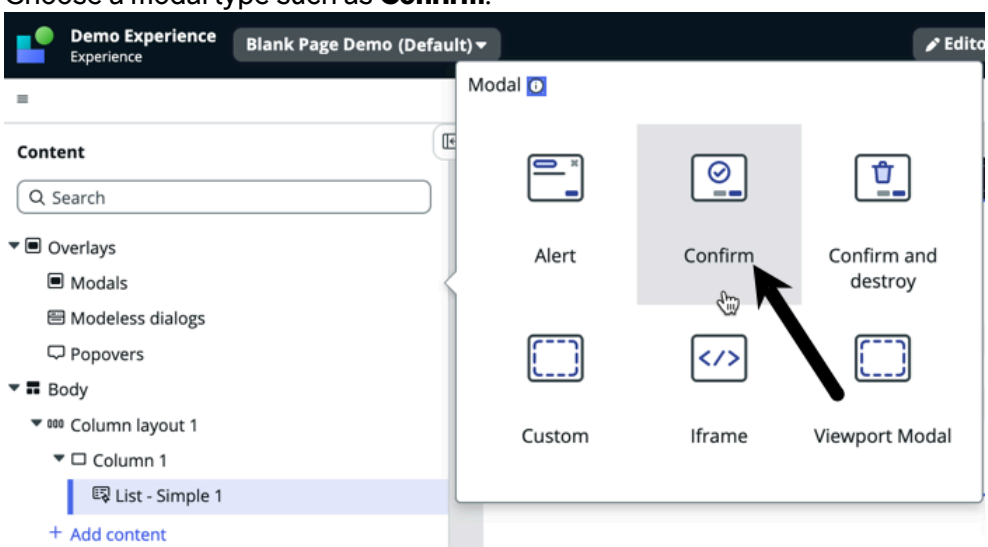
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

6. **Optional:** If you want a modal to pop up for your event, add the modal to the page before you bind your event to the component.
   A modal is a confirmation pop-up that appears when you click the component. For example, if you add a button component that deletes a record, you add a modal to ask the user to confirm that the user wants to delete the record. For more information, see Add modal to component.

   a. Select the **+** icon in the content tree next to **Modals**.



   b. Choose a modal type such as **Confirm**.



   c. Configure the modal.
   You can change the text in the modal, the names of the buttons in the modal, and the size of the modal screen. You can also set actions for the modal. When you finish configuring the modal, close it. Notice that the modal you created sits above the body of your page structure in the content tree.

7. Bind an event to a button component.
   You bind an event to the button to trigger an action. Select the button component to highlight it again, either in the content tree, or on the page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

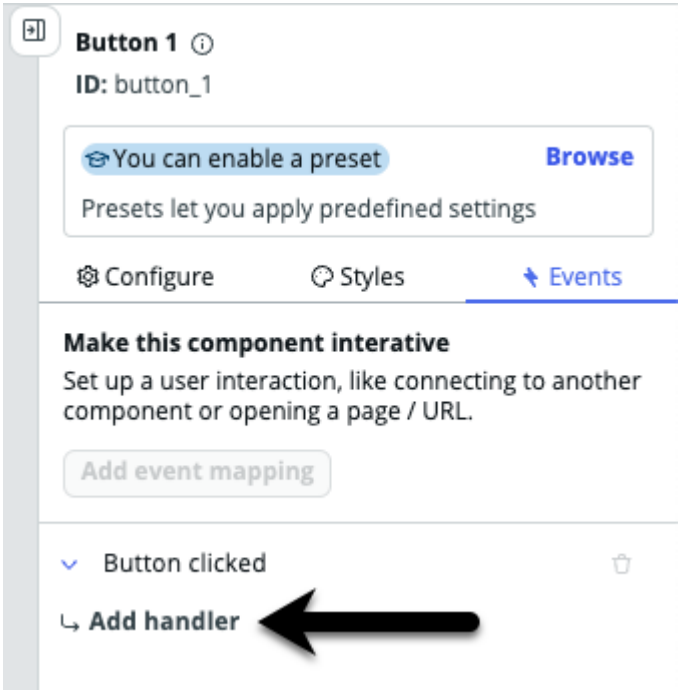
a. Select **+ Add content** in the content tree.

b. Select the **Button** component in the toolbox.

c. Select **None** when asked to choose a preset.

d. In the Configuration pane, click **Events**.

e. Select **Add handler**.

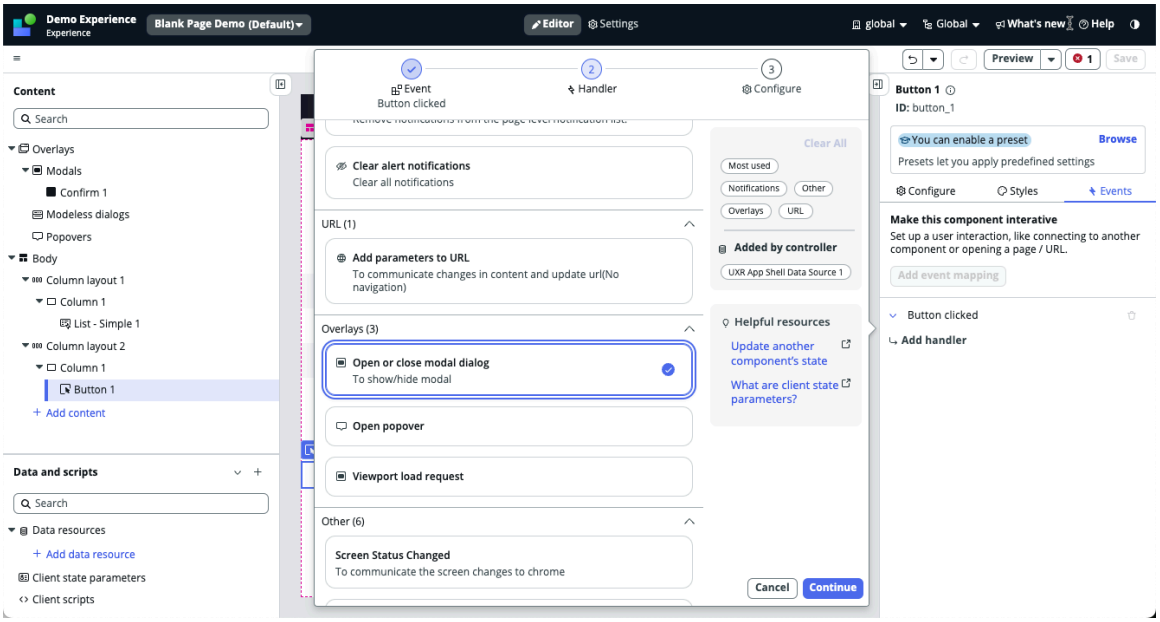


The button component only has the button-clicked event associated with it. Other components can have more than one event.

f. Select an handler that you want assigned to the button.

For example, to add an event handler for the button component, you could link it to another destination and add alert notifications. In this example, you can choose **Open or close modal dialog** to make the button open the modal that you created earlier. In this button scenario, select the Confirm and destroy modal dialog event handler that you created earlier.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



g. Select **Continue**.

h. Select a modal from the drop-down.

i. Select **Add** to add the modal event handler to your button component.

8. Click **Save**.
9. Test the modal by selecting **Preview**.
10. To trigger the modal that you created, click the button on the page.

Bind an event to a page

Use page event mappings to bind data elements within UI Builder so that you can add event actions to your page.

Before you begin

Role required: ui_builder_admin

About this task

You can bind an event to a page by using the following types of events:

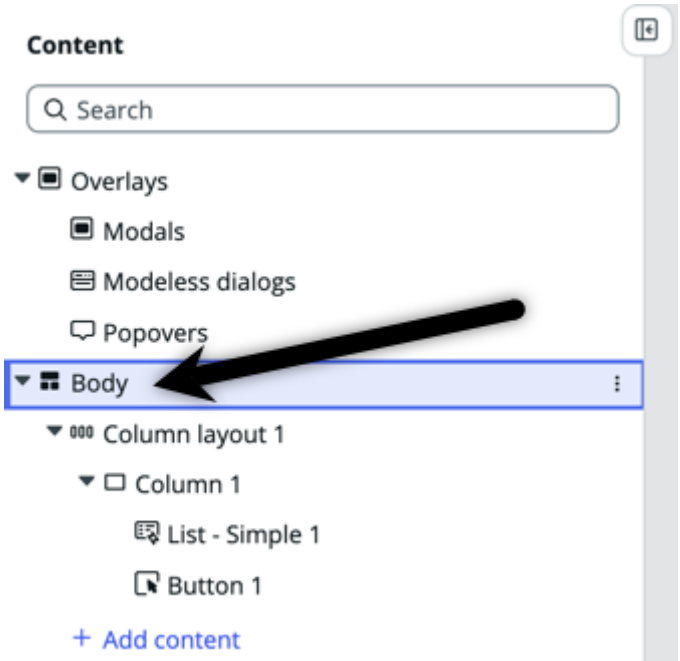
- Page event mappings. Add, remove, or clear alert notifications on your page.
- Variant event mappings. Add, remove, or clear alert notifications on your page variant.
- Dispatched events. Create dispatched events for your page to create relayed event mappings that model events after a parent event handler. Select a target parent event handler to model the payload fields after it.
- Handled events. A handled event is an event that is exposed and available for use by other users. After you create a handled event, it is available under *`Page event mappings`* for other users to use. You can also set up payload fields that you create manually or choose a template for your handled event, such as an open or close a modal dialog.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information.
3. Open or create a page.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. Highlight the page body in the content tree.

   The body is the top-level line of the content tree. When you highlight the whole page, you can add page-level events.



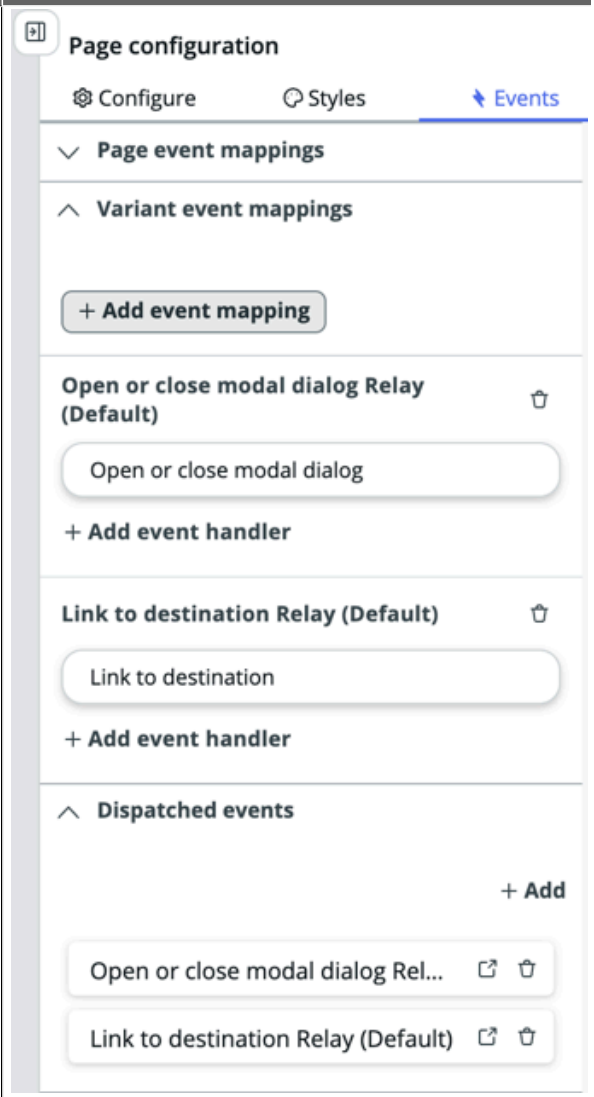
5. In the configuration panel, select the **Events** tab.
6. Add an event handler in one of the following ways.

To add this event handler	Do this
Page event mappings	*Page ready* source event: **a.** Select **Add event handler**. **b.** To add, remove, or clear alert notifications from the page, choose an event handler from *Page-level event handlers*. You can add, remove, or clear alert notifica tions on your page. **c.** To add the event handler to your page, se lect **Add**. *Page property changed* source event: **a.** Select **Add a new event handler**. **b.** Choose an event handler from *Page-level event handlers*. You can add, remove, or clear alert notifications on your page. You can also choose **Set loading state** to show the word **Loading** on your

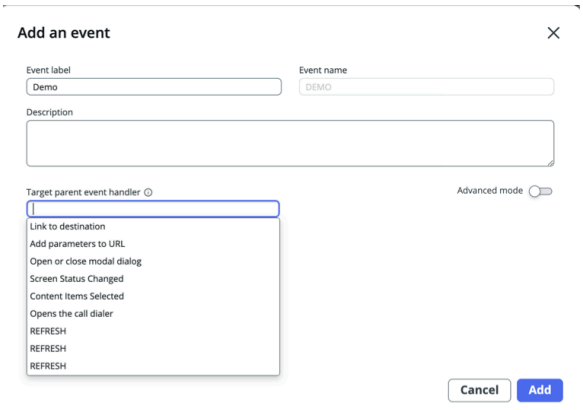
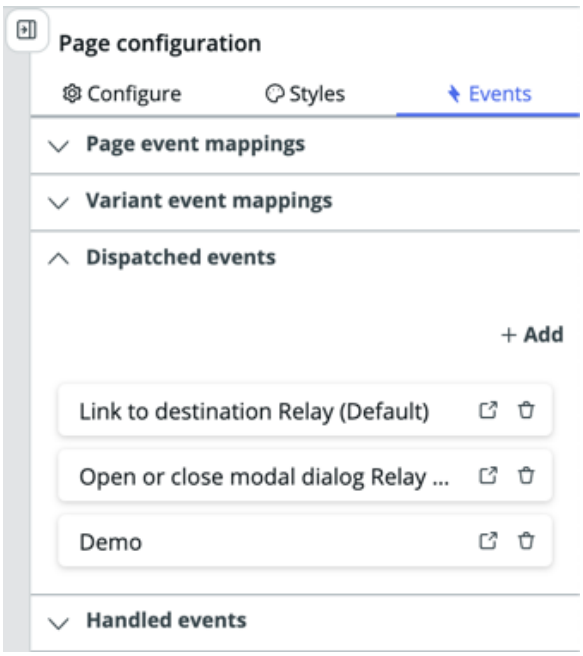
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

To add this event handler	Do this
	page so users can see that the page is load ing. You can also select **Update client state parameter** to change the client state name. **c.** To add the event handler to your page, click **Add**.
Variant event mappings	**Variant event mappings** You must have a variant page to perform a variant event mapping. The Variant event mappings that are available depend on the variant page. If the variant page has a list, you add event handlers for things like Row clicked, Nav item, and so on. When the vari ant has a Viewport, you may have different event handlers to set up. **a.** Select **Add event mappings**. For example, you can add a new event handler for **Row clicked**. **b.** Select an event handler from the list of available Inherited event handlers or local event handlers like the UXR App Shell Data Source. **c.** To add the event handler to your variant, se lect **Add**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

To add this event handler	Do this
	 ○ These event mappings are in the variant record that are found in the `sys_ux_screen` table. ○ The source events for these event map pings are any dispatched events that are defined for your page. ○ The available handlers for these event map pings are canvas-level events, the UXR App Shell Data Source, and operations from in herited data resources. ○ These event mappings are used as relays so that events that are dispatched from components on your page are relayed up to the user session handlers. ○ These mappings are automatically creat ed on a page save when you have mapped a component's dispatched event to anoth er event. For example, when you map a *`Button clicked`* event to the *`Link to`*

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

To add this event handler	Do this
	destination event, the event creates a dispatched relay event and a variant event mapping after you save the page.
Dispatched events	**a.** Select **Dispatched events**. **b.** Select + Add. **c.** In the **Add an event** screen, do the following actions: ▪ Type a label for your event. ▪ Select a target parent event handler. **d.** To add the event handler to your dispatched events, select **Add**.  **e.** The new event handler that you created appears in **Dispatched events**. The following example shows how to add a *Link to destination* event handler. 

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

To add this event handler	Do this
	○ These events are in the [sys_ux_event] ta ble. The page definition record contains ref erences to the [sys_ux_event] record. ○ These events serve as source events for variant event mappings. ○ To create new dispatched events for your page, select + Add. Configure the event label to auto-populate the event name and pay load fields for your new `sys_ux_event`. ○ These events are often used to create relay event mappings. You can select a *`Target parent event handler`* to model the payload fields after the selected event. You can also manually create payload fields for your dispatched event.
Handled events	○ These events are saved in the [sys_ux_ event] table. The page definition record contains references to the [sys_ux_event] record. ○ These events serve as source events for Page event mappings. ○ To create new dispatched events for your page, select + Add. These events are not mod eled after parent event handlers. Payload fields for handled events are created man ually. To use an existing handler's payload fields as a template, select a template and edit the fields as necessary.

7. Select **Save**.
8. To preview your page and test the data resource event that you set up, select Preview.

Bind an event to a data resource

Assign event handlers within UI Builder to individual data resources on your page. When a data resource successfully fetches new data, it executes an event handler to take a user to the next step in a flow.

Before you begin

Role required: ui_builder_admin

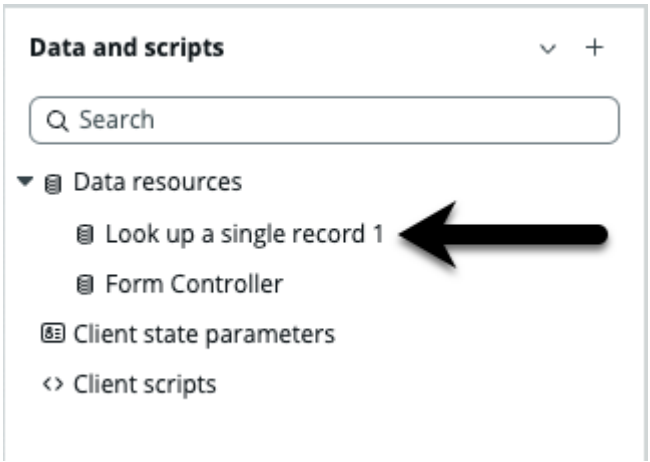
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

About this task

Bind an event to a data resource, to perform data-related actions on your page. For example, you can add a button to reload data to a page if it doesn't load the first time. You can also set up a data resource event to reload data behind the scenes of your page.

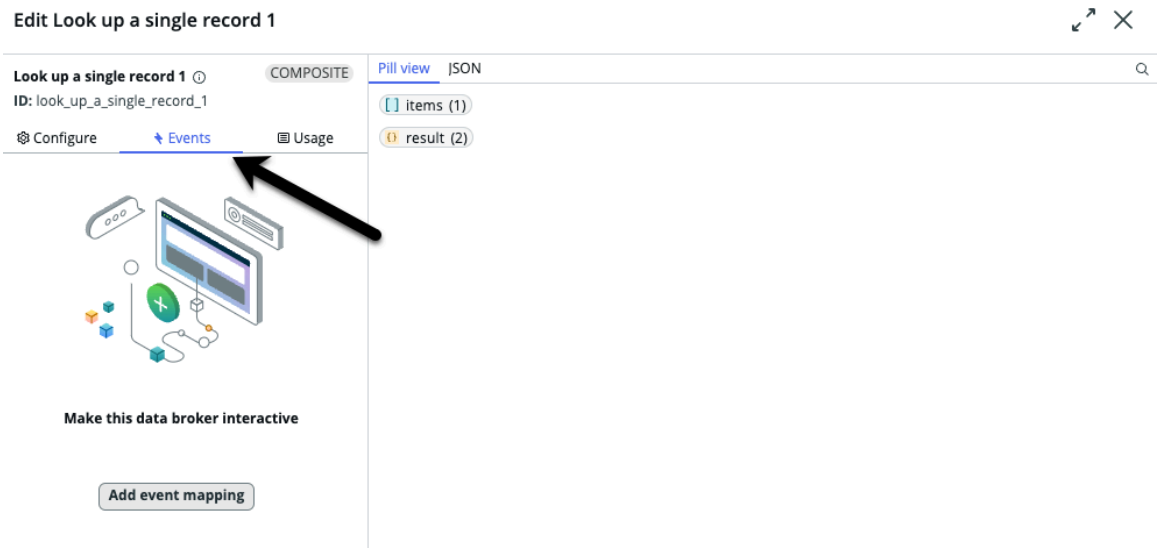
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   For more information, see Configure how users interact with your applications in UI Builder.
3. Create or open a page.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. **Optional:** If you do not have any components on your page, add one to your page.
   For more information, see Add and configure components.
5. Add a data resource to your page.
   For more information, see Add and configure data resources to a page.
6. Select a data resource instance.



7. To add an event handler to your data resource, select the **Events** tab in the configuration panel of the data resource.

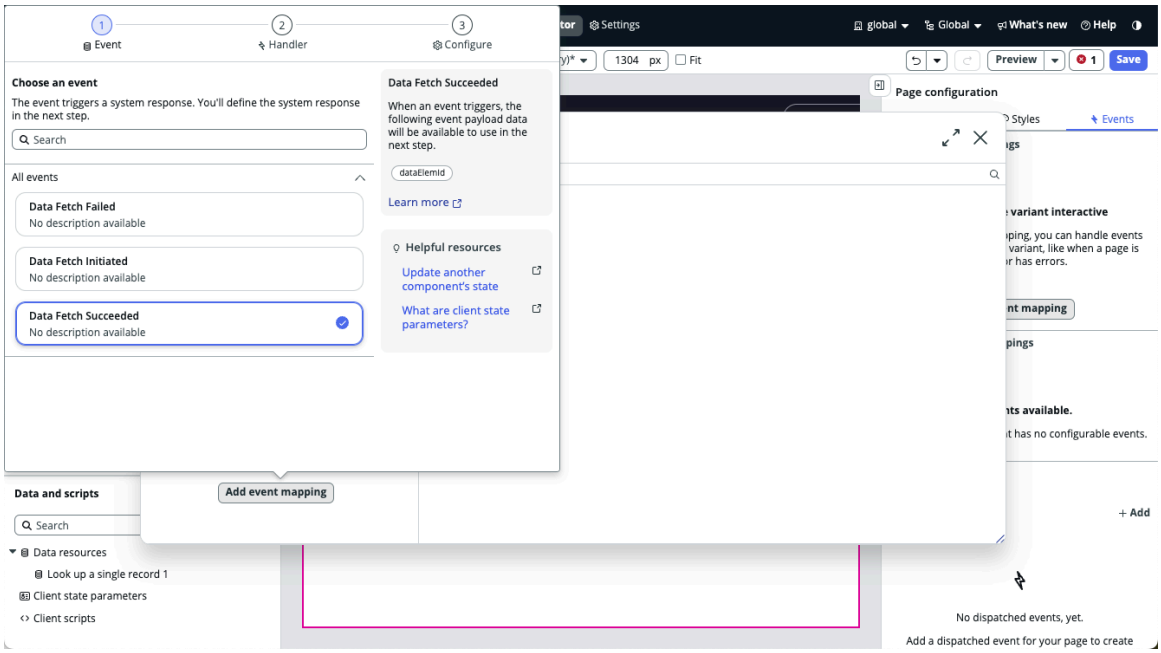
   You can add event handlers to your data resource to handle actions for Operation Initiated, Operation Succeeded, and Operation Failed.



8. Select **Add event mapping**.

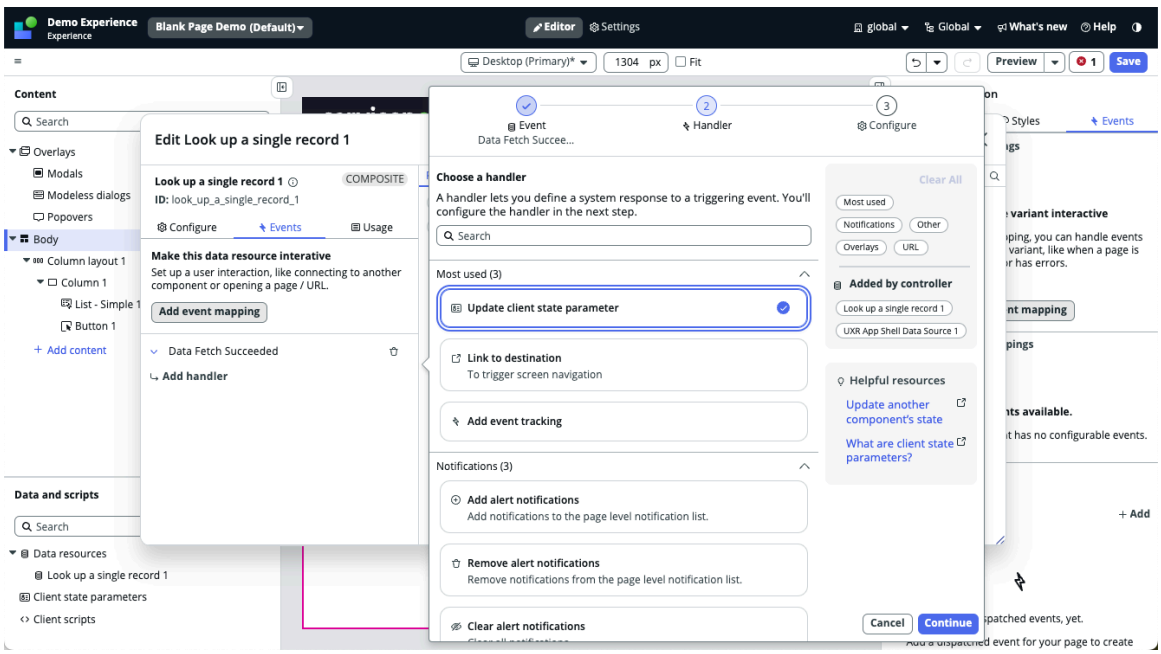
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

9. Select an event from the list.



10. Select **Continue**.

11. Select an event handler from the list.



12. Select **Continue**.

13. Fill in the required information for the event handler.
    The configuration varies depending on which event handler is selected.

14. Select **Add**.

15. Select **Save**.

16. To preview your page and test the data resource event that you set up, select  in the UI Builder header.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Link an event to another page

Add a link to the destination event handler within UI Builder so that an event action can open another page. You can also configure the event handler to follow the App Route to the desired page.

Before you begin

You must have a workspace page that contains a component that is intended to open another page when a user clicks it. The dashboard overview is an example of such a component. Components such as **Link to Destination** do not support the link to destination event handle. The component link property takes priority over the link to destination event handler.

Role required: ui_builder_admin

About this task

To configure an event action to open another page, you must know what page you want to open, what the required and optional parameters are for that page, and what payload values to set on the event handler to pass the required parameters to the destination page.

Tip: You may be able to find examples of both the components that you want to link from and the destination pages that you want to link to in the Base Agent Workspace Experience. This Next Experience is provided in the base system. If you create a page from a page template, you should only copy the contents of the template. Do not reference it. For more information about the difference between copying and referencing a page template, see Create a page from a template.

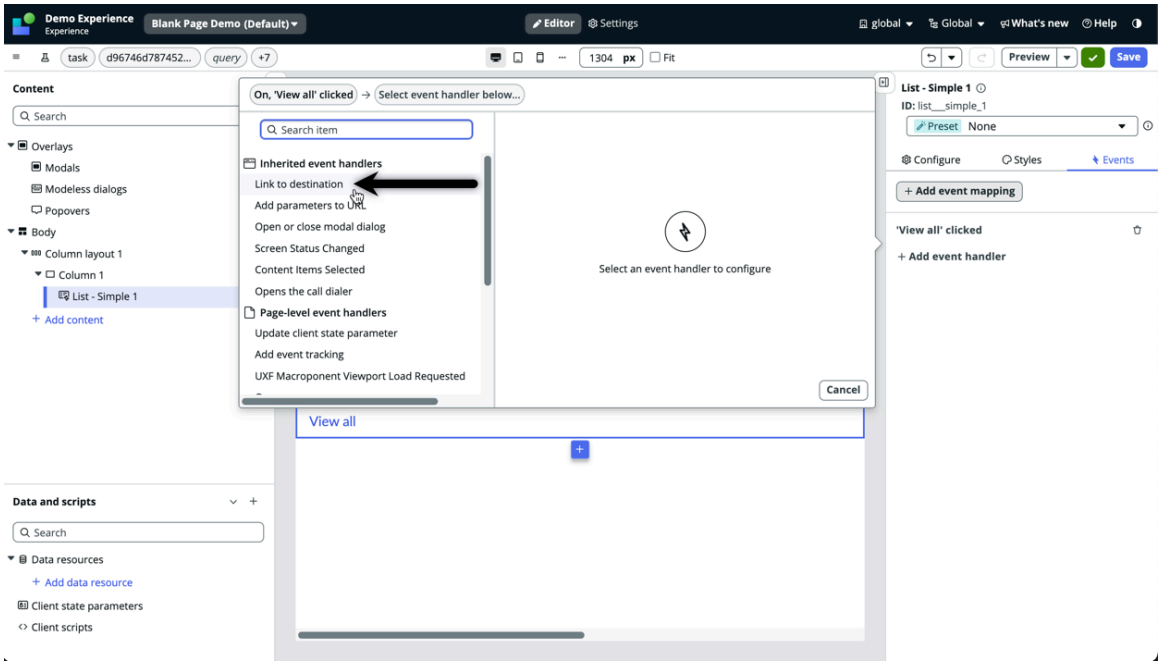
Procedure

1. Open your experience in UI Builder.
2. **Optional:** If the destination page doesn't exist in your experience, create one.
   For information about creating pages, see Create a page in UI Builder. Make sure that you set the required and optional parameters for the page so that you can use it as a destination. If a particular component in the page is a destination, you must include that component. You also must configure the properties on the component to consume the page parameters with `@context.props.<parameter-name>` values.

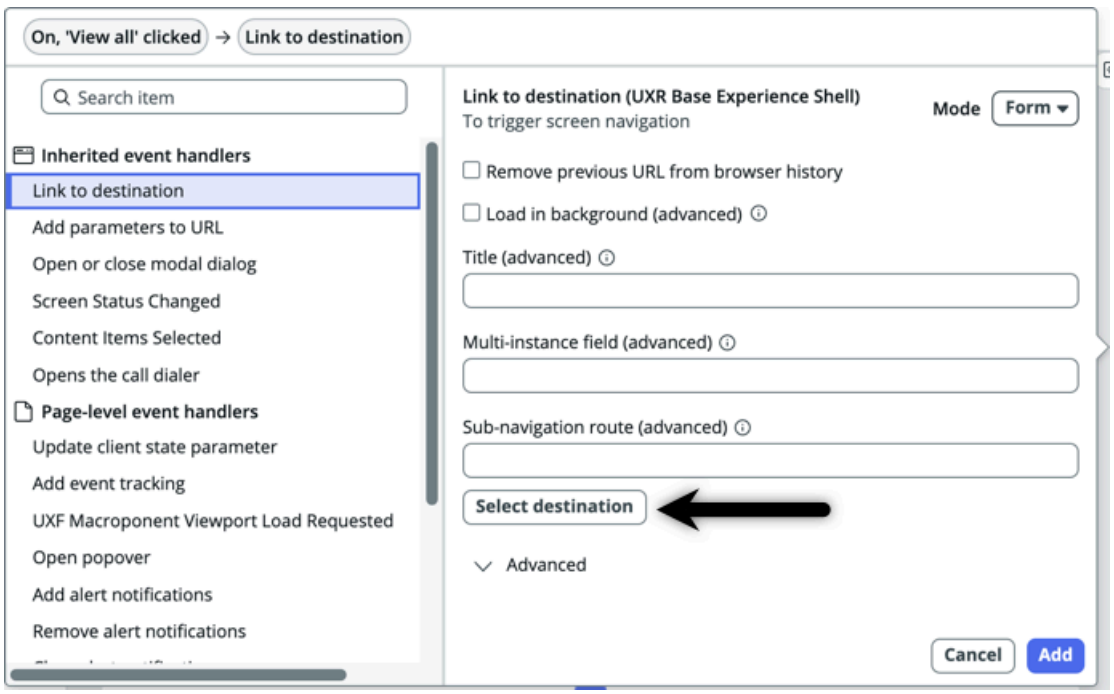
   You might consider creating the page from a page template. The Base Agent Workspace Experience has several page templates that are already configured to be destinations for other components. If you create a destination page from a template, the components are already configured with the correct properties. Any necessary state parameters or client scripts are also copied over. You have to add the page parameters. You can copy these parameters from the UX App Routes related list on the Agent app config [sys_ux_app_config] record of the experience that contains the page templates.

   To make sure that the pages that you are creating work reliably as destinations in your experience, your experience must have the same app shell UI as the experience with the page templates.
3. Switch to the page that you want to link to the destination page.
4. Navigate to the relevant component and select it.
5. Select the **Events** tab.
6. Select **+ Add event mapping**.
7. Select the event you want to use.
8. Select **+ Add event handler**.
9. In the Inherited event handlers section, select **Link to destination**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



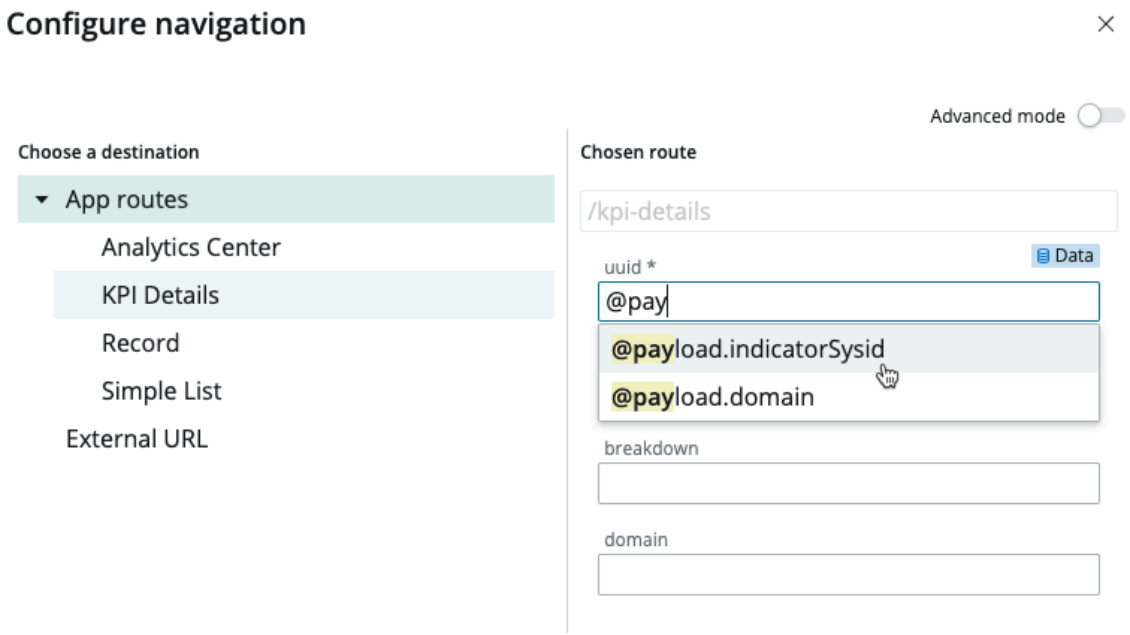
10. Click **Select destination**.



11. Expand **Pages** and select the page in the experience that you want to link to.
    Fields appear for each of the parameters on the destination page that the route leads to.
    Required parameters are marked with an asterisk (*).

12. Complete each required parameter field and the applicable parameter fields with an appropriate `@payload.*` value.
    If the developers of your component included default payload values in your event, you can select one through autocompletion. As shown in the following example, the payload value may not match the parameter name.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Note: You have the option to link to an External URL instead of specifying an **App route**.

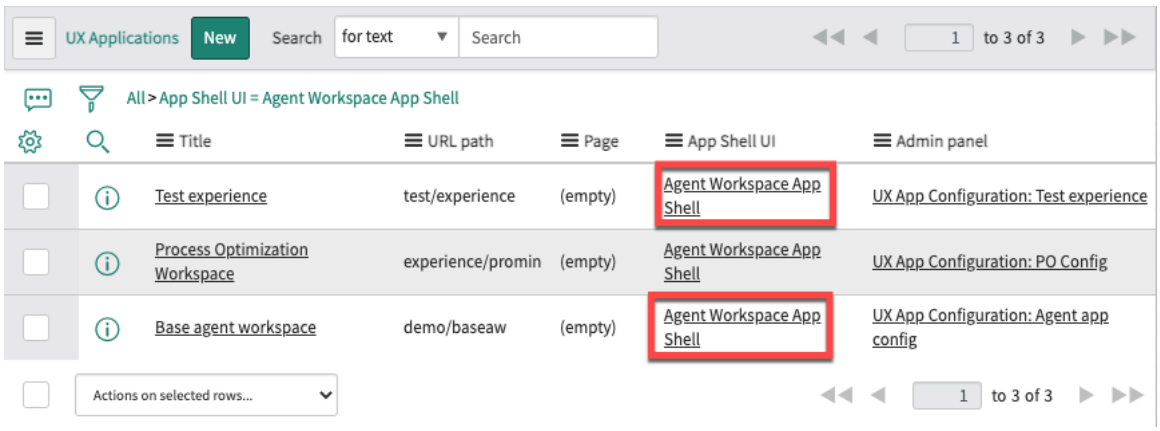
If no default values are provided, or you can't determine which values are correct for some fields, refer to the configuration and API documentation for the component in the . If you still can't find the necessary `@payload.*` values, contact Customer Service and Support.

Tip: If you create your linking component by creating a page from a Base Agent Workspace page template, the component contains Link to destination Relay event handlers. These event handlers do not work. However, they contain the applicable `@payload.*` values for the parameters.

Example: Configuring the event handlers for an Analytics Q&A component

Let's say that you want to take a new Next Experience and add a page with an Analytics Q&A component. First, you create the page from the Analytics Center page template that is provided in the Base Agent Workspace experience. Next, you create a target page for the first of the three events in Analytics Q&A and then you configure an event handler for that event.

By navigating to **Now Experiences Framework** > **Experiences**, you see the Test experience UX application. Because it uses the same Agent Workspace App Shell UI as the Base Agent Workspace, you can use the page templates from the Base Agent Workspace.



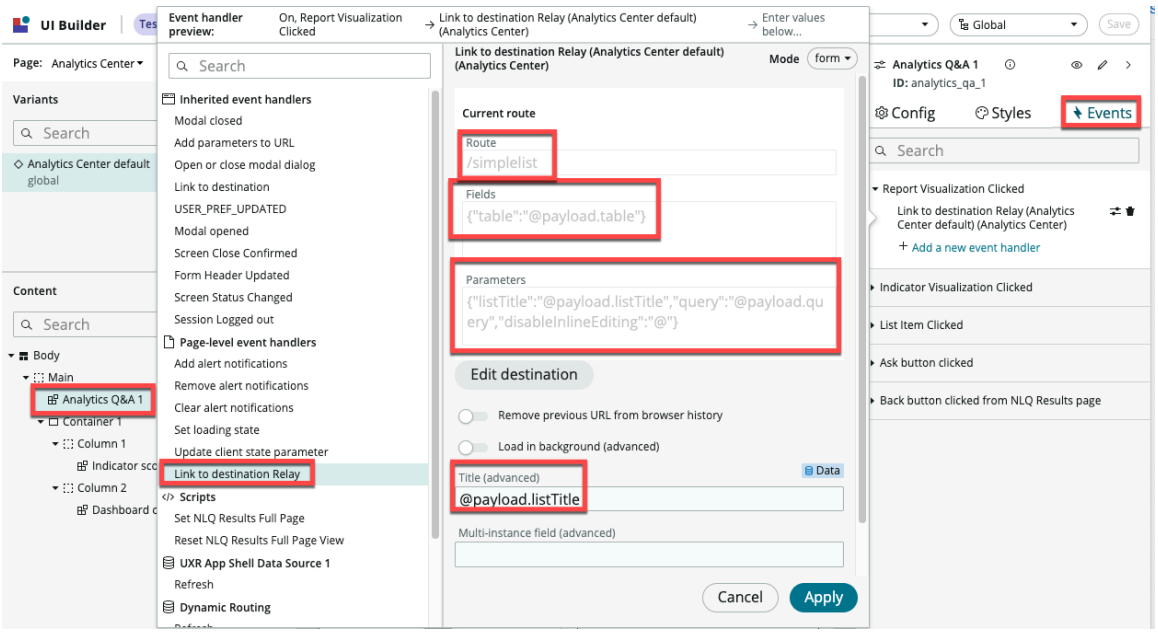
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

You next select the Test workspace admin panel, find an UX App Configuration record with no UX app routes or pages, and then click **Open**.

You select the Analytics Q&A 1 component and open the **Events** tab. From here, you can open the**Link to destination Relay** event handler for the **Report Visualization Clicked** event. When a question in Analytics Q&A returns a report, you can trigger this event by clicking a value in the report. When you click a value, you also see a list of the records that contribute to this value. In the **Route** field, you see that the destination is expected to be a page that is based on the Simple List page template. You also see the parameters of the page that the `@payload.*` values correspond to, and that the **Title** field can be populated with `@payload.listTitle`.

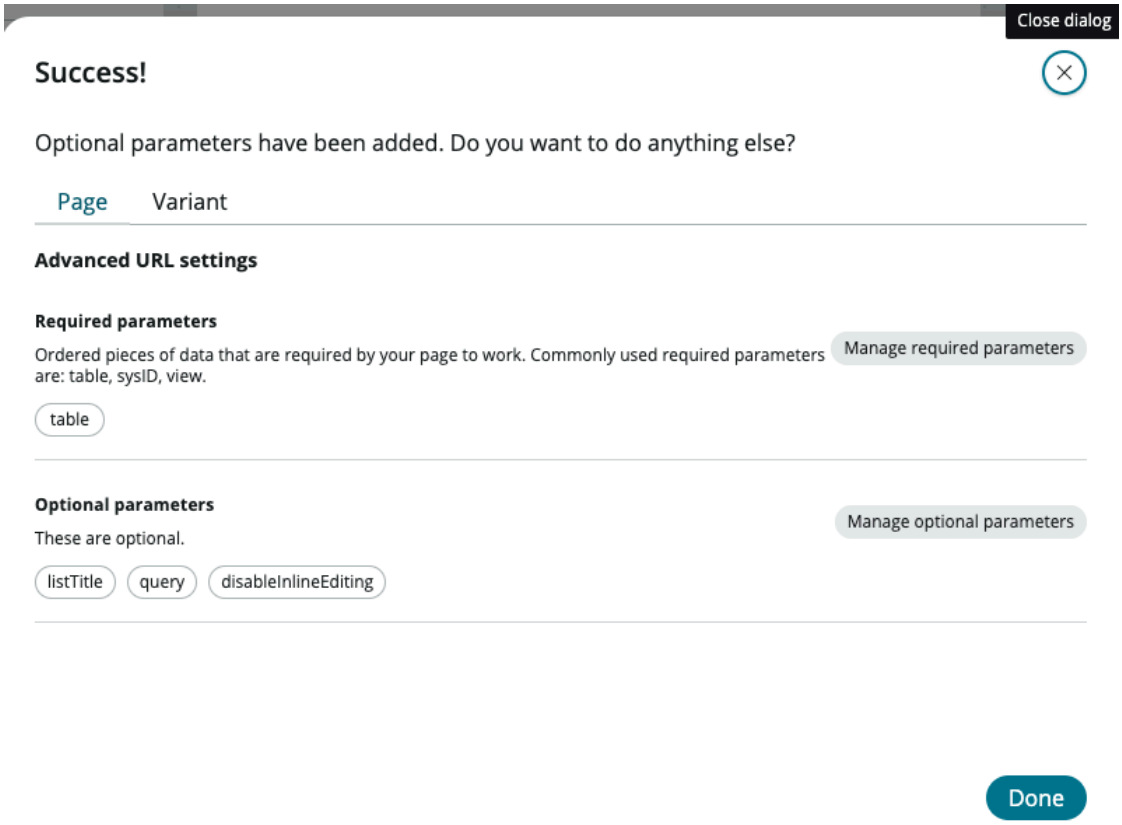
Parameters of suggested Simple List destination page and corresponding payloads

Parameter	@payload.* value
table (required)	@payload.table
listTitle	@payload.listTitle
query	@payload.query
disableInlineEditing	none

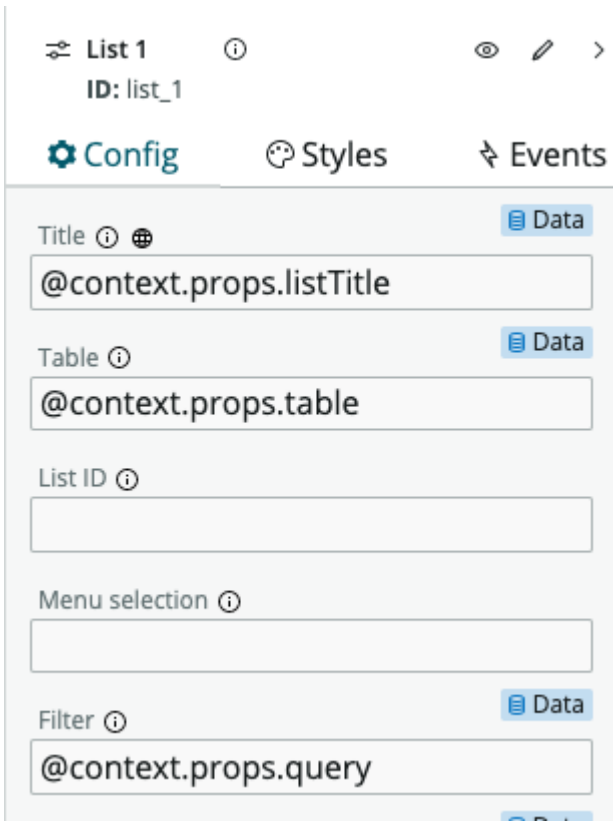


Next, you navigate to **Menu** > **Create page** and create a page that is based on the Simple List template. Let's say that you name the page as Record list. You then follow a similar process as when you created the Analytics Center page. This time, in the last steps of the process, you would add `table` as a required parameter and `listTitle`, `query`, and `disableInlineEditing` as optional parameters.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



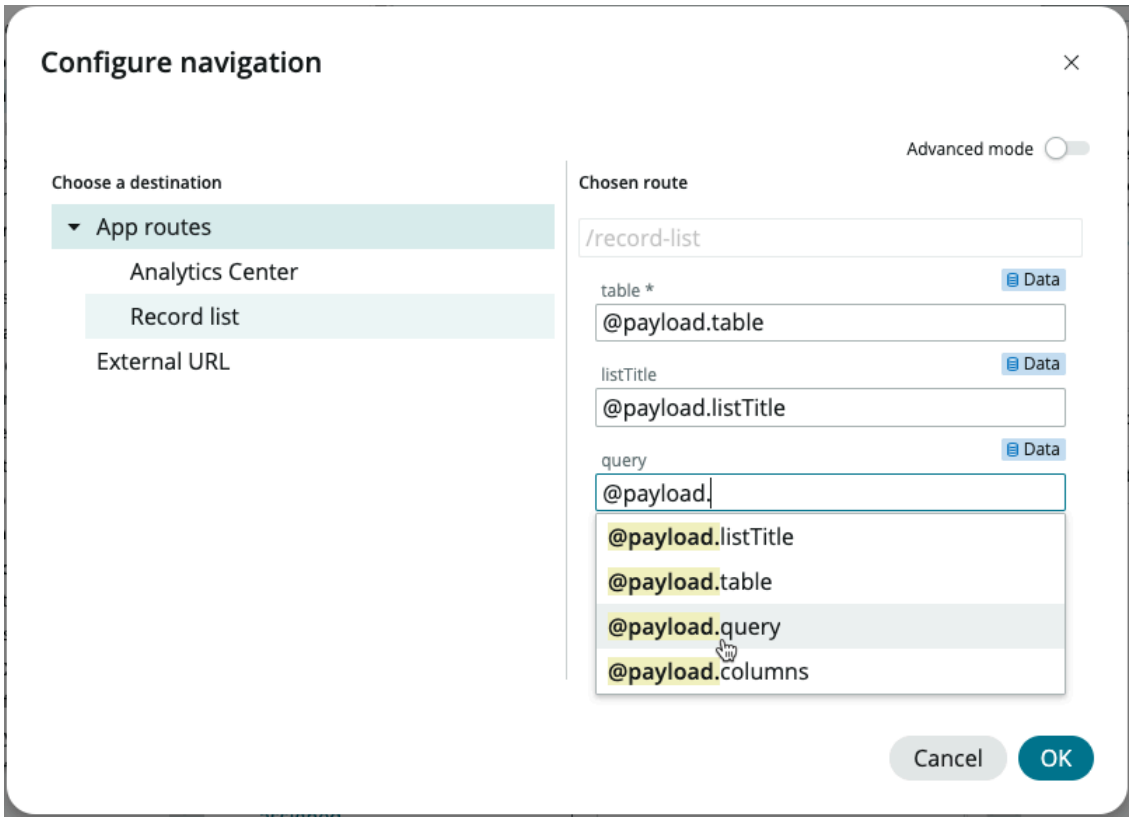
Because this page already contains a List component, when you open the **Config** tab for this component, you see that the parameters are already passed in the `@context.props.*` values.



Now, you return to the Analytics Center page. In the **Report Visualization Clicked** event, you add a new event handler. Next, you select the Record list page that you created and add the

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

`@payload.*` values in the **table**, **listTitle**, and **query** fields, following the information that you got from the **Link to destination Relay** event handler. Predictive typing helps you to fill in these fields.



After you click **OK** and add `@payload.listTitle` as the **Title**, the event handler is done. You can now delete the**Link to destination Relay** event handler for this event.

Bind an event to a declarative action

Bind data elements within UI Builder so that you can add event actions to a declarative action.

Before you begin

Role required: ui_builder_admin

About this task

Bind a handled event to a component so that an action is performed when a user selects the component.

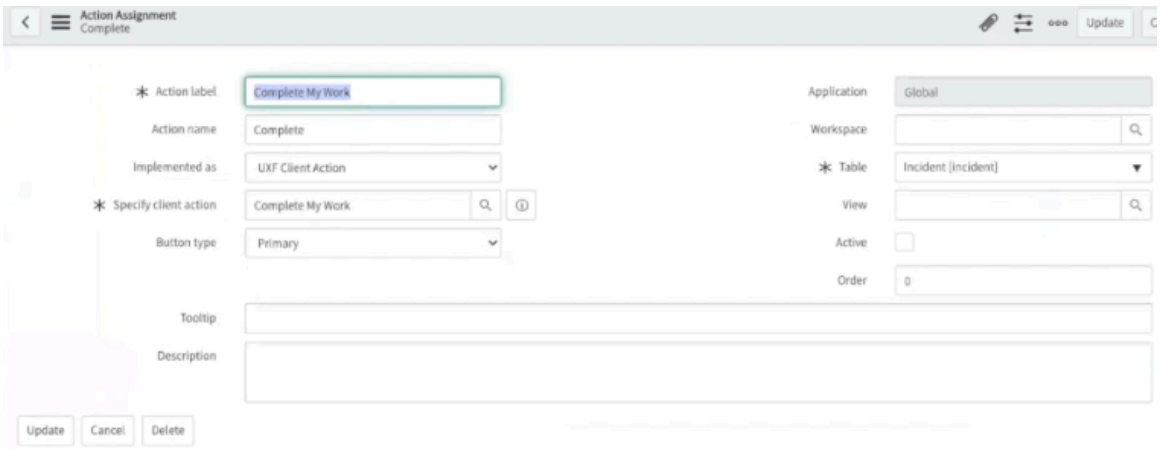
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   For more information, see Configure how users interact with your applications in UI Builder.
3. Open or create a page.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. Add a component to your page that can have a declarative action, such as an action bar or related list.
   For more information, see Add and configure components.
5. To create a declarative action definition in a table in the ServiceNow AI Platform®, navigate to **Workspace Experience** > **Actions & Components** > **List actions**.

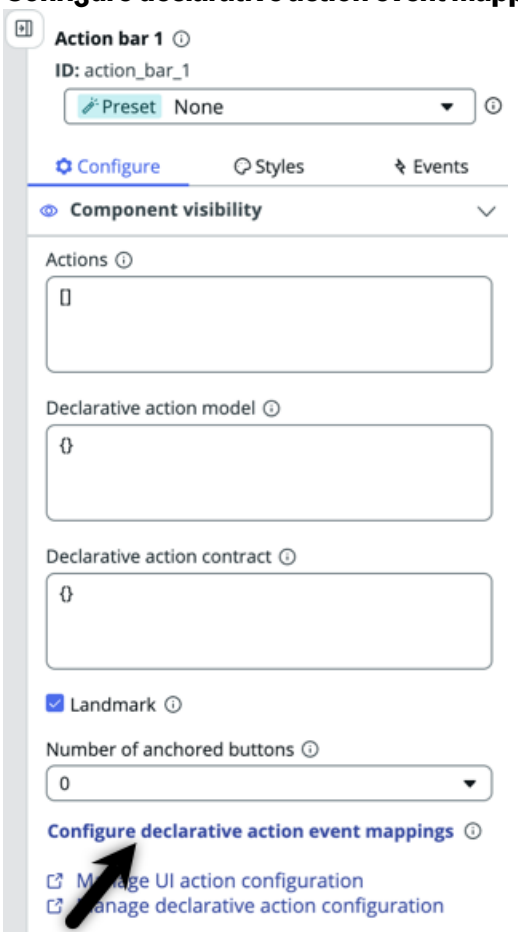
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Choose a table where you want the declarative action to be available in. For example, you could create a Complete my work action in an incident table or you could use an existing declarative action definition record.

Platform declarative action definition record



6. Select the **Active** check box, and then save or update the record.
7. Return to the UI Builder.
8. To invoke a handled event for the declarative action, go to the Configure panel and click **Configure declarative action event mapping**.



9. Choose the declarative action that you created earlier.
   To continue with the example in step 5, the declarative action could be something like **Complete my work**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

10. To define what the declarative action does on your page, click **+ Add event handler**.

    a. Give the event handler a name.
    The name should describe what action you want the event handler to perform.

    b. **Optional:** Provide a description of the event handler.

    c. Choose the handled event that you want to invoke.

    d. **Optional:** Add payload values for your event handler.

    e. Select **Save**.

11. Select **Done**.
12. Select **Save**, and then select Preview.
13. Test the declarative action on your page by clicking **Complete my work** to see if it works.



Disable component preset event mappings

You can disable event mappings that are automatically created by a component preset.

Before you begin
Role required: admin

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   For more information, see Configure how users interact with your applications in UI Builder.
3. Open or create a page.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. Select a component with a component preset configured.
5. Select the **Events** tab in the configuration panel.
6. Select the event that you want to disable.
7. Select **Disable**.
8. Select **Done**.

Result
The component preset event shows **Disabled** in the **Events** tab of the configuration panel.

Delete an event handler

Delete an event handler that you no longer need in UI Builder.

Before you begin
Role required: admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   For more information, see Configure how users interact with your applications in UI Builder.
3. Open or create a page.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. Select a component with a preset configured.
5. Select the **Events** tab in the configuration panel.
6. Select the event handler that you want to delete.
7. Select **Delete**.
8. Select **Delete** in the confirmation modal.
9. The event handler is removed from the associated event mapping.

   If you would like to delete the event mapping, see Delete an event mapping.

Delete an event mapping

Delete an event mapping that you no longer need in UI Builder.

Before you begin
Role required: admin

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   For more information, see Configure how users interact with your applications in UI Builder.
3. Open or create a page.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. Select a component with a preset configured.
5. Select the **Events** tab in the configuration panel.
6. Select the delete all icon ( ) next to the event you want to delete.
7. The event mapping is removed from the associated component.

Add events to track components with unsaved changes

Locate dirty state components through event configuration.

Before you begin
Role required: ui_builder_admin

About this task
You can add the **Screen status changed to dirty** event to check which components have unsaved changes. This avoids having to manually inspect advanced properties on events for each individual component. Once added, the event remains available in the configuration panel for direct access.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

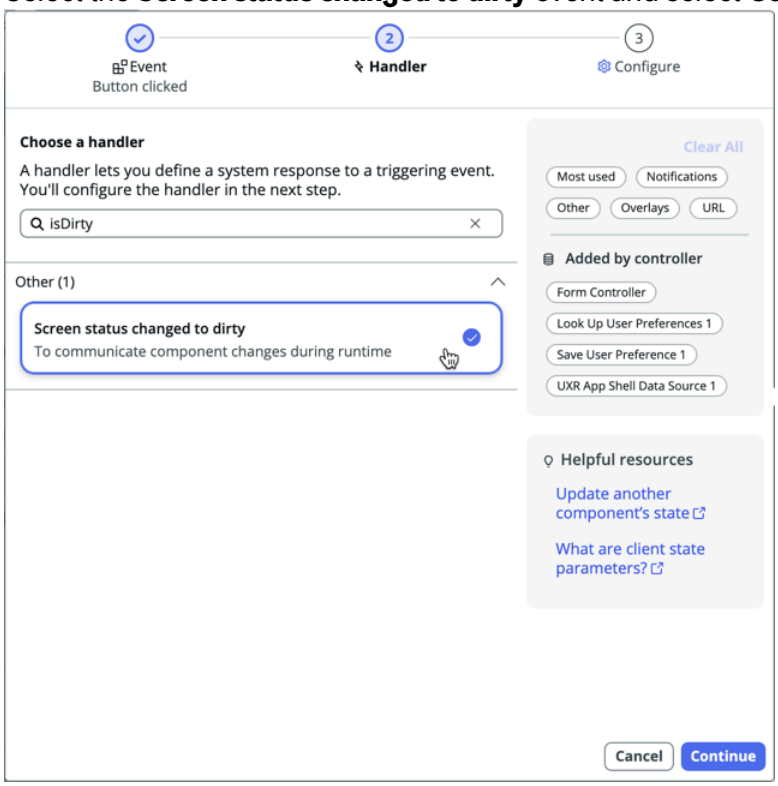
1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create or open a page.
4. In the content tree, select **+ Add content** to add a component.
5. Select a component from the toolbox and select **Add**.
   For example, select a **Button**.
6. Add an event to the component.

   a. Select the **Events** tab in the configuration panel.

   b. Under **Button clicked**, select **Add handler**.

   c. Under **Choose a handler**, enter `isDirty` (or `dirty`) into the search bar.

   d. Select the **Screen status changed to dirty** event and select **Continue**.



   e. Select the **Screen is dirty** check box to mark the property as `true`.

   f. Select **Add**.

The dirty state event now appears in the configuration panel, enabling you to quickly check the `isDirty` property and determine if the component has unsaved changes without adding additional events.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

UI interactions

Trigger UI interactions directly from events in UI Builder, allowing you to link event-driven behavior to reusable interaction logic.

UI interactions are reusable flows that combine UI elements and logic into a single unit you can trigger from component events in UI Builder. They help you avoid repeating the same wiring across multiple pages and make your designs easier to maintain.

Why use UI interactions

- **Reduced duplication**: Instead of creating separate actions for every component, you define the flow once and reuse it.
- **Improve consistency**: Interactions apply the same behavior across multiple pages or experiences.
- **Simplify maintenance**: Changes to an interaction automatically apply wherever it's used.

Note: For record-based, sequential UI with a long-running process, consider using Playbook.

Practical uses for UI interactions

- Trigger a confirmation dialog when a user selects a button
- Show an alert message after a record is saved
- Show a form view inside a modal when validation fails
- Chain multiple steps together, such as updating data and displaying a success message

Accessing UI interactions

- To view list of all existing UI interactions from within UI Builder: select the **UI interactions** tab from the UI Builder home page.
- To create UI interactions from within UI Builder: select **Create** > **UI interactions** from the UI Builder home page.
- To view a list of all existing UI interactions from the platform, navigate to **All** > **Now Experience Framework** > **UI interactions** > **All**. The UI Builder home page opens with the UI interactions list displayed.
- To create UI interactions from the platform, navigate to **All** > **Now Experience Framework** > **UI interactions** > **Create new**. The creation modal opens in UI Builder where you define the interaction name, type, and description.

UI interaction workflow

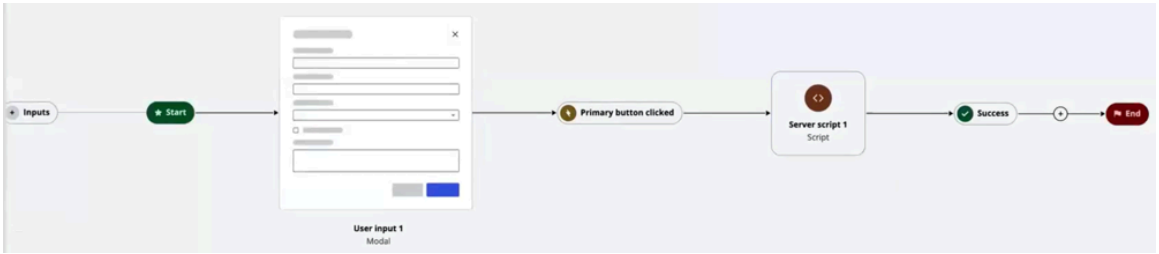
A UI interaction defines a reusable flow of user and system actions. At a high level, every interaction works this way:

1. Inputs: a configurable piece of data that a UI interaction needs to run. Supported types include string, true/false, choice, reference, and JSON. Inputs can be defined upfront or promoted from a step property.
2. Trigger: a component, declarative action, or event starts the interaction and passes inputs.
3. Start: one or more start nodes determine where execution begins.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

4. Steps: the interaction runs its steps in sequence, optionally branching with events or parallel And paths.
5. End: execution reaches an End node, completing the flow.

UI interaction workflow example



How UI interactions differ from declarative actions

Declarative actions determine where and when an action appears, most commonly on a form or list. UI interactions encapsulate what happens and can be triggered by any page event or declarative action. Key differences include:

- Declarative actions place buttons on lists and forms without requiring modifications of the page.
- UI interactions define the end-to-end behavior of UI, logic, and script when triggered.
- Declarative actions can trigger UI interactions.
- Declarative actions can be implemented as UI interactions instead of UXF client actions, since they work without page ownership or modification, even when triggering UI like modals.

Triggers for UI interactions

You can run a UI interaction from several entry points, depending on the use case:

- **Declarative action**: ideal when you need a button on a form or list without page customization.
- **UI Builder page events**: trigger interactions from buttons, component events, or custom UI behavior within a page.

UI interaction types

When you create an interaction, you choose a type that determines the context it works with:

- **Generic**: works with any component
- **Form**: designed for form-specific behaviors such as save, validate, and scripting
- **List**: tailored for list-based actions such as list refresh and scripting

UI interaction toolbox

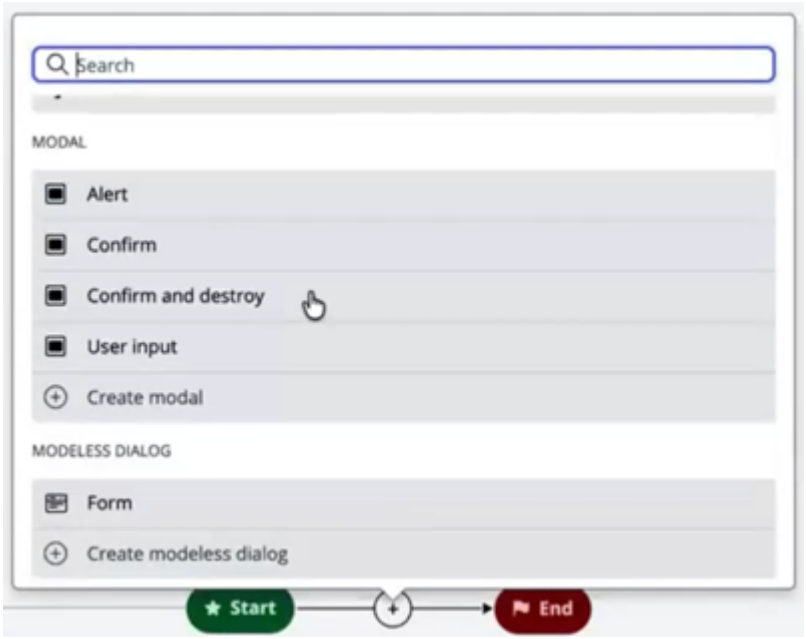
The toolbox provides all building blocks available inside the UI interaction editor, including:

- UI elements like modals, modeless dialogs, and dynamic forms
- Custom components with Component Builder
- Logic steps such as conditions, if/else branches, and navigation
- Form or list actions, when available for that interaction type
- Client and server script steps
- Data-binding inputs and configurable properties

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: Toolbox components depend on the interaction type such as form, list, or generic.

UI interactions toolbox example



Create a UI interaction

Create a UI interaction and attach it to a component event in UI Builder. UI interactions are reusable flows that combine UI, logic, and scripts into a single unit, including custom UI built with Component Builder, and can be triggered from any component event on a page.

Before you begin

Role required: ui_builder_admin

About this task

In this task, you create a UI interaction and define its behavior using the diagram editor. You select an interaction type, add and configure steps, define logic and branching, and specify inputs that the interaction requires at runtime.

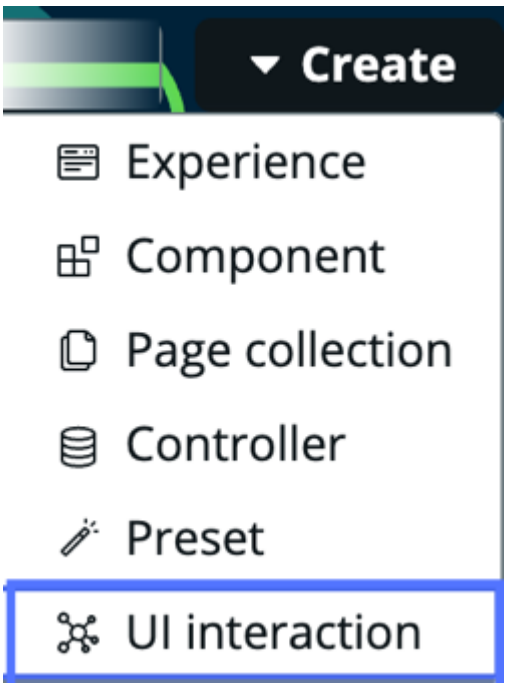
Important: UI interactions don't run on their own. After creating an interaction, you must attach it to a component or page event so it can run when that event occurs.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. From the UI Builder home page, select **Create** > **UI interaction**.

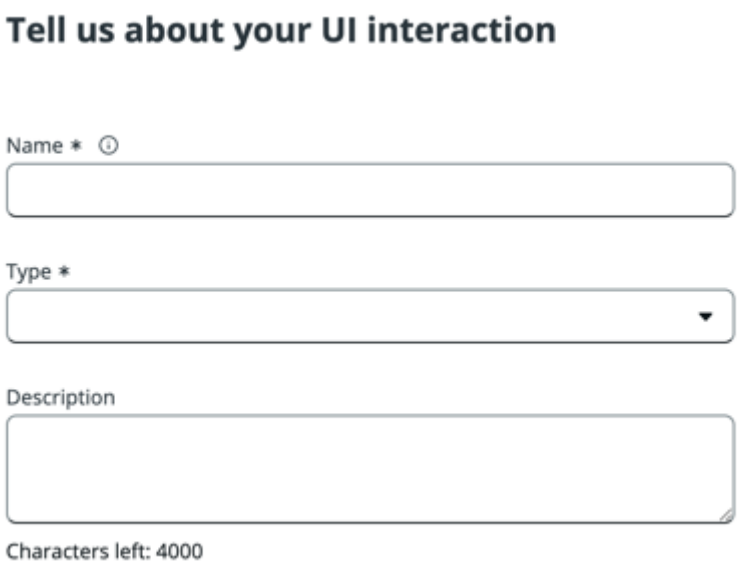
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Create UI interaction



3. In the creation modal, enter a name for the interaction.

UI interaction creation modal



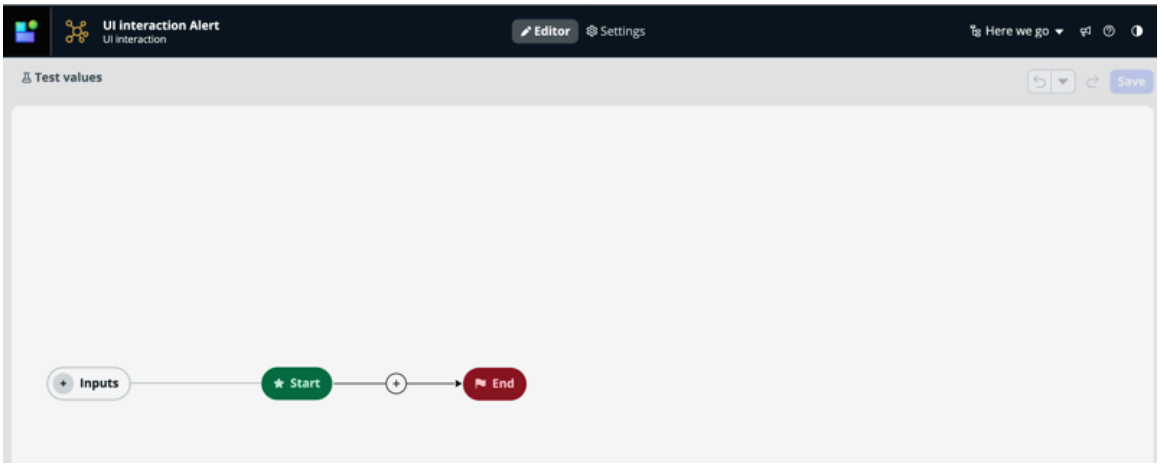
4. Select an interaction Type.

   Generic interactions are used with no dependencies on form or list controller data. Form interactions use form-specific steps such as validating or saving a record. List interactions use list-specific steps such as refreshing or querying a list. If you're unsure, choose Generic.

5. **Optional:** Enter a description that summarizes what the interaction does and when it runs. For example, `Displays a confirmation modal before deleting a record`.
6. Select **Create**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

UI interaction editor

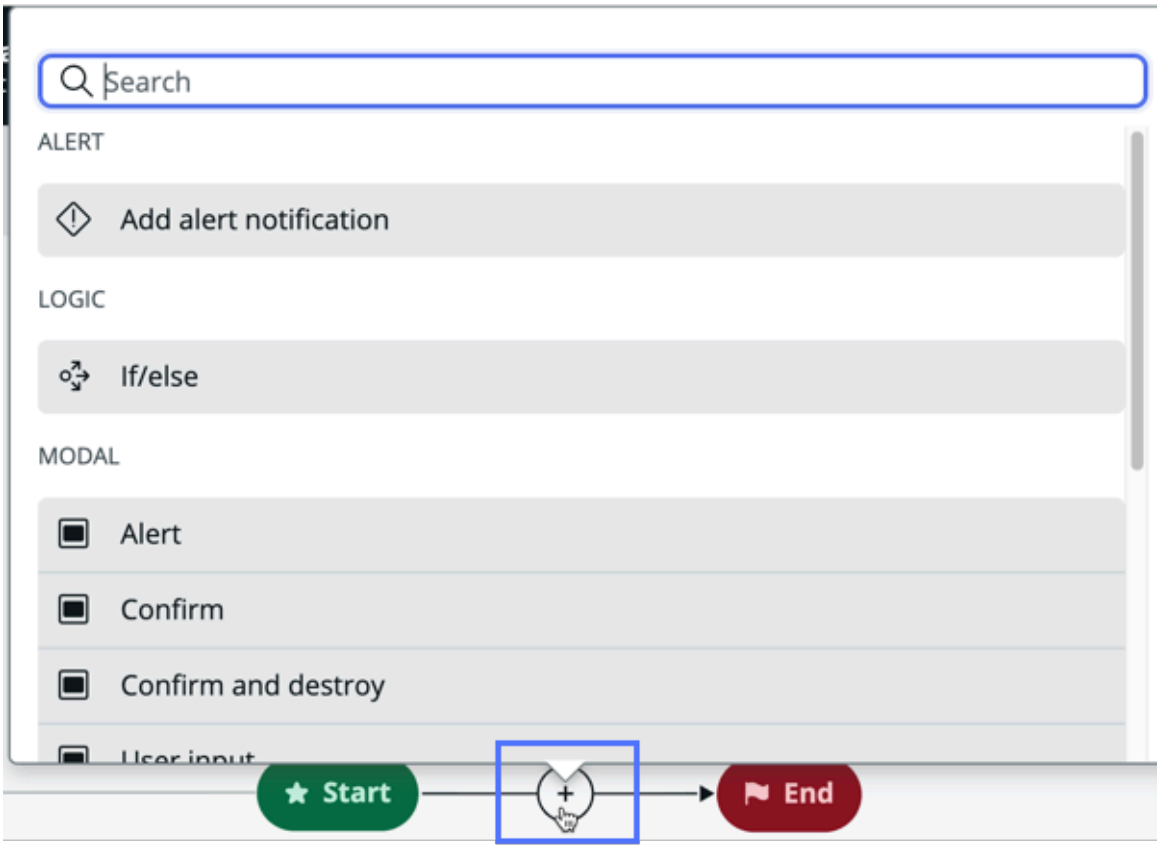


The diagram editor opens with a start and end node connected by an add icon ⊕.

7. Select the add icon ⊕ between the start and end nodes to open the toolbox, then select a step to add it to the diagram.

   Note: The steps available in the toolbox depend on the interaction type that you selected.

UI interaction toolbox



8. Select the step to open its properties in the configuration panel.
9. Configure the step properties in the configuration panel.
   Properties vary depending on the step. For example, a modal step lets you set button labels and field content, while a script step lets you write server-side logic.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

10. Repeat steps 7–9 to build out your interaction.
    Steps execute in sequence. To run steps in parallel, select the **And** step to create parallel branches. Each branch represents a separate path of steps and executes in order from top to bottom. To create conditional paths, add an **If/Else** step from the toolbox. This step creates branches that run when conditions are met. For more information, see Edit an existing UI interaction.
11. Use the Inputs pill to add any data your interaction needs at runtime.
    Supported input types are: String, True/False, Choice, Reference, and JSON.
12. Select **Save**.

Result

The UI interaction is created and ready to be attached to a component or page event. Once attached, it runs its configured steps whenever the specified event occurs. See Trigger a UI interaction from a page event.

To trigger this UI interaction from a form or list button using a declarative action, see Trigger a UI interaction from a declarative action.

Edit an existing UI interaction

Edit or modify an existing UI interaction by inserting new steps, swapping or deleting events, removing steps, or updating step configurations in the diagram editor.

Before you begin

Role required: ui_builder_admin

About this task

When you create a UI interaction, you can return to the diagram editor to modify its flow. You can insert steps before or after existing steps, swap the outgoing event that connects one step to the next, delete events or steps you no longer need, update step configurations, and add or remove branches.

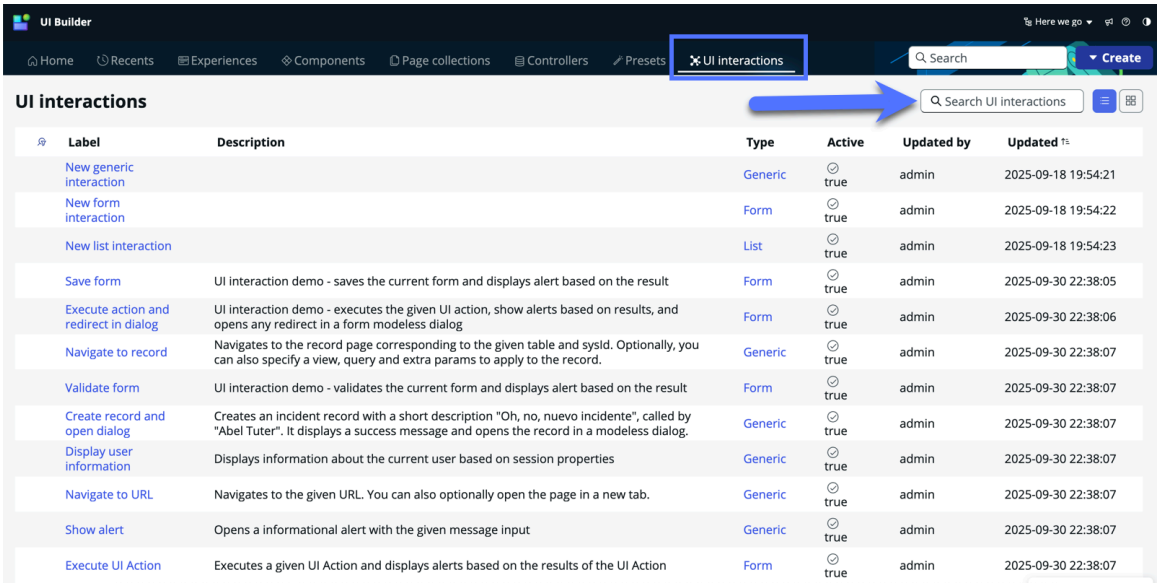
Important: Before editing a UI interaction that is already in use, check the Settings and Usage pages to understand where the interaction is triggered. Consider duplicating the interaction before making changes to avoid unintended impacts to active pages or declarative actions. For information on duplicating an interaction, see Duplicate a UI interaction.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Select the **UI interactions** tab.
3. In the search field, enter the name of the interaction you want to edit.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

UI interactions list



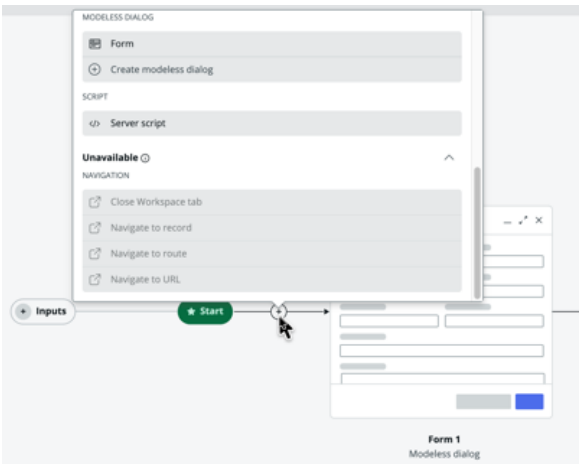
Label	Description	Type	Active	Updated by	Updated
New generic interaction		Generic	true	admin	2025-09-18 19:54:21
New form interaction		Form	true	admin	2025-09-18 19:54:22
New list interaction		List	true	admin	2025-09-18 19:54:23
Save form	UI interaction demo - saves the current form and displays alert based on the result	Form	true	admin	2025-09-30 22:38:05
Execute action and redirect in dialog	UI interaction demo - executes the given UI action, show alerts based on results, and opens any redirect in a form modeless dialog	Form	true	admin	2025-09-30 22:38:06
Navigate to record	Navigates to the record page corresponding to the given table and sysId. Optionally, you can also specify a view, query and extra params to apply to the record.	Generic	true	admin	2025-09-30 22:38:07
Validate form	UI interaction demo - validates the current form and displays alert based on the result	Form	true	admin	2025-09-30 22:38:07
Create record and open dialog	Creates an incident record with a short description "Oh, no, nuevo incidente", called by "Abel Tuter". It displays a success message and opens the record in a modeless dialog.	Generic	true	admin	2025-09-30 22:38:07
Display user information	Displays information about the current user based on session properties	Generic	true	admin	2025-09-30 22:38:07
Navigate to URL	Navigates to the given URL. You can also optionally open the page in a new tab.	Generic	true	admin	2025-09-30 22:38:07
Show alert	Opens a informational alert with the given message input	Generic	true	admin	2025-09-30 22:38:07
Execute UI Action	Executes a given UI Action and displays alerts based on the results of the UI Action	Form	true	admin	2025-09-30 22:38:07

4. Select the interaction from the search results.

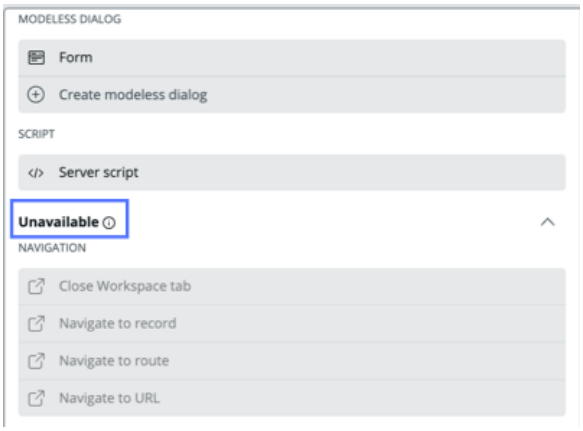
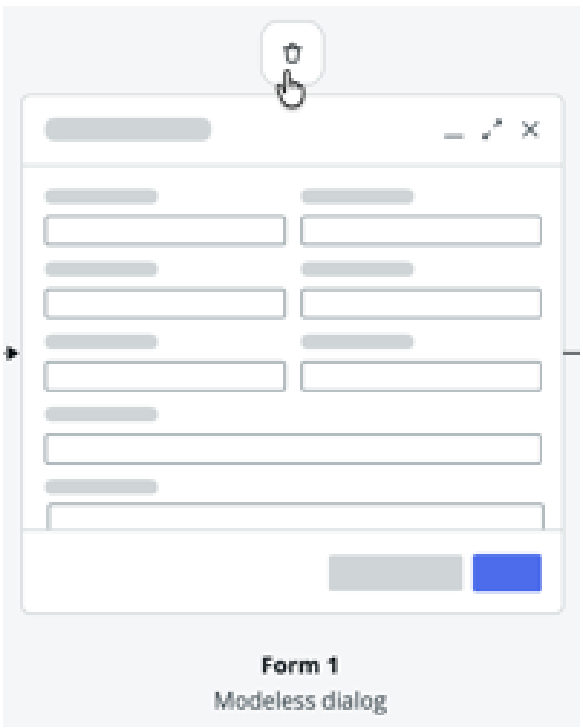
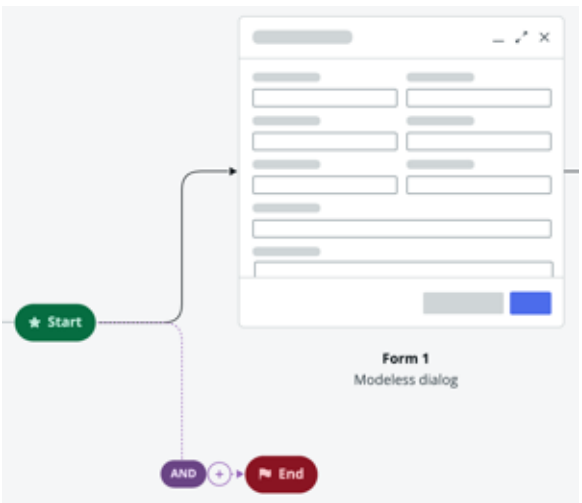
   The diagram editor opens with the existing flow displayed on the canvas.

5. Choose how you want to edit the interaction.

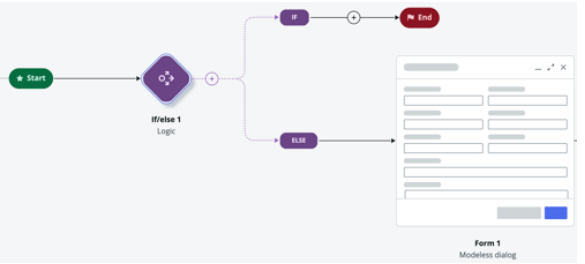
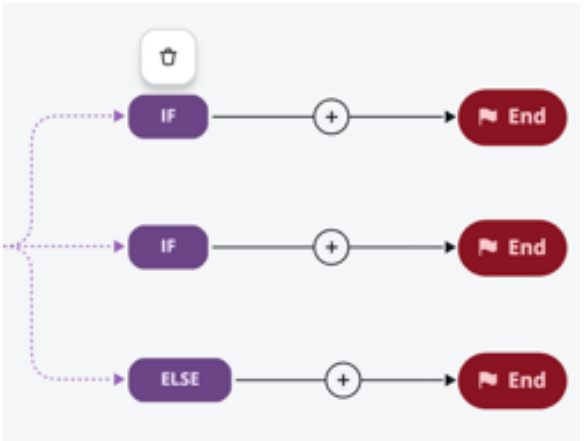
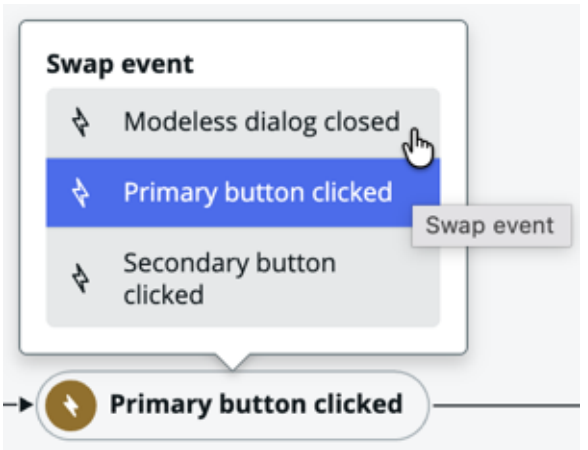
Editing scenarios

Editing scenario	Steps	UI depiction
Insert a step before or after an existing step	Select **+** before the step, then choose a step from the toolbox. The step is inserted and the topmost outgoing event is automatically assigned to continue the flow. **Note:** Steps with no outgoing events appear in an Unavailable section at the bottom of the toolbox. To add an unavailable step, use an And branch instead.	

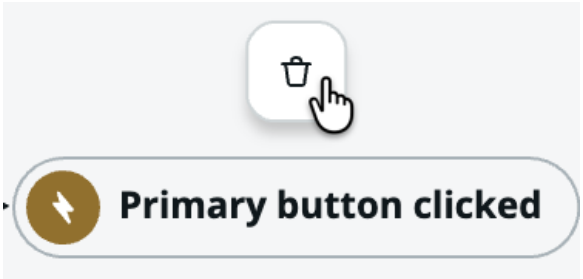
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Editing scenario	Steps	UI depiction
Work with unavailable steps	If a step cannot be added at the selected location, it appears in the Unavailable section. Review the explanation to understand why the step is not supported at that point and which alternatives are available.	
Delete a step	Select the step and select the delete icon . When a step is deleted, any dependent downstream bindings reset to their defaults, with a visual indicator confirming the reset. Any outgoing events for this step are deleted and displayed with red borders.	
Add a parallel branch	Insert an **And** step from the Logic section of the toolbox. Branches execute in order from top to bottom. If an **And** branch exists with no steps, you can't add another **And** branch.	

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Editing scenario	Steps	UI depiction
Insert If/Else logic	Insert an If/Else step from the Logic section of the toolbox. When inserted before an existing step, the downstream flow is preserved in the Else branch. Define conditions using formula building or a script. Branches evaluate top to bottom and the first true branch executes. **Note:** An If/Else step includes one If branch and one Else branch. These branches are required and can't be deleted. Any additional If branches are optional and can be removed.	
Delete If branch	Select the If branch and select the delete icon . You can delete an If branch only when more than one If branch exists. You cannot delete the last remaining If branch or the Else branch.	
Swap outgoing events	Select the event node on the diagram and choose which outgoing event continues the flow. **Note:** Each outgoing event can be used only once per step. Events already in use are read-only. Remove an existing branch to change event assignments	

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Editing scenario	Steps	UI depiction
Delete an event	Select the event on the diagram and select the delete icon . Any associated connections and downstream steps are removed.	

6. Select **Save**.

Result
The updated UI interaction is saved. Any page events or declarative actions that reference this interaction automatically use the updated flow the next time the interaction is triggered.

Duplicate a UI interaction

Create an independent copy of an existing UI interaction to use as a starting point for a new interaction or to safely experiment with changes.

Before you begin
Role required: ui_builder_admin

About this task
Duplicating a UI interaction creates an independent copy of the original, including all steps, inputs, and configuration. The duplicate is not linked to the original. Changes to the duplicate do not affect the original interaction or any page events or declarative actions that reference it.

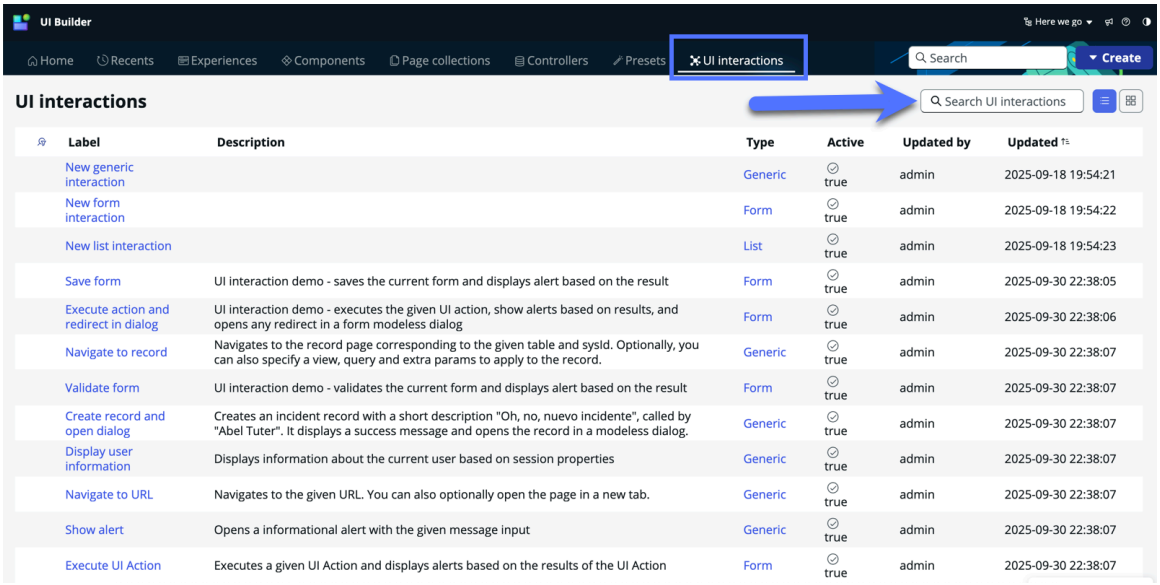
If the UI interaction includes custom UI built with Component Builder, such as a custom modal or modeless dialog, you must also duplicate the custom component separately. The duplicated interaction continues to reference the original component. Update it to reference the new copy.

Note: Duplicating a UI interaction does not duplicate any associated declarative actions. If the original interaction is triggered by a declarative action, create or update a declarative action separately to reference the duplicated interaction.

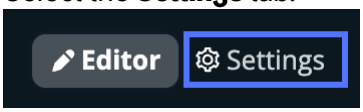
Procedure
1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Select the **UI interactions** tab.
3. In the search field, enter the name of the interaction you want to edit.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

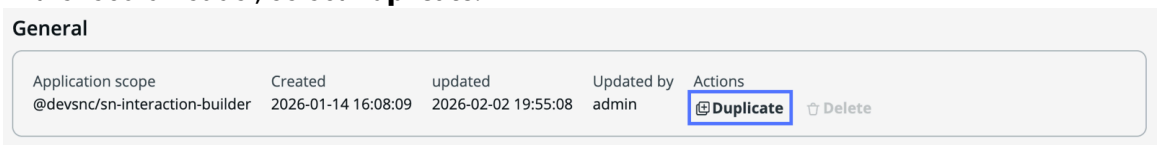
UI interactions list



4. Locate the UI interaction record that you want to duplicate.
5. Open the record.
6. Select the **Settings** tab.



7. In the record header, select **Duplicate**.



8. Update the Name and Description fields, as needed.
9. Select **Create**.

Result

A new UI interaction record is created in your current application scope, ready for editing.

What to do next

To modify the duplicated interaction, see Edit an existing UI interaction.

To use your duplicated interaction, trigger it using one of these methods:

- From declarative action: Corresponding declarative actions are inactive by default and must be enabled to test.
- From UI Builder page event. See Trigger a UI interaction from a page event.

Trigger a UI interaction from a page event

Attach a UI interaction to a UI Builder page event so the interaction runs when a user performs an action on a component.

Before you begin

You have already created a UI interaction that you want to trigger.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Role required: ui_builder_admin

About this task

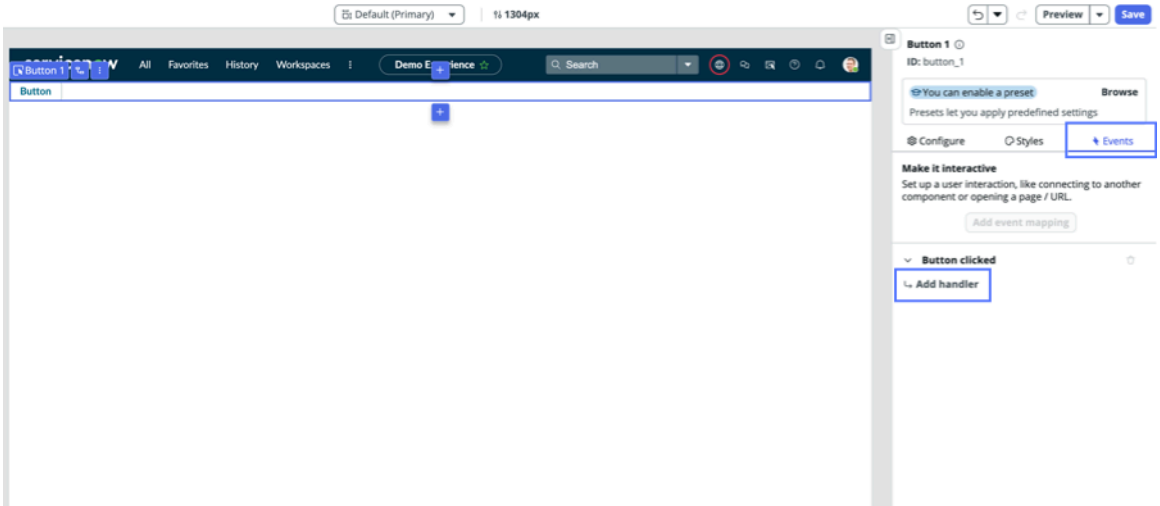
UI interactions don't run on their own. To execute an interaction, you must attach it to an event that occurs on a page or component. UI interactions are triggered by page events, including events mapped from component actions.

In this task, you configure a component event on a page to trigger an existing UI interaction.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   For more information, see Configure how users interact with your applications in UI Builder.
3. Create or open a page within the experience.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. Select the component that should trigger the interaction.
   If the page doesn't already include the component you want to use, add it to the page first. For more information about adding components to a page, see Add and configure components.
5. In the component configuration panel, select the **Events** tab.

Events tab

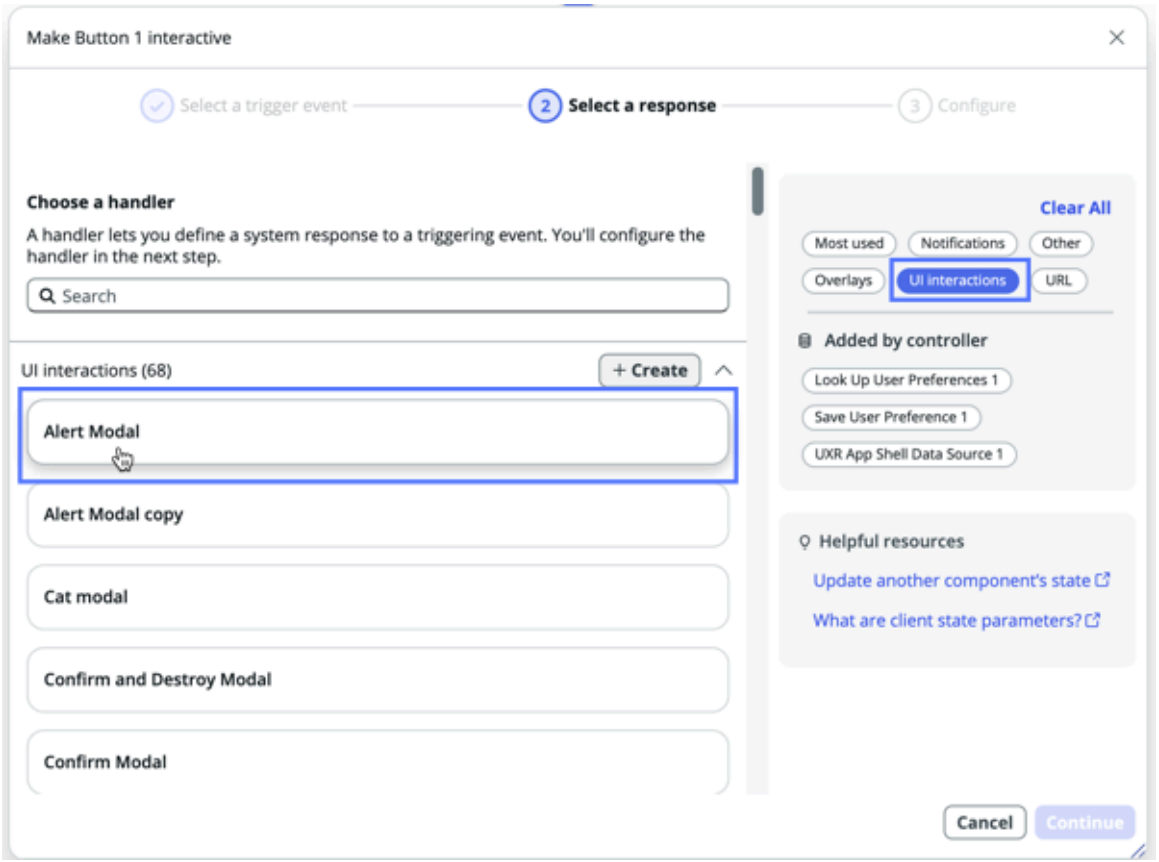


6. Select **Add handler**.
7. Search for your UI interaction by name, or select the **UI interactions** pill to filter results.
8. Select the UI interaction that you want to trigger.

> **Note:** If your interaction type is Form or List and the applicable controller isn't already on the page, it is added automatically. If multiple controllers exist on the page, you're prompted to choose which one feeds data to the interaction.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add handler modal



9. Select **Add**.
10. Select **Save**.
11. Select **Preview**, then perform the action that fires the configured event to verify the interaction runs.

Result

When the specified page event occurs, the UI interaction runs and executes its configured actions.

Delete a UI interaction

Delete a UI interaction when it's no longer needed or when you must replace it with a new version. Although UI interactions can be edited after creation, some foundational properties are intentionally locked to avoid data and runtime errors. In those cases, deletion might be required.

Before you begin

Before deleting a UI interaction, verify whether it's currently in use. Deleting an interaction may break event-driven behavior such as button clicks or modal launches that reference it. The Usage section on the interaction's Settings page shows all pages and declarative actions that reference the interaction.

To avoid data mismatches and runtime errors, some properties of a UI interaction can't be changed after creation or once the interaction is in use:

- UI interaction type
- Input IDs and input types once the interaction is referenced by an active declarative action or page event

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

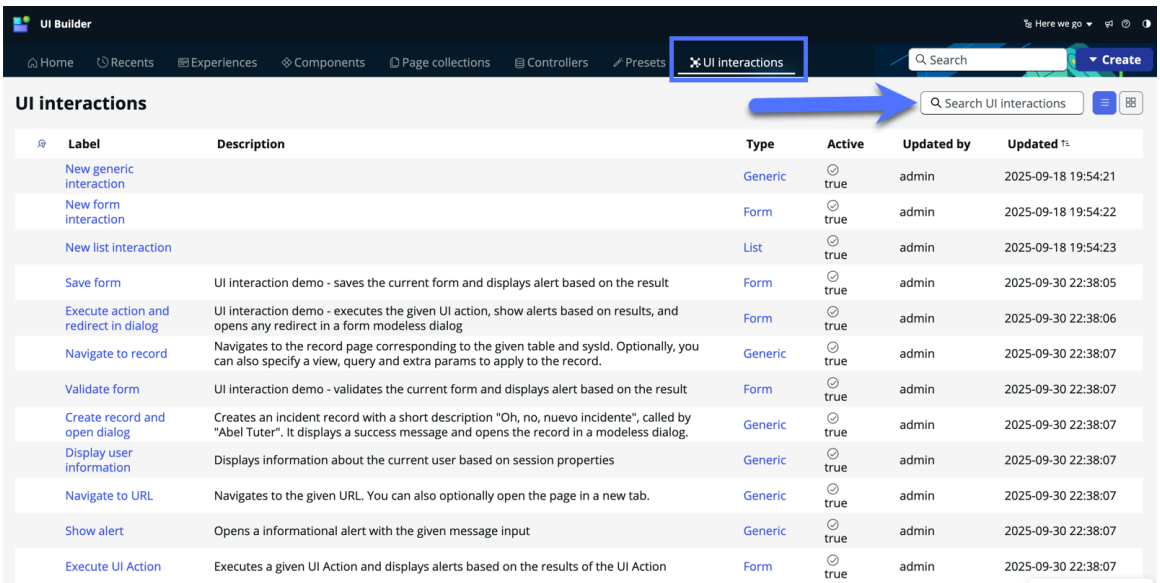
Role required: ui_builder_admin

> **Important:** Deletion is permanent and can't be undone. An interaction can be deleted only if it isn't referenced by any declarative actions or page events.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Select the **UI interactions** tab.
3. In the search field, enter the name of the interaction you want to delete.

UI interactions list



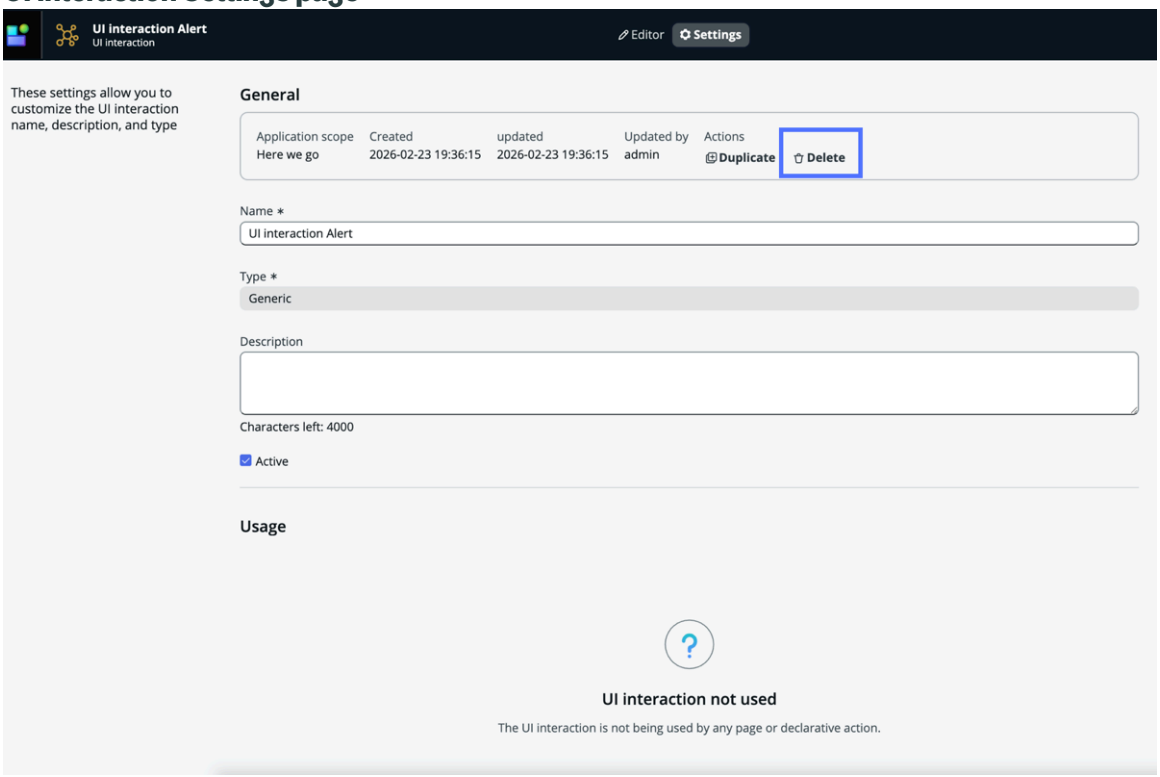
Label	Description	Type	Active	Updated by	Updated
New generic interaction		Generic	true	admin	2025-09-18 19:54:21
New form interaction		Form	true	admin	2025-09-18 19:54:22
New list interaction		List	true	admin	2025-09-18 19:54:23
Save form	UI interaction demo - saves the current form and displays alert based on the result	Form	true	admin	2025-09-30 22:38:05
Execute action and redirect in dialog	UI interaction demo - executes the given UI action, show alerts based on results, and opens any redirect in a form modeless dialog	Form	true	admin	2025-09-30 22:38:06
Navigate to record	Navigates to the record page corresponding to the given table and sysId. Optionally, you can also specify a view, query and extra params to apply to the record.	Generic	true	admin	2025-09-30 22:38:07
Validate form	UI interaction demo - validates the current form and displays alert based on the result	Form	true	admin	2025-09-30 22:38:07
Create record and open dialog	Creates an incident record with a short description "Oh, no, nuevo incidente", called by "Abel Tuter". It displays a success message and opens the record in a modeless dialog.	Generic	true	admin	2025-09-30 22:38:07
Display user information	Displays information about the current user based on session properties	Generic	true	admin	2025-09-30 22:38:07
Navigate to URL	Navigates to the given URL. You can also optionally open the page in a new tab.	Generic	true	admin	2025-09-30 22:38:07
Show alert	Opens a informational alert with the given message input	Generic	true	admin	2025-09-30 22:38:07
Execute UI Action	Executes a given UI Action and displays alerts based on the results of the UI Action	Form	true	admin	2025-09-30 22:38:07

4. Select the interaction from the search results.
5. Verify that the interaction is not in use.
   a. Open the Settings tab.
   b. Review the Usage section.
      If any declarative actions or page events are listed, the interaction can't be deleted.
   c. Remove or update those references before proceeding.
   d. After all references are removed, return to the Settings page and confirm that the Usage section is empty before proceeding with deletion.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

6. From the Settings tab, select **Delete**.

UI interaction Settings page



Result

The UI interaction is permanently removed. Any component or page that previously referenced this interaction will no longer trigger its actions. Update or remove related event handlers as needed.

UI interaction toolbox steps

Find details about UI interaction toolbox steps, organized by interaction type, with step descriptions and any available outgoing events. Steps are added to a UI interaction using the toolbox in the UI interaction editor. The available steps depend on the type you selected when you created the UI interaction, whether Generic, Form, or List.

Available for all interaction types

The following steps are available regardless of whether your UI interaction type is Generic, Form, or List.

Steps that show **None** in the Events column connect directly to the End node or the next step. They execute and do not return a confirmation that they have finished. To continue building the interaction after one of these steps, use an **And** branch.

Steps that show **Success and/or Error** events allow you to define what happens next depending on the outcome, giving you full control over the interaction flow.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

UI interaction toolbox steps for Generic, Form, or List

Step	Category	Description	Events
Add alert notification	Alert	Adds an alert notification banner to the UI.	Alert action selected
If/Else	Logic	Evaluates conditions to branch the interaction flow. Branches evaluate top to bottom; the first true branch executes.	None
And	Logic	Executes multiple branches in parallel (executes in order top to bottom). Use when activities should happen at the same time.	None
Server script	Script	Executes a server-side script. Requires at least one user role.	Success, Error
Close workspace tab	Navigation	Closes the current Workspace tab. Can optionally prompt the user before closing.	None
Navigate to record	Navigation	Navigates the user to a specific record page.	None
Navigate to route	Navigation	Navigates the user to a custom (non-record) route.	None
Navigate to URL	Navigation	Navigates the user to a URL, optionally in a new browser tab.	None
Alert	Modal	Displays a modal that notifies users without requiring further input.	Modal closed
Confirm	Modal	Displays a modal prompting the user to confirm or cancel an action.	Modal closed, Primary button clicked, Secondary button clicked
Confirm and destroy	Modal	Displays a modal for irreversible actions (such as deletion), with directive language and warnings.	Modal closed, Primary button clicked, Secondary button clicked

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

UI interaction toolbox steps for Generic, Form, or List (continued)

Step	Category	Description	Events
User input	Modal	Displays a modal with fields for the user to complete. Fields can be added manually or sourced from a table.	Modal closed, Primary button clicked, Secondary button clicked
Create modal	Modal	Opens UI Builder component building to create a custom modal.	Modal closed; additional events can be defined by the custom component
Form	Modeless dialog	Displays a form inside a modeless dialog, allowing users to multi-task within their current context.	Modeless dialog closed, Primary button clicked, Secondary button clicked
Create modeless dialog		Opens UI Builder component building to create a custom modeless dialog.	Modeless dialog closed; additional events can be defined by the custom component

Form interaction type only

The following steps are only available when the UI interaction type is set to Form. Form interactions require a form controller on the page and provide access to form-specific data and actions.

Field actions modify individual fields on the form. They do not produce success or error events; the step connects directly to the next step or end.

UI interaction toolbox steps for Field actions

Step	Description	Events
Add field message	Adds a message below a specified field.	None
Clear field message(s)	Clears messages from a specified field.	None
Clear field messages by type	Clears all field messages of a given type. For example, all info messages.	None
Clear field value	Clears the value of a specified field.	None
Set field invalid	Sets a field to valid or invalid.	None
Set field label	Updates the label of a specified field.	None
Set field placeholder value	Sets the placeholder value of a specified field.	None

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

UI interaction toolbox steps for Field actions (continued)

Step	Description	Events
Set field read-only	Sets a specified field to read-only.	None
Set field required	Sets a field as required or optional.	None
Set field value	Sets the value of a field. Can trigger or bypass dirty state.	None
Set field visibility	Shows or hides a specified field.	None

Form actions operate on the form as a whole and include capabilities such as saving the form, validating the form, or displaying alerts.

UI interaction toolbox steps for Form actions

Step	Description	Events
Add form alert	Displays an alert at the top of the form component.	None
Clear all form alerts	Removes all alert notifications from the form.	None
Clear form alerts by type	Removes form alert notifications of a specified type.	None
Execute client script	Runs a client script on the form.	Success, Error
Execute UI action	Executes a UI action, client and server side, that has been enabled for Configurable Workspace.	Success, Error
Refresh form	Refreshes the form component.	Success
Save	Saves the form.	Success
Show/hide annotations	Shows or hides annotations on the form.	None
Submit form using UI action	Submits the form using a specified UI action.	Success, Error
Validate	Validates the form.	Success, Error

List interaction type only

The following steps are only available when the UI interaction type is set to List. List interactions require a list controller on the page and provide access to list-specific data and actions.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

UI interaction toolbox steps for List actions

Step	Description	Events
Execute client script	Runs a client script on the list.	Success, Error
Group by column	Groups the list by a specified column.	None
Refresh list	Refreshes the list component.	Success
Set query	Updates the encoded query of the list.	None
Sort	Sorts the list by a specified column, ascending or descending.	None

Working with UI interaction demo data

Demo UI interactions give you simple, ready-made examples that demonstrate common UI Interaction patterns. You can duplicate these samples to explore how interactions work, experiment safely, and use them as starting points when building your own experiences.

Key considerations for working with demo interactions

- Duplicate before customizing. Demo interactions are reference examples; create a copy before modifying.
- Backward compatibility isn't guaranteed. Demo content may change between releases.
- Custom components must be duplicated separately.
- Corresponding declarative actions are inactive by default and must be manually set to true before testing.
- Demo interactions don't cover long-running processes, usage tracking, automatic refresh behavior, multi-step journeys, or complex branching logic.

Available demo data

Demo Interaction	What it does	Why It's Useful	Use case
DEMO Cascade delete	Prompts you with a confirmation message before deleting records, including information about related child records that will also be deleted.	Helps you understand how to design safe delete flows that clearly communicate downstream impacts before continuing.	Safe record deletion

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Available demo data (continued)

Demo Interaction	What it does	Why It's Useful	Use case
	Note: Unlike other demo interactions, Cascade delete must be duplicated into Global scope before it executes correctly.		
DEMO Close incident	Simplifies closing an incident by updating the record and showing a confirmation.	Shows how to build quick, common life-cycle update actions for record forms such as closing, resolving, and advancing status.	Record life-cycle transition
DEMO Report knowledge gap	Opens a short form enabling you to report missing or incomplete knowledge content.	Demonstrates how to gather user input in a lightweight modal and submit it as a record.	User#submitted feedback
DEMO Create new record	Opens a modeless dialog with a form for creating a new record without leaving the current page.	Teaches you how to invoke record creation in-context using UI Interactions.	In#context record creation
DEMO Reassign records	Opens a modal that lets you select an Assignment Group and Assigned To value, then updates the selected records.	Useful for showing guided update flows, including dependent fields and multi-record reassignment.	Bulk reassignment
DEMO Open modal to display table, sys_id, form view	Opens a modal that dynamically displays the table, sys_id, and form view of the record the user is viewing. This data is pulled from the form controller, not passed in as inputs.	Illustrates how to surface additional record details in a modal instead of navigating away.	Contextual record inspection
DEMO Save form and close tab	Saves the current Workspace form and closes the tab (or shows an alert if closing isn't possible).	Demonstrates Workspace#aware actions, helping you automate save and finish patterns commonly needed by agents.	Save#and#finish workflow

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Viewing demo data in UI Builder

1. From the UI Builder home page, select the **UI interactions** tab to view the list of available interactions.
2. Use either the list filter from the Label column to search for interactions that begin with DEMO or enter `DEMO` in the search field.
3. Select any DEMO interaction to view how it works or duplicate it for your own use.

Duplicate a demo UI interaction

Create a duplicate of a demo UI interaction example to work with a separate, editable copy.

Before you begin

Role required: ui_builder_admin

About this task

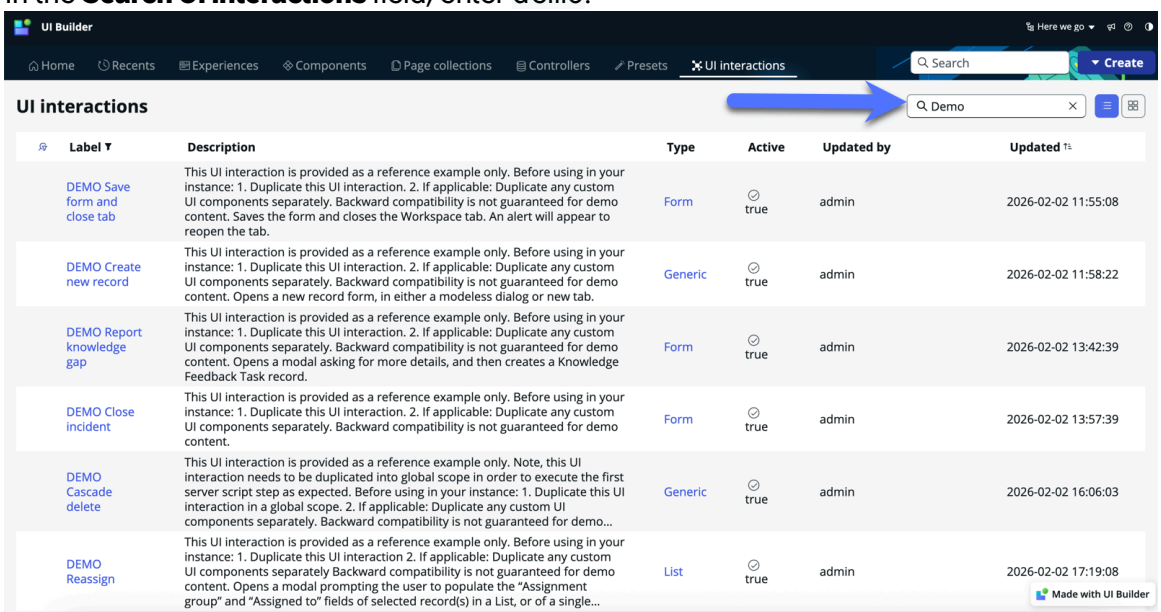
Demo UI interaction records define sample UI logic and scripts used to demonstrate common interaction patterns.

When you duplicate demo data, you create a separate copy that you can run or customize without impacting the original record. The duplicated interaction is independent of the original demo and can be safely modified. Demo interactions appear in the UI interactions list like any other interaction, allowing you to open, review, or duplicate them for use in your own scope.

Note: The Cascade Delete demo interaction example must be duplicated in the global scope for its first server script step to run correctly. This requirement is unique to Cascade Delete; all other demo interaction examples function normally when duplicated in your application scope.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Select the **UI interactions** tab.
3. In the **Search UI interactions** field, enter `demo`.

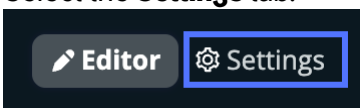


The list filters to show interactions containing **demo** in their labels or descriptions.

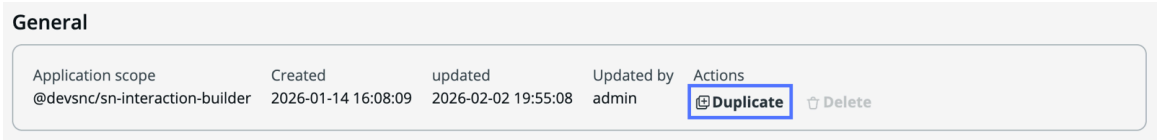
4. Locate the demo data record that you want to duplicate.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

5. Open the record.
6. Select the **Settings** tab.



7. In the record header, select **Duplicate**.



8. Update the Name and Description fields, as needed.
9. Select **Create**.

Result
A new UI interaction record is created in your current application scope, ready for editing.

What to do next
To use your duplicated interaction, trigger it using one of these methods:

- From declarative action: Corresponding declarative actions are inactive by default and must be enabled to test.
- From UI Builder page event. See Trigger a UI interaction from a page event.

Manage the visual style of UI Builder experiences

Themes enable you to change the visual style of your UI Builder experiences so that they express the look and feel of your brand

Getting started with themes

By default, the Polaris theme is applied to UI Builder experiences. For more information, see Working with themes in Next Experience.

Create a new theme with Theme Builder

Theme Builder enables you to create, edit, manage, and apply Next Experience themes using a friendly visual interface. For more information, see Configuring Next Experience with Theme Builder.

Create a custom theme

If you are an admin user comfortable working in style records and JSON code, you can create a custom theme to override the default Polaris style. This option gives you the most control over typefaces, colors, and images (including the banner and logo) in a theme. For more information, see Configuring Next Experience themes and preferences.

Working with dark theme

If you are interested in dark theme, see Working with the dark theme.

Create custom style classes and rules

Use style classes rules in UI Builder to apply custom visual styling to UI elements by referencing CSS classes, see Create custom style classes and rules.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

View experience theme

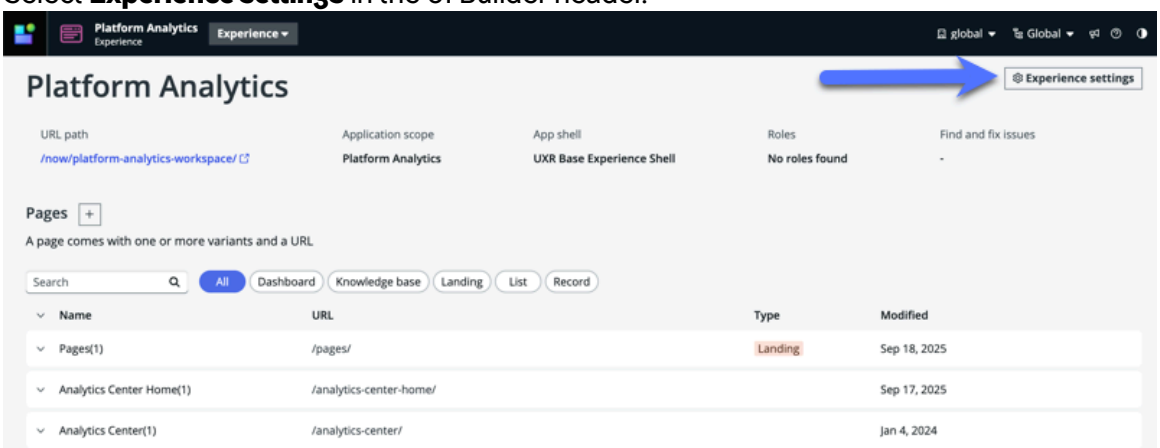
View the theme applied to your experience. The theme sets the visual style of the experience and provides a consistent look and feel across all pages.

Before you begin

Role required: ui_builder_admin

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Select **Experience settings** in the UI Builder header.



4. Scroll to the **Branding and theming** section and note what is listed in **Themes**. By default, the Polaris theme is applied to UI Builder experiences.

What to do next

If you want to customize the theme, there are two options:

- **Theme Builder** enables you to create, edit, manage, and apply Next Experience themes using a friendly visual interface. For more information, see Configuring Next Experience with Theme Builder.
- If you are an admin user comfortable working in style records and JSON code, you can create a custom theme to override the default Polaris style. This option gives you the most control over typefaces, colors, and images (including the banner and logo) in a theme. For more information, see Configuring Next Experience themes and preferences.

Create custom style classes and rules

Use style classes rules in UI Builder to apply custom visual styling to UI elements by referencing CSS classes.

Before you begin

Role required: admin

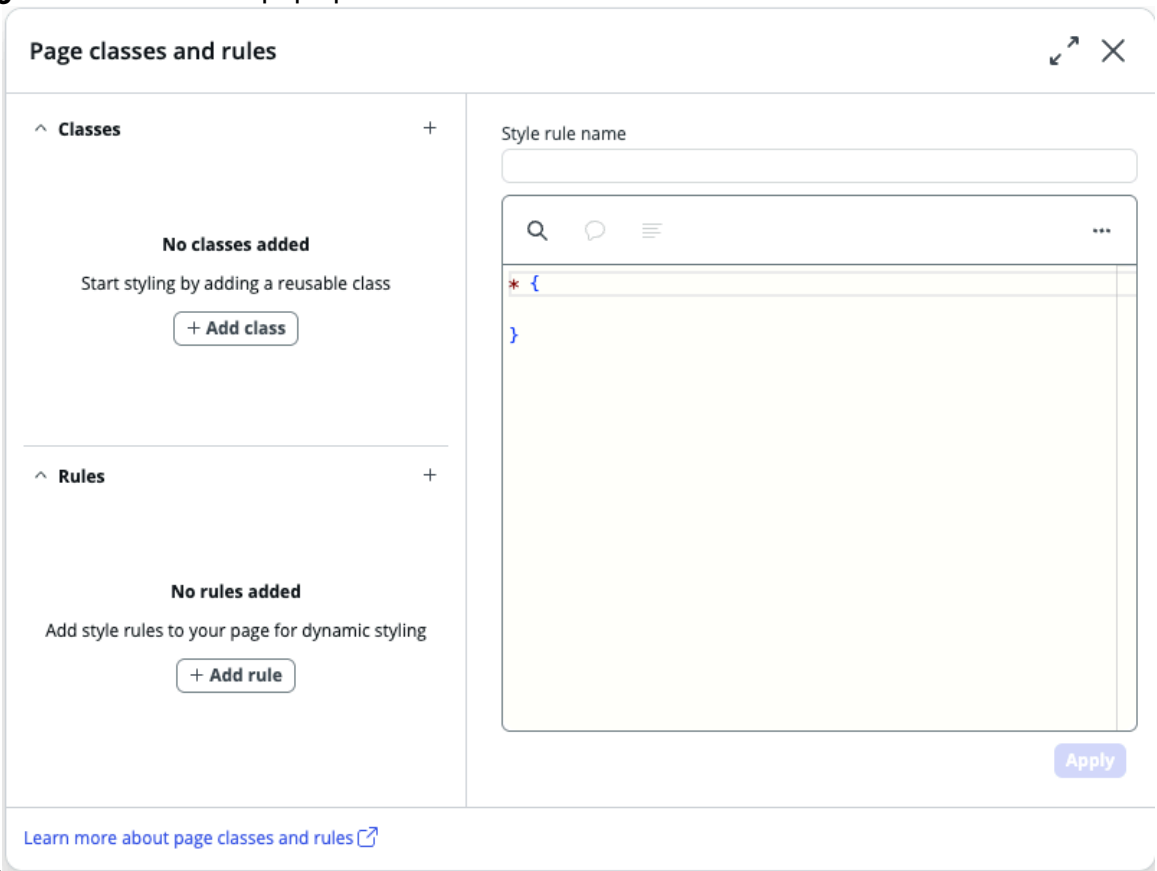
About this task

Control the appearance of fields, containers, buttons, and other interface components without writing inline styles making the UI cleaner, more consistent, and easier to maintain. All style classes and rules can be applied to any container or component on the page. For more information about CSS styles, see the ServiceNow Developer site and the ServiceNow Community site.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create or open a page.
4. Add a component to your page.
5. Select the **Styles** tab in the configuration panel.
6. Select the **...** icon next to the **Style class** field.
   The **Page classes and rules** pop up will appear.



7. **Optional:** Select **+ Add class**.
   a. Enter a name in the **Style class name** field.
   b. Provide the CSS style you would like to apply to the component.
   c. Select **Apply**.
8. **Optional:** Select **+ Add rule**.
   a. Enter a name in the **Style rule name** field.
   b. Provide the CSS rule you would like to apply to the component.
   c. Select **Apply**.
9. Close the **Page classes and rules** pop up.
10. Select the **Style class** field.
11. Select a style class or style rule from the list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

12. Select **Save**.
13. Select **Preview** to test the applied style class or style rule.

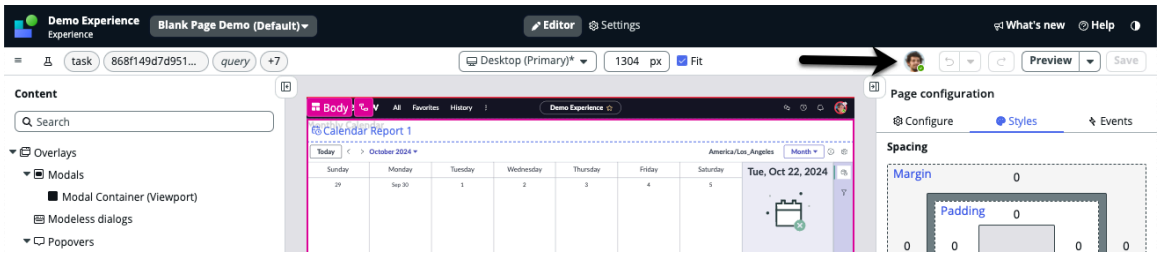
Collaborate with other UI Builder developers

UI Builder provides real-time collaboration tools and user presence indicators for more efficient and intuitive UI development.

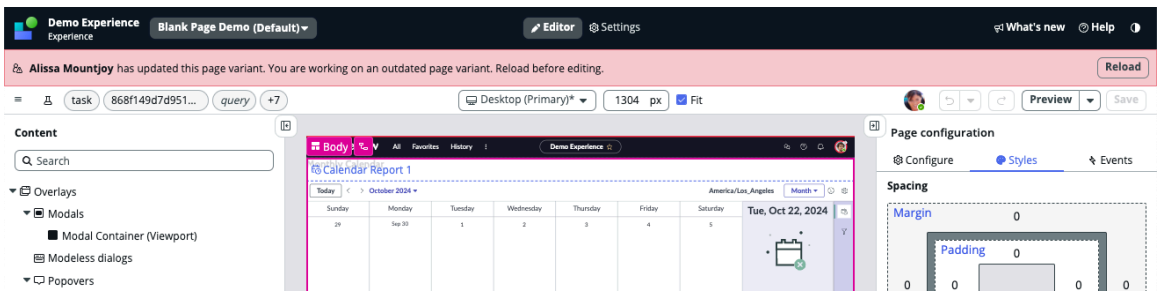
UI Builder lets you see who is online and working on the same page in an experience.

The avatars of other developers appear in the UI Builder header next to the undo button. A dot on the avatar of the user represents online status.

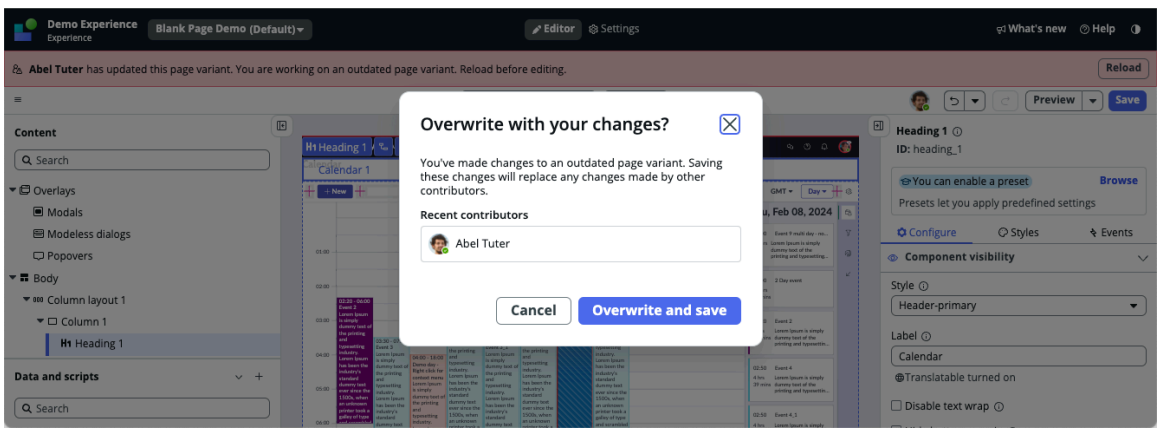
- Green dot if the user is currently working on the page
- No dot if the user is not logged in



A banner will appear asking you to reload the page to view the latest version if changes are made by other users while you are editing a page.



If you have made changes at the same time as another user you will be prompted to **Overwrite and save** your changes to proceed with previewing the page.



Find and compare page changes

Track and compare changes between different versions of a page when collaborating with other page builders.

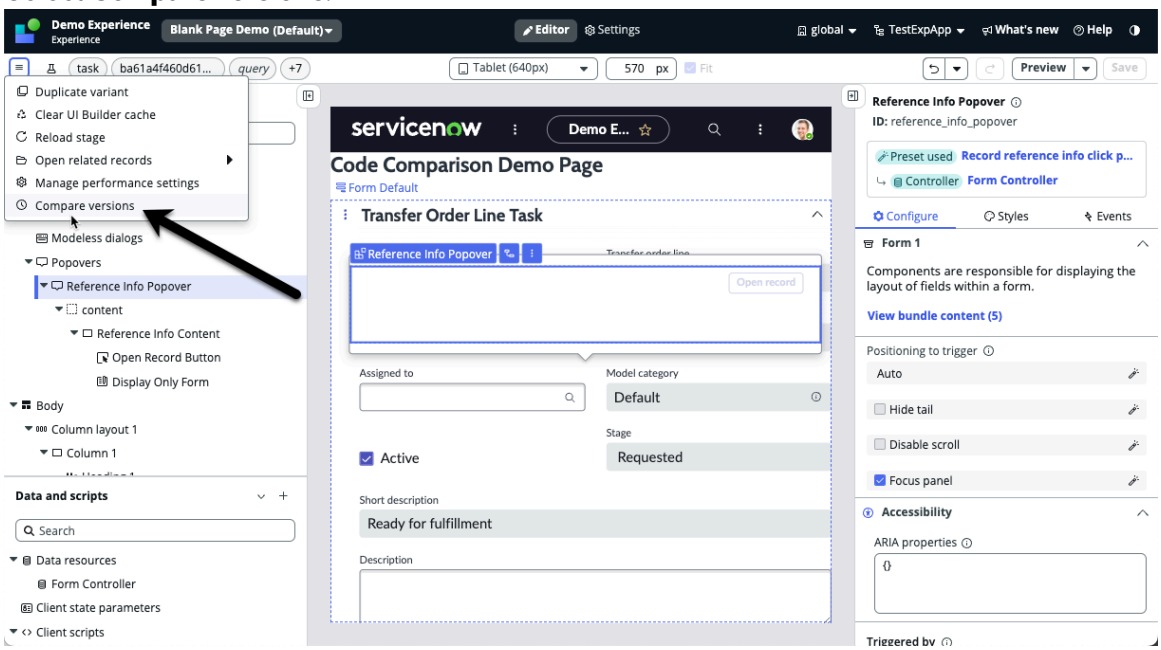
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

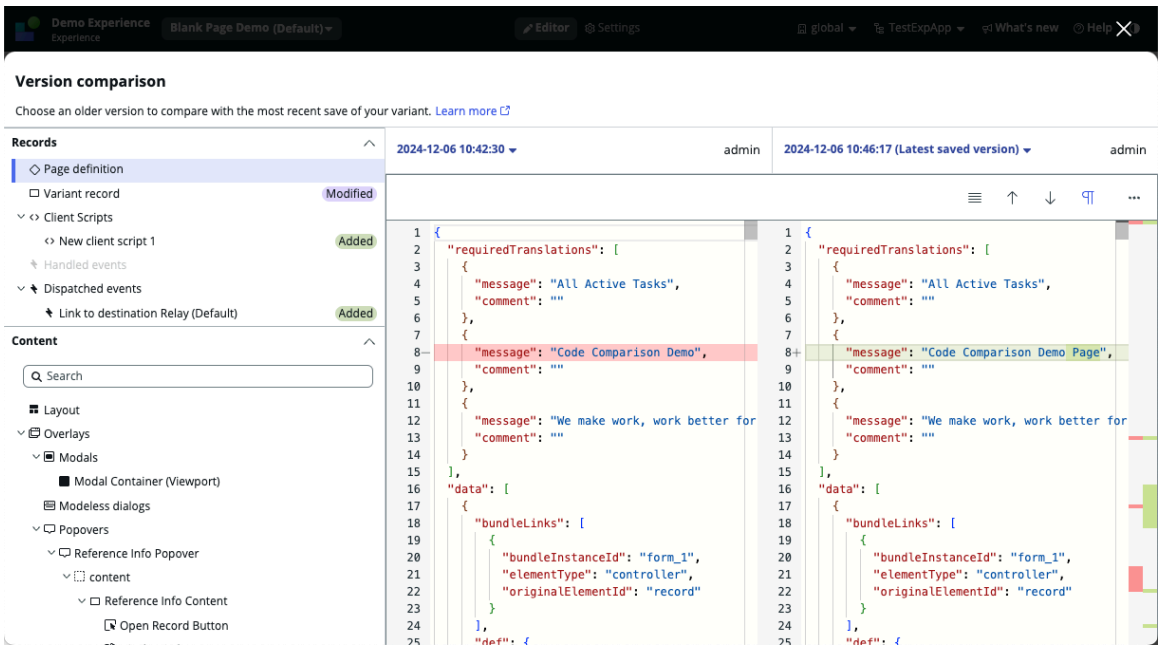
Role required: ui_builder_admin

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open the experience that contains the page where you want to compare changes.
3. Select the page with the changes you want to compare.
4. Select the menu icon.
5. Select **Compare versions**.



The **Version comparison** drawer opens.



6. Review changes by selecting an item that displays one of the following:
   - Added
   - Modified

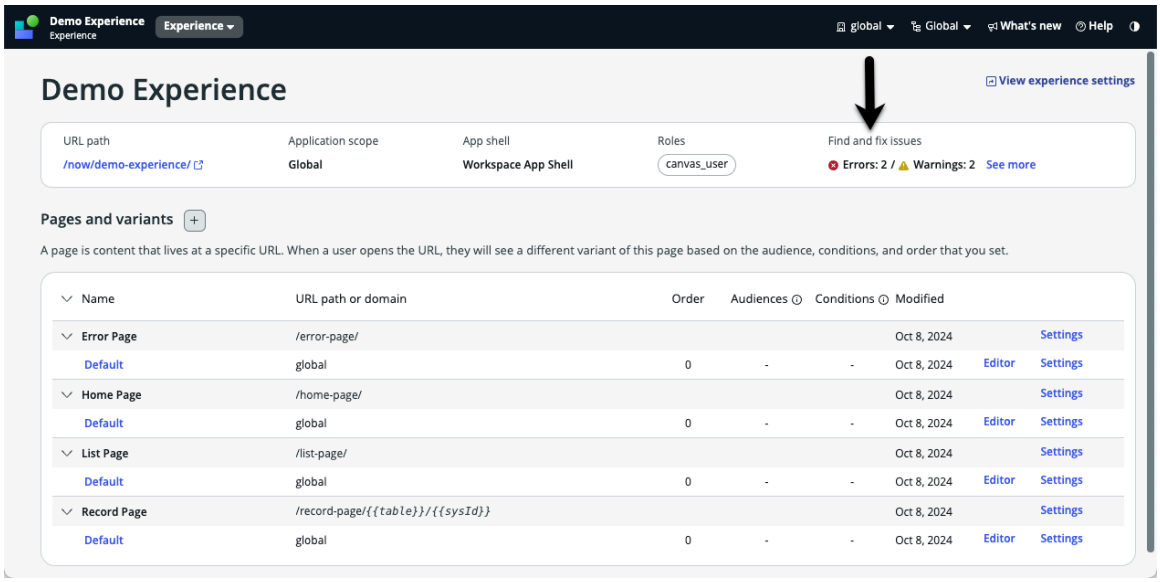
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Moved
- Deleted

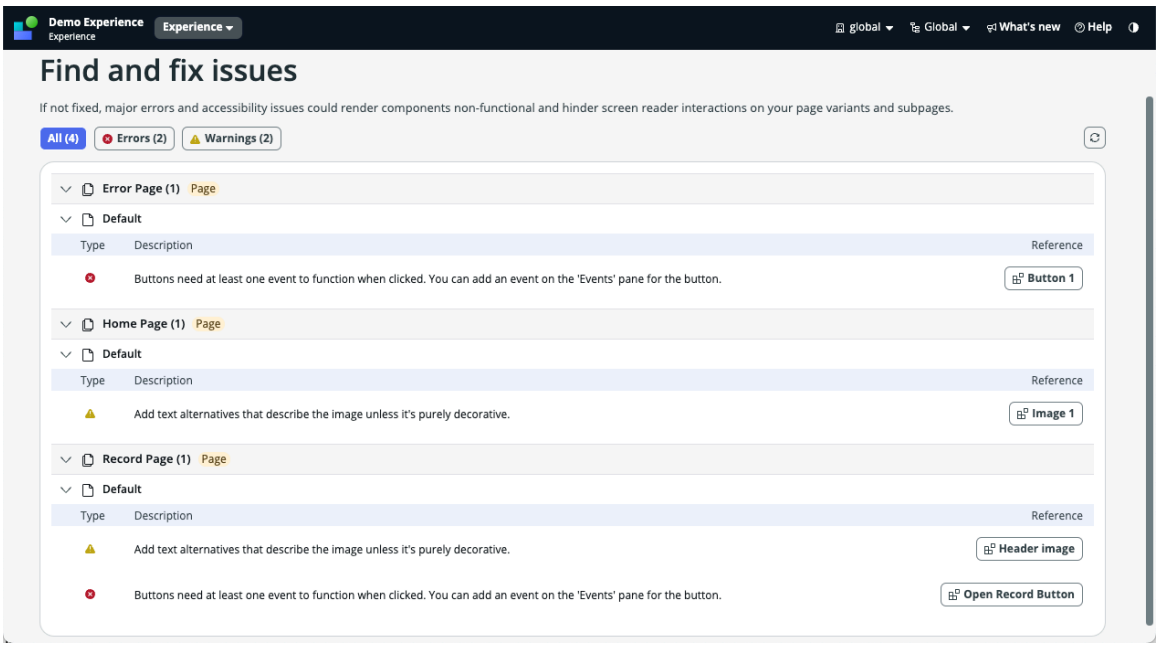
Find and fix issues in UI Builder

UI Builder can help identify common configuration issues and guidance on how to fix them.

UI Builder can help you find and fix issues by checking your experience for missing configurations, errors, and accessibility standards. The list of issues can be found within the experience view of UI Builder under the **Find and fix issues** section.

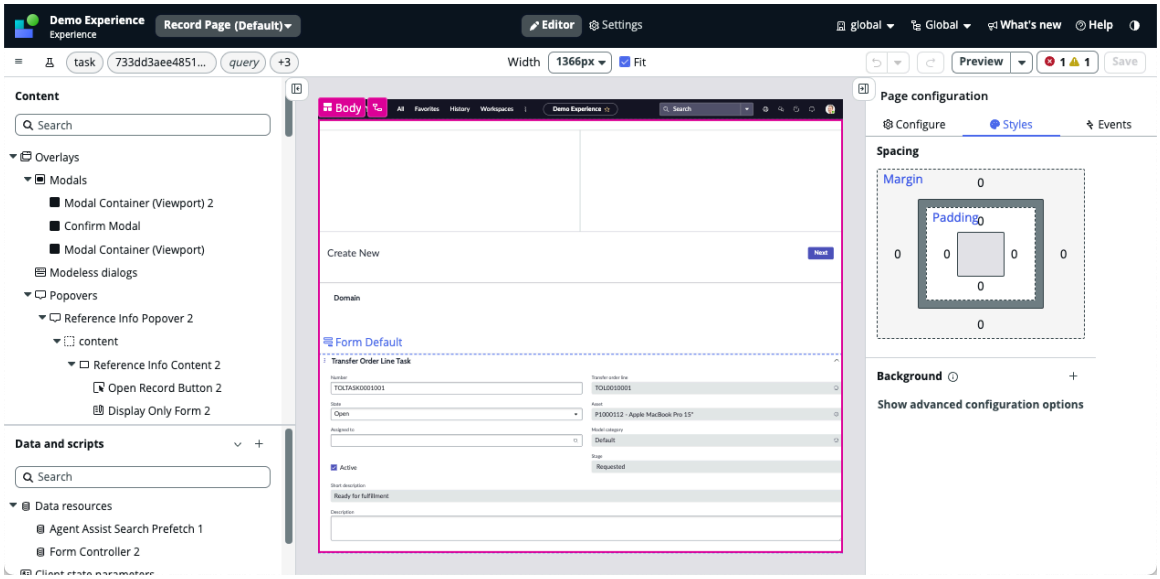


Find and fix issues list

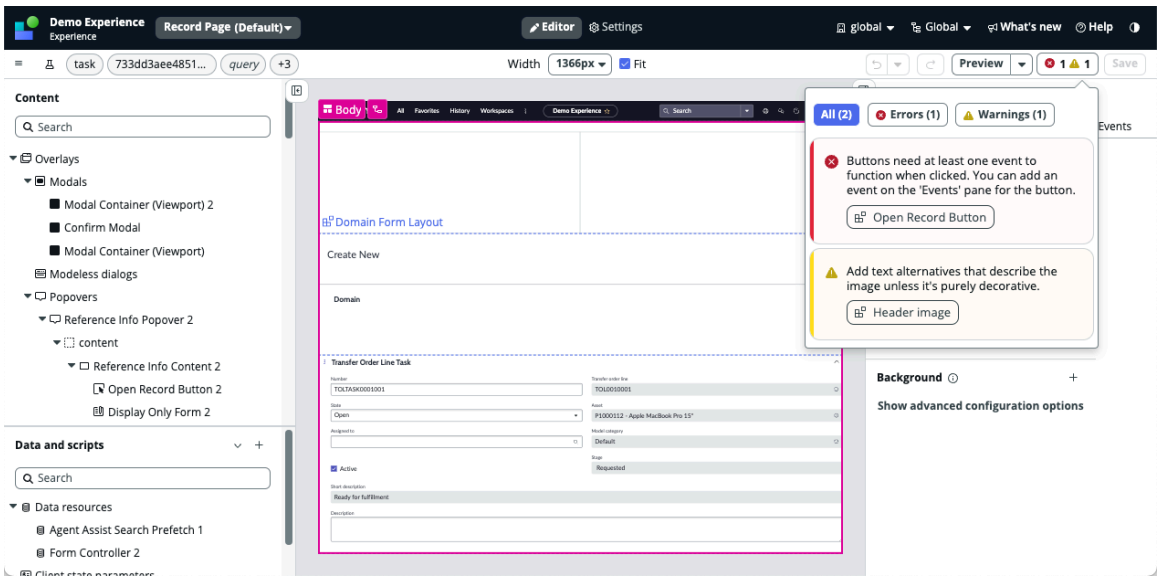


You can also find UI Builder issues in the header of the page builder next to the **Save** button. The button auto updates as it finds errors and recommendations for the content on your page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



To find more information, select the builder icon to view the list of recommendations made by UI Builder. Each error and warning will list the component name that contains the issue.



Find and fix issues in UI Builder

Find and fix issues found in your UI Builder experience.

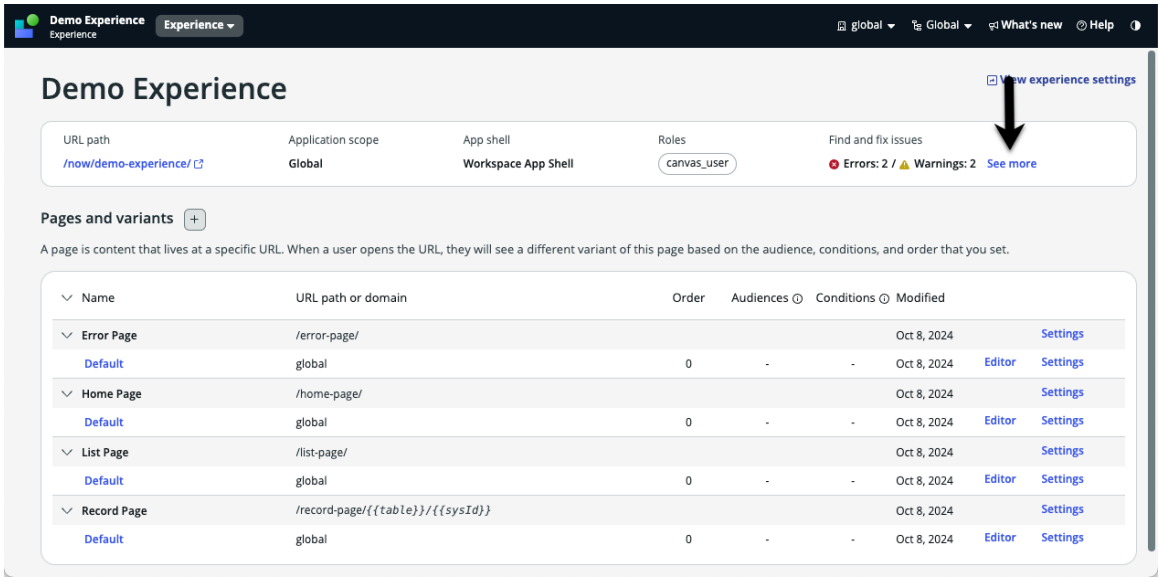
Before you begin

Role required: ui_builder_admin

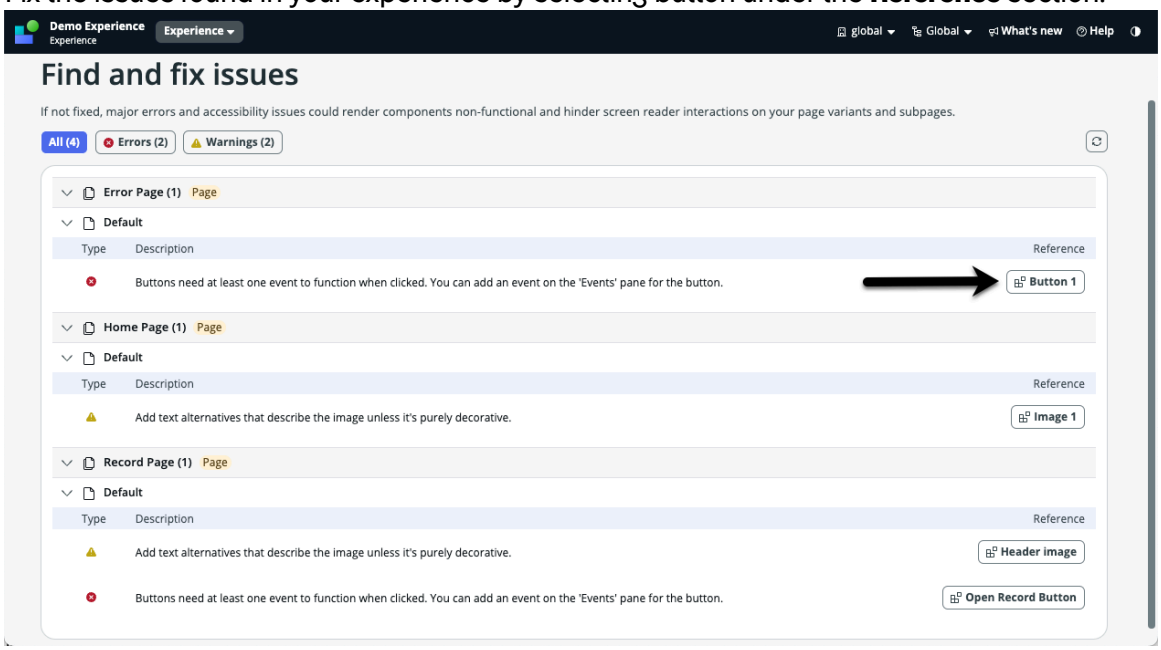
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open the experience you want to check for issues by selecting **Experiences**.
3. Select **See more** under the **Find and fix issues** section.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



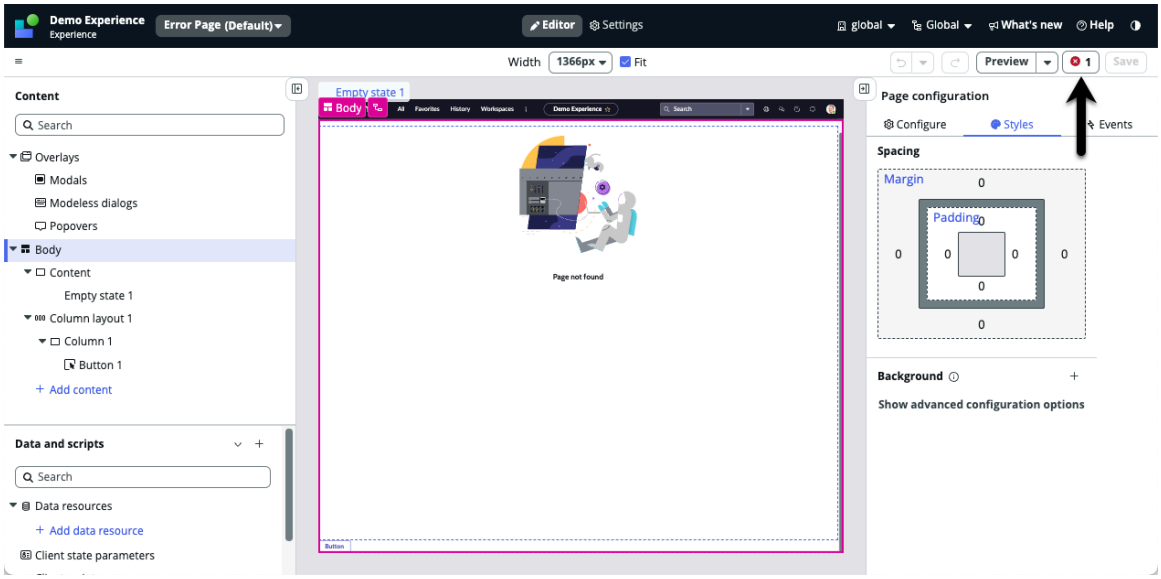
4. Fix the issues found in your experience by selecting button under the **Reference** section.



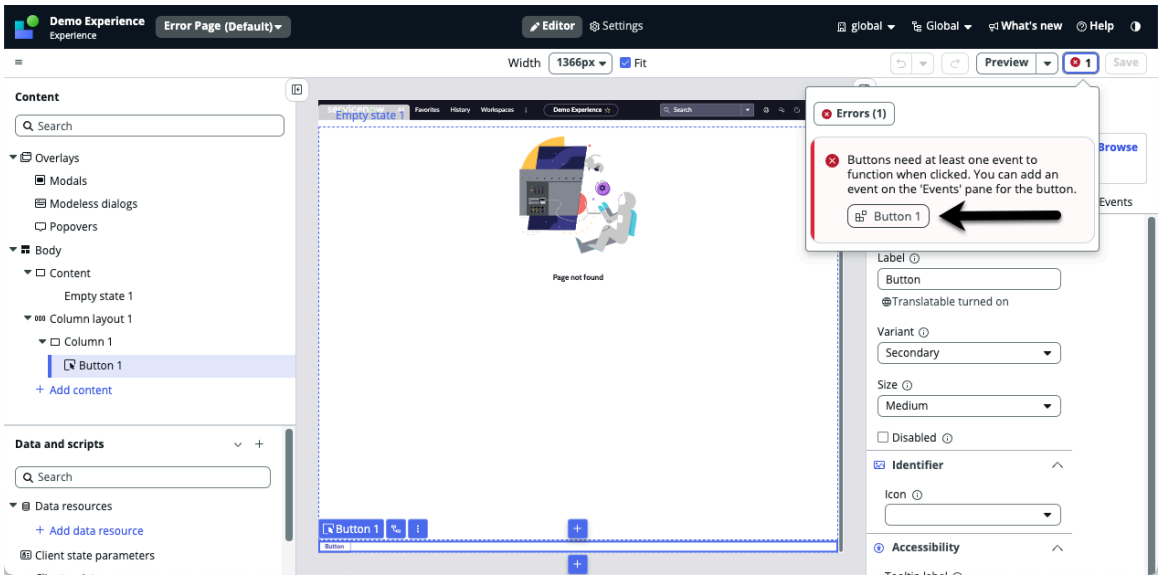
A new tab will open to the page with the issue.

5. Select the error button in the header to get a brief description of the issue and which component contains the issue.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



The component name will be listed in the error or warning.



6. Select the component with the issue and follow the instructions listed in the warning or error text.
7. Select **Save**.

How to use the Now Assist panel in UI Builder

Use the Now Assist panel in UI Builder to request guidance, streamline page building, and better understand the content on your pages.

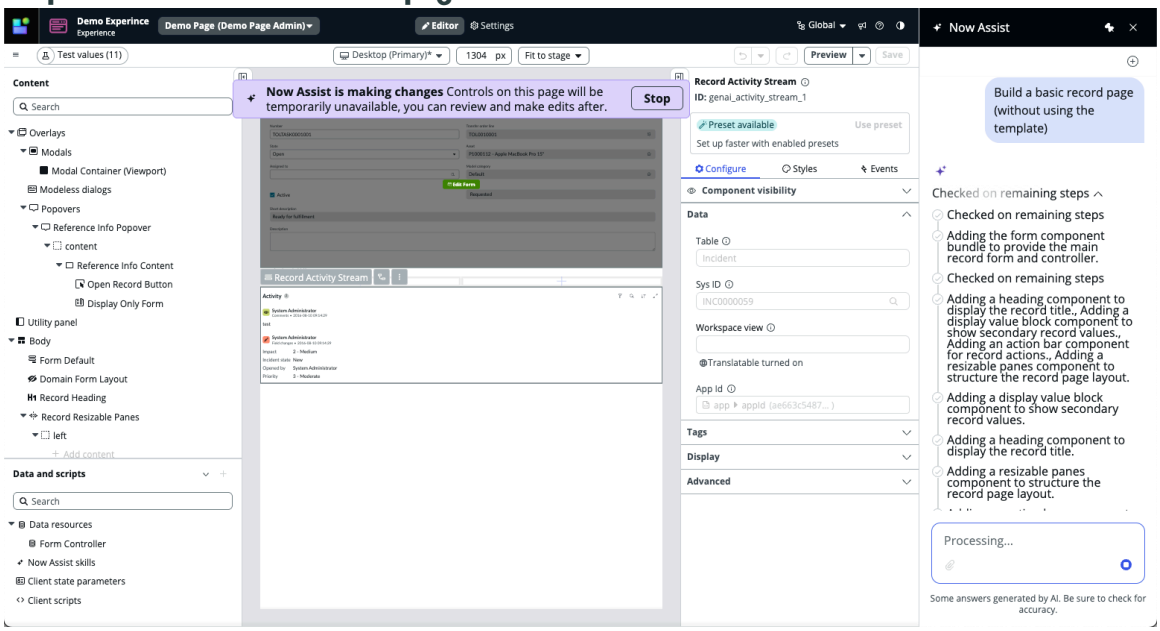
Now Assist for UI Builder overview

Now Assist for UI Builder is a powerful tool that can be used to enhance the way you build experiences in UI Builder. Whether you need help configuring a page, adding components or data, or uncovering insights form a page, Now Assist for UI Builder can help you design and deliver the pages you want.

Now Assist for UI Builder enables you to leverage the generative and agentic AI capabilities of the ServiceNow AI Platform. You can configure the Now Assist panel to be accessible from any page within UI Builder.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Request to build a basic record page



Note: You must install the ServiceNow Otto for Creator plugin in the AI Admin Hub console to use the Now Assist panel. For more information, see Now Assist Admin console.

For more information on the Now Assist panel, see Now Assist panel.

What to ask for Conversational Help

Conversational Help uses generative AI to provide instant support from anywhere in your experience, enabling you to ask questions and get quick answers without disrupting your current activity. Conversational Help answers questions based on product documentation, but cannot provide guidance specific to what you've configured in your instance.

Ask questions in the Now Assist panel to discover new features and learn how to complete specific tasks. For example:

- `Can I configure side navigation?`
- `How do I bind dynamic data to a repeater?`
- `Where can I find best practices and design considerations when using *Next Experience Components* on my page?`
- `How do I create a reusable page template?`

What to ask for Page Building help

Now Assist Page Building uses generative AI to add content such as components, data bindings, or events, enabling you to quickly build pages using a simple text prompt. Page building is based on internal recipes that ensure content added to a page will function properly.

- `Can you create a record page?`
- `Bind the heading to the current user's name.`
- `Create a client state parameter named FOO that is type STRING and has an initial value of BAR`
- `Add a handler so that when I click the Submit button, it saves the record.`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

What to ask for Page Insights

Now Assist Page Insights can extract information from pages, components, and data resources. Use the Now Assist page insights to save time when investigating a page built by a different user or when identifying where data is coming from within a component.

- `List all the components on my page and say if they are using presets`
- `How are the components bound to the data resources?`
- `What components have theme overrides?`
- `Are there any unused client scripts?`

Get help with the Now Assist panel in UI Builder

Ask questions and get quick answers without leaving UI Builder.

Before you begin

- Role required: ui_builder_admin
- The ServiceNow Otto for Creator plugin is activated

About this task

Responses and outputs from Now Assist can vary, even when the same prompt is used multiple times. Always review the generated content carefully to ensure accuracy and completeness.

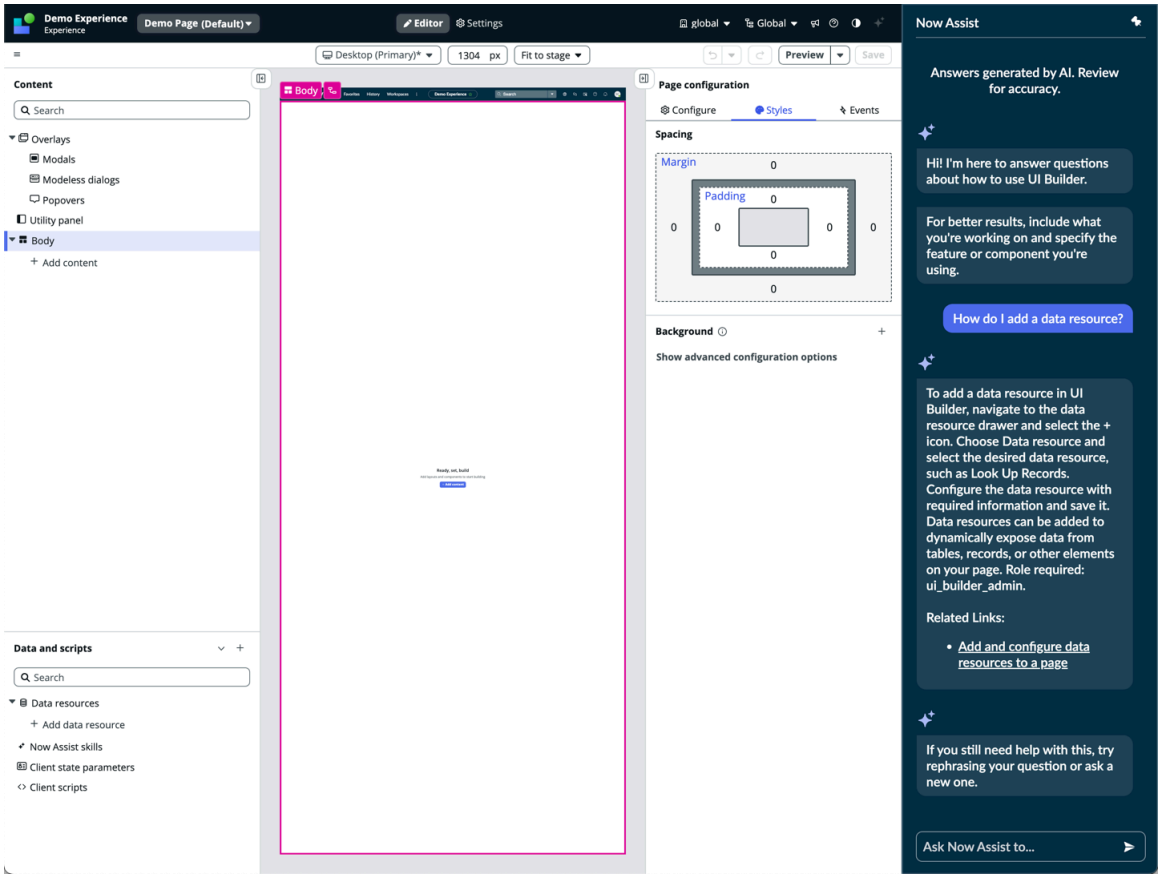
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create or open a page.
4. In the upper right, select the Now Assist icon  to open the Now Assist panel.

   Tip: The Now Assist panel opens in a pinned position by default, keeping it fixed on your screen. Select the pin icon  to make the panel movable. When unpinned, drag the panel freely across the workspace for greater flexibility and visibility.

5. Enter your question or task in the text field.
6. Press the Enter key or select the **Send** button.
   Now Assist uses generative AI to provide immediate answers to your questions.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



7. Optional: Continue the conversation with follow-up questions as needed.

Advanced UI Builder

UI Builder can be used by application developers with a variety of skill levels. We recommend developers with a high level of experience, sometimes referred to as pro-coders, perform the procedures in this section.

Configure components and repeaters (advanced feature)

You can add and configure components as you build pages in UI Builder. You can also use repeaters to repeat components or multiple components with results from a data resource.

Add and configure components

Learn how to add components to your page in UI Builder. A page is built by adding components.

Before you begin

Role required: ui_builder_admin

About this task

Components are the building blocks used to create custom pages in UI Builder. Learn how to add a component to a page. After you add a component to a page, you must configure the component. For more information about configuring components, see Component documentation on the ServiceNow Developer Site.

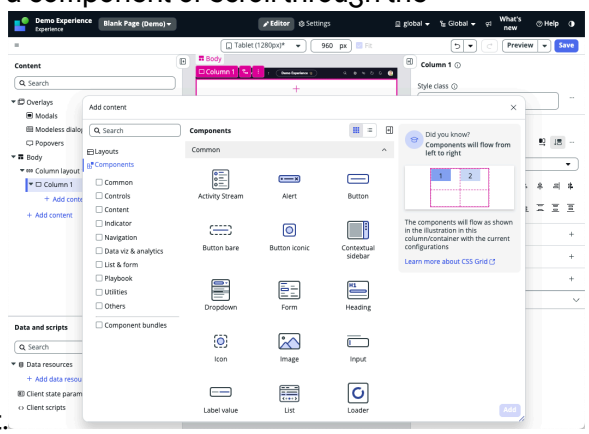
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.

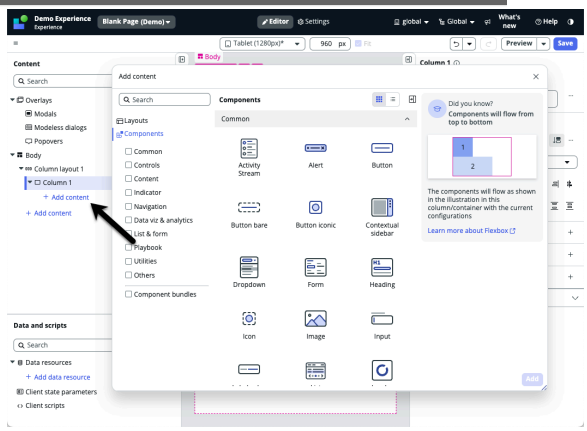
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

See Configure how users interact with your applications in UI Builder for more information on creating experiences.

3. Open or create a page.
   If you open an existing page, make sure you are in the same scope as the original page. If not, change the scope before you start editing the page. Application scoping protects applications by identifying and restricting access to application files and data. Administrators set the scope to specify what parts of an application are accessible to other applications. Application scope protects data and application files. See Learn about security and roles for more information about application scope.
4. Select an existing container or create a column layout.
   See Organize components in UI Builder pages for more information.
5. Add components to your page.
   UI Builder comes with a library of components to choose from. You add components as the building blocks of your page. For example, you can add a heading, data visualizations, and so on. The following table shows you the different ways you can add a component to a page.

To add a component	Do the following
Directly from your page	From your page, select **+** and choose a component from the toolbox. You can search for a component or scroll through the list. 
Floating menu above page	Select the floating menu above your component. ○ Select **Add before** or **Add after**. ○ Choose a component from the toolbox.
Content tree	You can add a component from the content tree in the following two ways. ○ Select **+Add content** beneath a container in the content tree. Then choose a component from the toolbox. ○ Move your mouse device over a container in the content tree, select the **Menu** icon, and select **Add component**. Then choose a component from the toolbox.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

To add a component	Do the following
	

6. Now configure the properties of the components that you just added.
   Configuring components means to customize them to your needs. For more information about configuring components, see Next Experience Components.

   a. Select the component that you want to configure.

   b. Select the **Configure** tab from the configuration panel in UI Builder.

   c. Customize the component properties for the component.
   For example, you could add a name for a button component. Some components, like data visualizations, require a data source before you can configure the properties. Each component has different configuration properties based on the requirements and options for each component. For example, the **Button** component configuration is simple, while a **List** component requires more configuration.

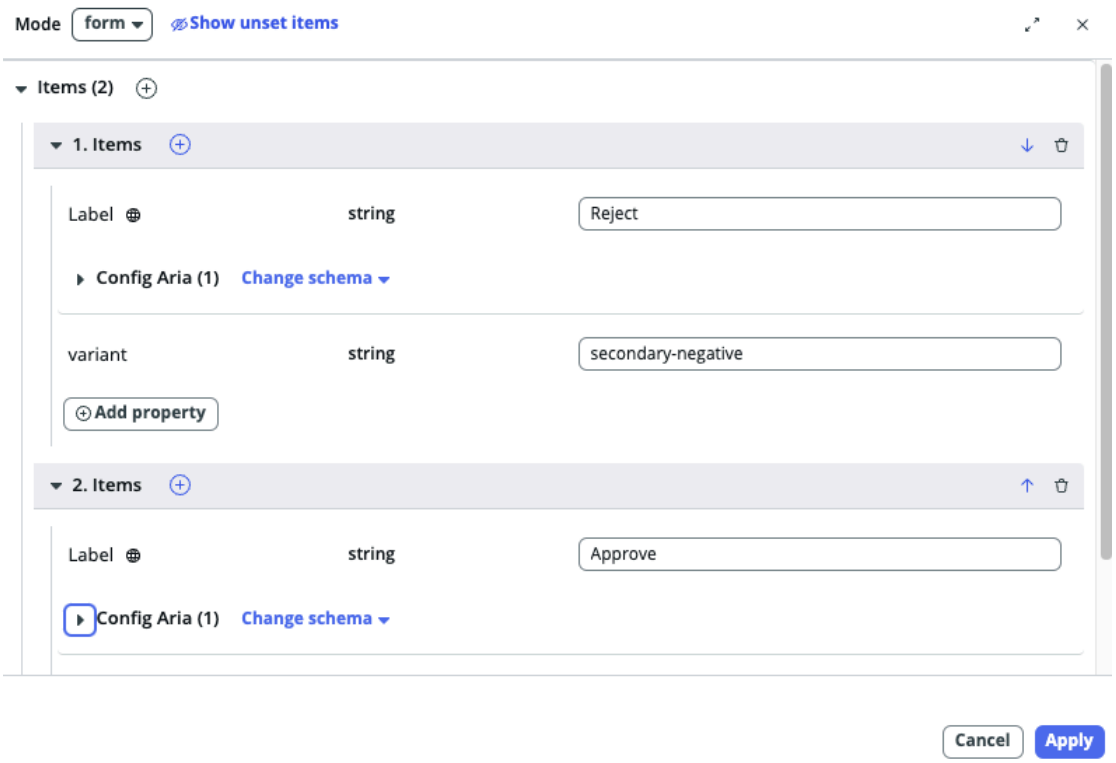
   d. For components with configurable JSON properties, you can use UI Builder's low-code JSON editor to edit component properties without needing to edit JSON code.

   The JSON editor UI displays all available properties, even properties not defined in the dummy data or defined values. You can update properties or add your own custom properties within the JSON editor. For more information about configuring components, see Edit code with the Now Code Editor (advanced feature).

   Note: The low-code JSON editor is only available to properties that have a schema and whose JSON input matches said schema.

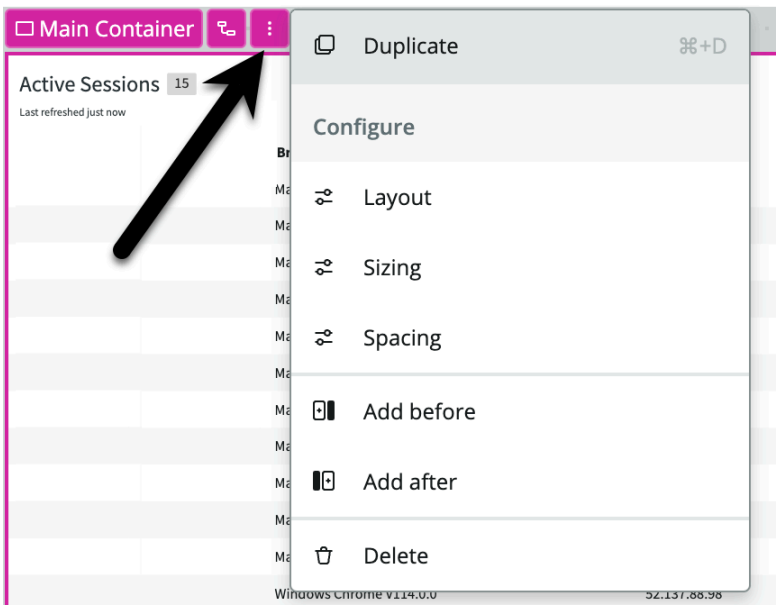
   The low-code JSON editor supports simple objects, simple array properties, complex arrays, arrays of objects, and partially supports complex objects. For complex arrays, you can add, delete, or move the position of array items. You can also select **Hide unset items** to hide objects with empty or null values for a simplified editing experience.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



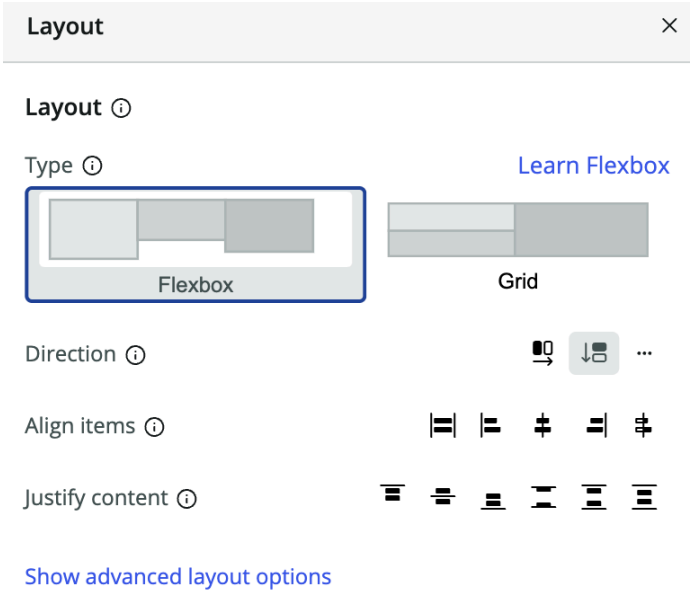
e. When working with a container component, you can edit some styles by using the floating panels.

Select the **Menu** icon to view a menu.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Select the **Layout**, **Sizing**, or **Spacing** option to open a floating panel with the most common options displayed. Drag the floating panel to another location as needed. Only one floating panel can be open at a time.



For more information about configuring components, see Next Experience Components.

7. **Optional:** Select the **Events** tab to add an event handler to your component.
   Add an event handler to add actions to the components on your page. For example, a button component is static and doesn't do anything until you bind an event action to it, such as saving a record. Some components don't have an event action applied to them, such as a heading component. But many components require you to map an event to your component to actually make it perform an action. See Bind an event to a component for more information on how to add event handlers to your component.
8. **Optional:** You can override any styles for a component by adding CSS styling under the **Styles** tab.
   For more information, see Change the default appearance of components.

   Note: Style changes only affect a single component at a time. To change the visual style of all the components in your experience, you must apply a theme to your experience. For more information, see Manage the visual style of UI Builder experiences.

9. **Optional:** Add additional containers to your page to display your components in an organized way.
   For example, one container could have a heading component. Another container below it could include a list component, a button component, and so on.
   a. Select the **+** at the top of your container to add a container component before the existing component and select the **+** at the bottom of a component to insert a container component after it.
   b. Drag a container component from the Components list to your existing container on the page; Hold over the top line of the container to insert the new container before the existing container or over the bottom line of the existing container to add it after.
10. **Optional:** Add more components to your page by selecting the **+** on the top or bottom of the components on the page (the **+** changes to **+ Add**).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



a. Select the **+** at the top of your component to add a component before the existing component or select the **+** at the bottom of a component to insert a component after it.

b. Drag a component from the component list to insert the component before or after an existing component.

11. **Optional:** To make a modal appear when you select a component such as a button, you must add the modal to the component first.
A modal is a confirmation pop-up that appears when you select the component. For example, if you add a button component that deletes a record, you add a modal to ask the user to confirm they want to delete the record. See Create modals in UI Builder for more information.

12. Select **Save** often on your page as you work.

13. View and test your page by selecting the **Preview** button.

What to do next

You have added and configured components on your page. See Dynamically expose data in UI Builder pages (advanced feature) for more information. A data resource in UI Builder is the data that a page fetches to display content in components. Components use data resources as a way to reuse data and configurations across different experiences, and make the components dynamic on a page.

Supported functions in the UI Builder component formula editor

Learn about the various functions supported in the UI Builder component formula editor.

The following table lists the functions you can use in the UI Builder component formula editor. For more information about the component formula editor, see Customize UI Builder pages using components.

Operators available for condition builders

Operator label	Example condition	Equivalent query operator	Example query	Example output
All empty	[Short description] [ALL_EMPTY]	ALL_EMPTY	short_descriptionALL_EMPTY	All records in which there is no value in the **Short description** field.
All equal	[Short description] [ALL_EQ] [Network storage unavailable]	ALL_EQ	short_description=Network storage unavailable	All records in which the value for the **Short description** field is equal to "Network storage is unavailable."
All greater than	[Impact][ALL_GT] [2 - Medium]	>	impact>2	All records in which the **Impact**

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Operators available for condition builders (continued)

Operator label	Example condition	Equivalent query operator	Example query	Example output
				field has a value of **3 - Low**
All greater than or equal	[Impact][ALL_GTE][2 - Medium]	>=	impact>=2	All records in which the **Impact** field has a value of **2 - Medium** or **3 - Low**.
All less than	[Reassignment count][ALL_LT][2]	<	reassignment_count<2	All records in which the value in the **Reassignment count** field is any number less than (but not equal to) **2**.
All less than or equal	[Reassignment count][ALL_LTE][2]	<=	reassignment_count<=2	All records in which the value in the **Reassignment count** field is one of the following: • a number less than 2 • 2
All not equal to	[Impact][ALL_NEQ][1 - High]	!=	impact!=1	All records in which the value in the **Impact** field is anything but **1 - High**.
All not empty	[Impact][ALL_NOTEMPTY]	ALL_NOTEMPTY	impactALL_NOTEMPTY	All records in which the **Impact** field has any value.
All not one of	[Impact][ALL_NOTONEOF][1 - High, 2 - Medium]	ALL_NOTONEOF	impactALL_NOTONEOF1,2	All records in which the **Impact** field is populated by anything except the following values: • **1 - High** • **2 - Medium**

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Operators available for condition builders (continued)

Operator label	Example condition	Equivalent query operator	Example query	Example output
All one of	[Impact] [ALL_ONEOF][1 - High, 2 - Medium]	ALL_ONEOF	impactALL_ONEOF1,2	All records in which the **Impact** field is populated by one of the following values: • **1 - High** • **2 - Medium**
Any empty	[Short description] [ANY_EMPTY]	ANY_EMPTY	short_descriptionANY_EMPTY	Any record in which there is no value in the **Short description** field.
Any equal	[Short description] [ANY_EQ] [Network storage unavailable]	ANY_EQ	short_description=Network storage unavailable	Any record in which the value for the **Short description** field is equal to "Network storage is unavailable."
Any greater than	[Impact][ANY_GT] [2 - Medium]	ANY_GT	impactANY_GT2	Any records in which the **Impact** field has a value of **3 - Low**
Any greater than or equal	[Impact] [ANY_GTE][2 - Medium]	ANY_GTE	impactANY_GTE2	Any record in which the **Impact** field has a value of **2 - Medium** or **3 - Low**.
Any less than	[Reassignment count][ANY_LT] [2]	ANY_LT	reassignment_countANY_LT2	Any record in which the value in the **Reassignment count** field is any number less than (but not equal to) **2**.
Any less than or equal	[Reassignment count][ANY_LTE] [2]	ANY_LTE	reassignment_countANY_LTE2	Any record in which the value in the **Reassignment count** field is one of the following: • a number less than 2 • 2

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Operators available for condition builders (continued)

Operator label	Example condition	Equivalent query operator	Example query	Example output
Any not equal	[Impact] [ANY_NEQ][1 - High]	ANY_NEQ	impactANY_NEQ1	Any record in which the value in the **Impact** field is anything but **1 - High**.
Any not empty	[Impact] [ANY_NOTEMPTY]	ANY_NOTEMPTY	impactANY_NOTEMPTY	Any record in which the **Impact** field has any value.
Any not one of	[Impact] [ANY_NOTONEOF] [1 - High, 2 - Medium]	ANY_NOTONEOF	impactANY_NOTONEOF1,2	Any record in which the **Impact** field is populated by anything except the following values: • **1 - High** • **2 - Medium**
Any one of	[Impact] [ANY_ONEOF][1 - High, 2 - Medium]	ANY_ONEOF	impactANY_ONEOF1,2	Any record in which the **Impact** field is populated by one of the following values: • **1 - High** • **2 - Medium**
CONCAT	CONCAT(value)	CONCAT	CONCAT("Welcome, ",@context.session.user.firstName)	Create a new string that combines all supplied strings into one
EMPTY	EMPTY(value)	EMPTY	EMPTY(@context.session.user.roles)	Returns true if value is null or undefined
IF	IF(if, then, else)	IF	IF(@context.props.bare,"bare page","not bare page")	If condition is true, return the value then. If condition is false, return the value else.
LEN	LEN(list)	LEN	LEN([1,2,3])	Returns the number of items in the array

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Operators available for condition builders (continued)

Operator label	Example condition	Equivalent query operator	Example query	Example output
Pick	PICK(array, field)	Pick	PICK(@context.session.user_preferences,"name")	Creates a new array where each item is picked from field in each item in the array. If the field does not exist, the item in the new array will be EMPTY
Range	RANGE(from, to)	Range	RANGE(1,10)	Creates an array of numbers, starting with from, up to, incrementing by step
Sum	SUM(array)	Sum	SUM([1,2,3])	Starting at 0, add the number value of each item in the array and return the resulting summation
Translate	TRANSLATE(text)	Translate	TRANSLATE("Welcome back")	Returns the string from the first argument after the characters specified in the second argument are translated into the characters specified in the third argument.
Where empty	[Short description] [WHERE_EMPTY]	WHERE_EMPTY	short_descriptionWHERE_EMPTY	Extract records where there is no value in the **Short description** field.
Where equal	[Short description] [WHERE_EQ] [Network storage unavailable]	WHERE_EQ	short_descriptionWHERE_EQ^Network storage unavailable	Extract records where the **Short description** field is equal to "Network storage is unavailable."
Where greater than	[Impact] [WHERE_GT][2 - Medium]	WHERE_GT	impactWHERE_GT2	Extract records where the **Impact** field has a value of **3 - Low**

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Operators available for condition builders (continued)

Operator label	Example condition	Equivalent query operator	Example query	Example output
Where greater than or equal	[Impact] [WHERE_GTE][2 - Medium]	WHERE_GTE	impactWHERE_GTE2	Extract records where the **Impact** field has a value of **2 - Medium** or **3 - Low**.
Where less than	[Reassignment count] [WHERE_LT][2]	WHERE_LT	reassignment_countWHERE_LT2	Extract records where the value in the **Reassignment count** field is any number less than (but not equal to) **2**.
Where less than or equal	[Reassignment count] [WHERE_LTE][2]	WHERE_LTE	reassignment_countWHERE_LTE2	Extract records where the value in the **Reassignment count** field is one of the following: • a number less than 2 • 2
Where not equal	[Impact] [WHERE_NEQ][1 - High]	WHERE_NEQ	impactWHERE_NEQ1	Extract records where the value in the **Impact** field is anything but **1 - High**.
Where not empty	[Impact] [WHERE_NOTEMPTY]	WHERE_NOTEMPTY	impactWHERE_NOTEMPTY	Extract records where the **Impact** field has any value.
Where not one of	[Impact] [WHERE_NOTONEOF] [1 - High, 2 - Medium]	WHERE_NOTONEOF	impactWHERE_NOTONEOF1,2	Extract records where the **Impact** field is populated by anything except the following values: • **1 - High** • **2 - Medium**
Where one of	[Impact] [WHERE_ONEOF] [1 - High, 2 - Medium]	WHERE_ONEOF	impactWHERE_ONEOF1,2	Extract records where the **Impact** field is populated by one of the following values:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Operators available for condition builders (continued)

Operator label	Example condition	Equivalent query operator	Example query	Example output
				• 1 - High • 2 - Medium

Add repeaters to components

In UI Builder, use repeaters to repeat one or more components with results from a data resource.

Before you begin

Role required: ui_builder_admin

About this task

The repeater component acts as a basic loop that repeats the data you provide in multiple components within UI Builder. Repeaters use an array or an array of objects. For example, the array `[{"task": "A"},{"task": "B"}]` repeats the content inside it two times. Repeaters enable you to bind values to a data array property. `@item.value.{property_name}` binds the values to the component inside the repeaters. If you want to bind a task, you can bind it as `@item.value.task`, and the repeater displays the correct value.

Components inside a repeater repeat the number of items in the data source, regardless of whether the component output is made dynamic or not. For example, say that you place a Header component inside a repeater with three data elements but don't change the Label field. Then you see three instances of the Header with the same output Label.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Open the editor for the page where you want to add the repeater.
   If you haven't created a page for your experience, see Create a page in UI Builder for more information on how to create a page.
4. Connect a data resource to your page.

   For example, add the **Look up multiple records** data resource to your page. See Add and configure data resources to a page for more information on connecting a data resource.

   a. In the data resource drawer, select **+ Add data resource**.

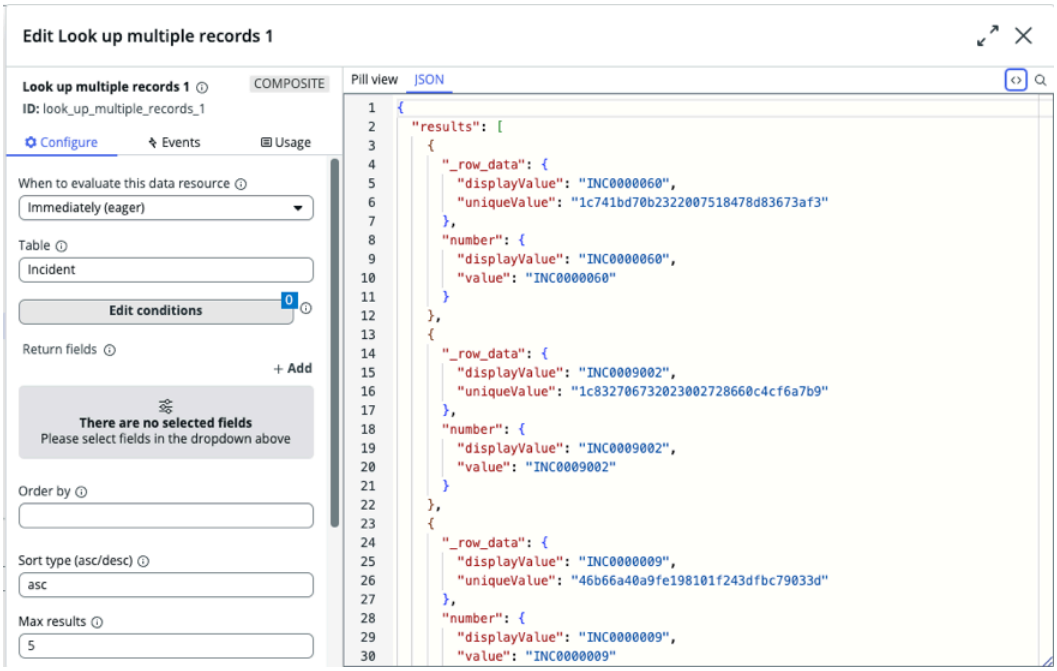
   b. In the modal, search for the **Look up multiple records** data resource and select **Add**.

   c. In the **Table** field, enter `Incident`.

   d. In the **Max results** field, enter `5`.

   The **Look up multiple records** data resource is configured.

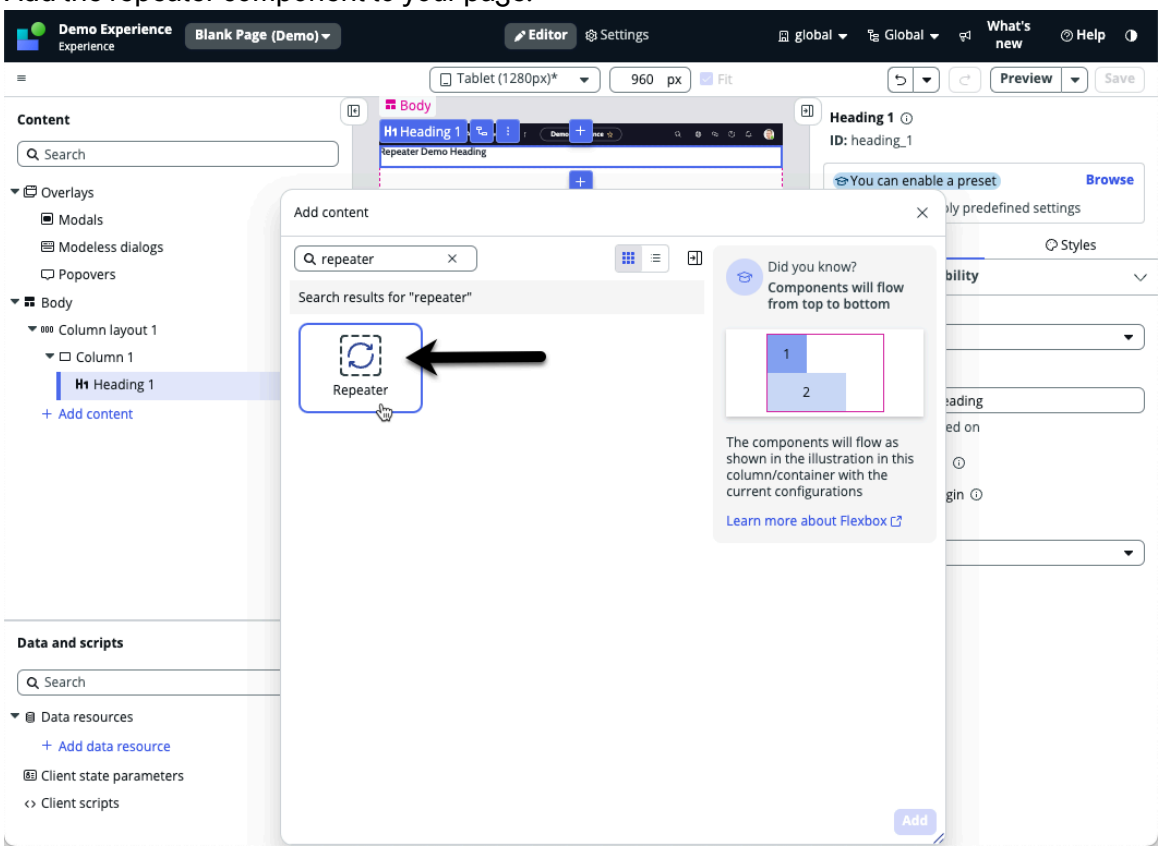
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



e. Close the modal.

5. Select an existing container or create a column layout.
See Organize components in UI Builder pages for more information.

6. Add the repeater component to your page.



For information about adding components, see Add and configure components.

7. In the **Configure** tab, select **None - Configure the component manually**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

8. In the **Configure** tab, add an array that defines the data for repeated components.

   a. Point to **Data array** and select the Bind data icon ( ).

   b. Under **Data types**, select **Data resource**.

   c. In the next column, select the **Look up multiple records 1** pill.

   d. Under **Pill view**, double-click or drag **results** to add the **@data.look_up_multiple_records_1.results** output.

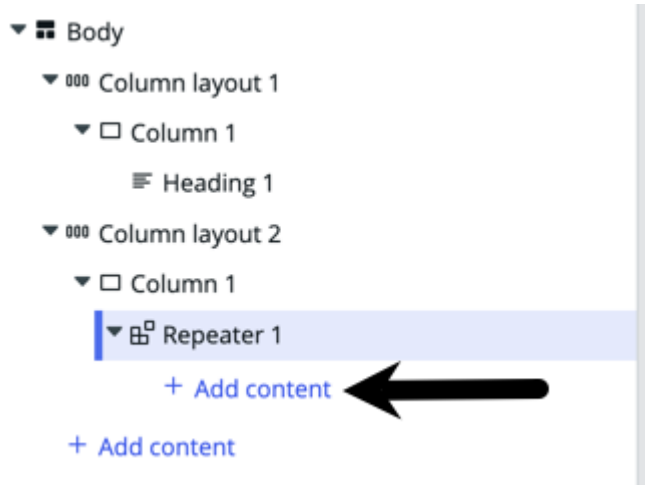
   e. Select **Apply**.

      The results from the data resource are bound to the repeater component. Within the repeater, one component represents each result.



9. Under the repeater that you added in the previous step, select **+ Add content**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



10. Add a **Heading** component.
11. In the **Configure** tab, select **None - Configure the component manually**.
12. Configure the component you nested in the repeater.

    For example, bind the **displayValue** to the component. See Connect data to your components for more information.

    a. Move your cursor to the **Label** field and select the Bind data icon ( ).

    b. Under **Data types**, select **Repeater**.

    c. Under **Pill view**, select the **value** pill.

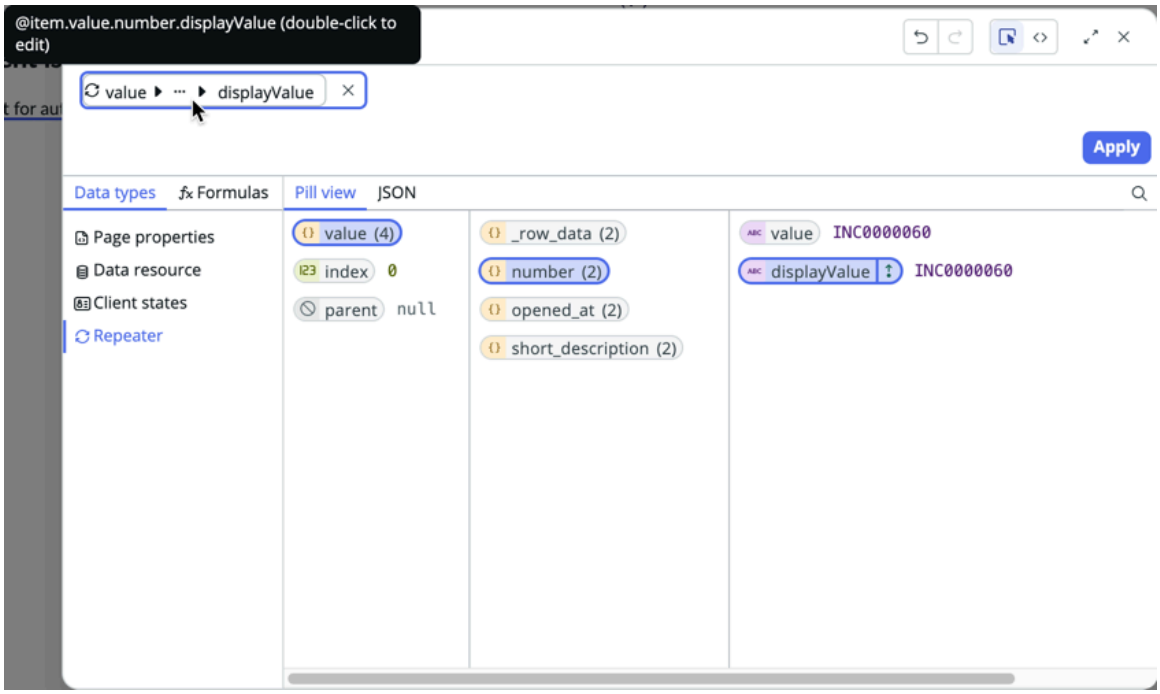
    d. In the next column, select the **number** pill.

    e. In the next column, double-click or drag **displayValue** to add the **@item.value.number.displayValue** output.

    f. Select **Apply**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The **displayValue** is bound to the Heading component.



13. Select **Save**.

    The component you bound to the data resource appears on the stage and is repeated five times, once for each result returned from the Incident table.

14. View and test your page by selecting the **Preview** button.
    A tab opens to display the page preview with repeated data.

Optimize page loading performance (Advanced Feature)

Performance settings enhances leading speeds and user engagement by displaying high-priority content first, making key components interactive sooner for a faster perceived performance.

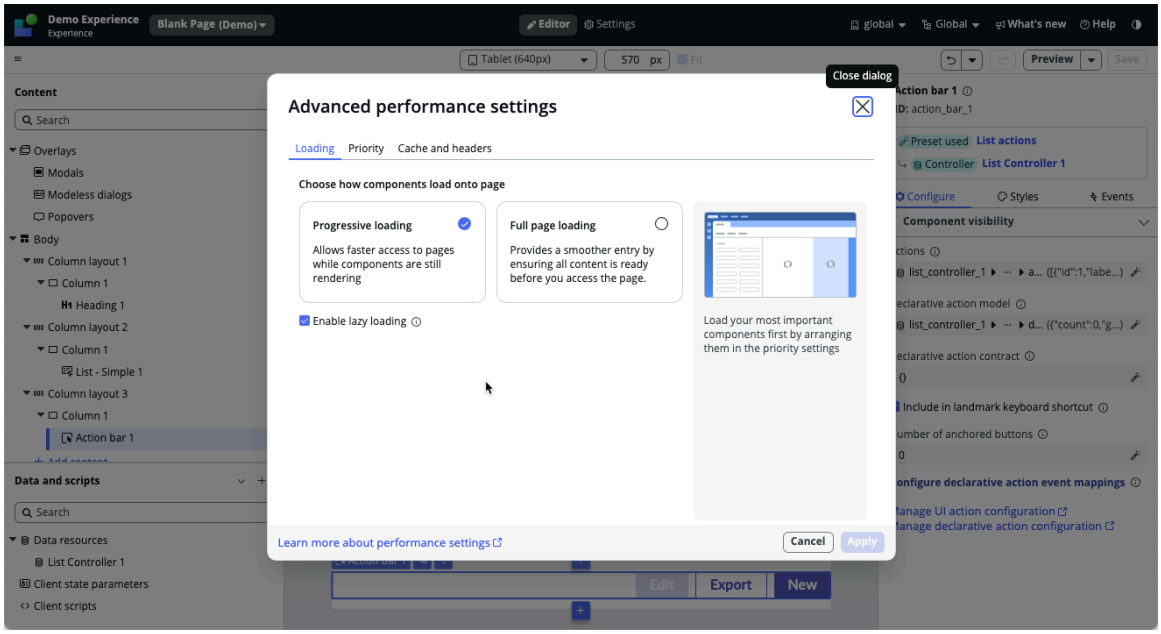
Before you begin

Role required: ui_builder_admin

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create or open a page.
4. Add a component to your page.
   You need a component on your page before you can adjust performance settings.
5. Select the menu icon.
6. Select **Manage performance settings**.
   The **Advanced performance settings** modal opens.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



7. Select how components load onto a page.

Choice	Description
Progressive loading	Enable quicker page access while compo nents continue to load.
Full page loading	Ensure a seamless experience by loading all content before you access the page. Enter the maximum time you want to wait before dis playing content in the **Page loader threshold duration** field.

8. Select the priority in which components load on a page.
   a. Select the **Priority** tab.
   b. Select the component you want to give priorty to.
   c. Check **Make high priority**.
9. Select which data to cache when a page loads.
   a. Select the **Cache and headers** tab.
   b. Select a data resource from the list.
   c. Check **Activate data policy**.
10. Select **Apply**.

Dynamically expose data in UI Builder pages (advanced feature)

Sync data between ServiceNow tables and data with the pages you build with UI Builder. Pages display synced data in real time and update data/tables when a user inputs information.

Learning about data resources in UI Builder

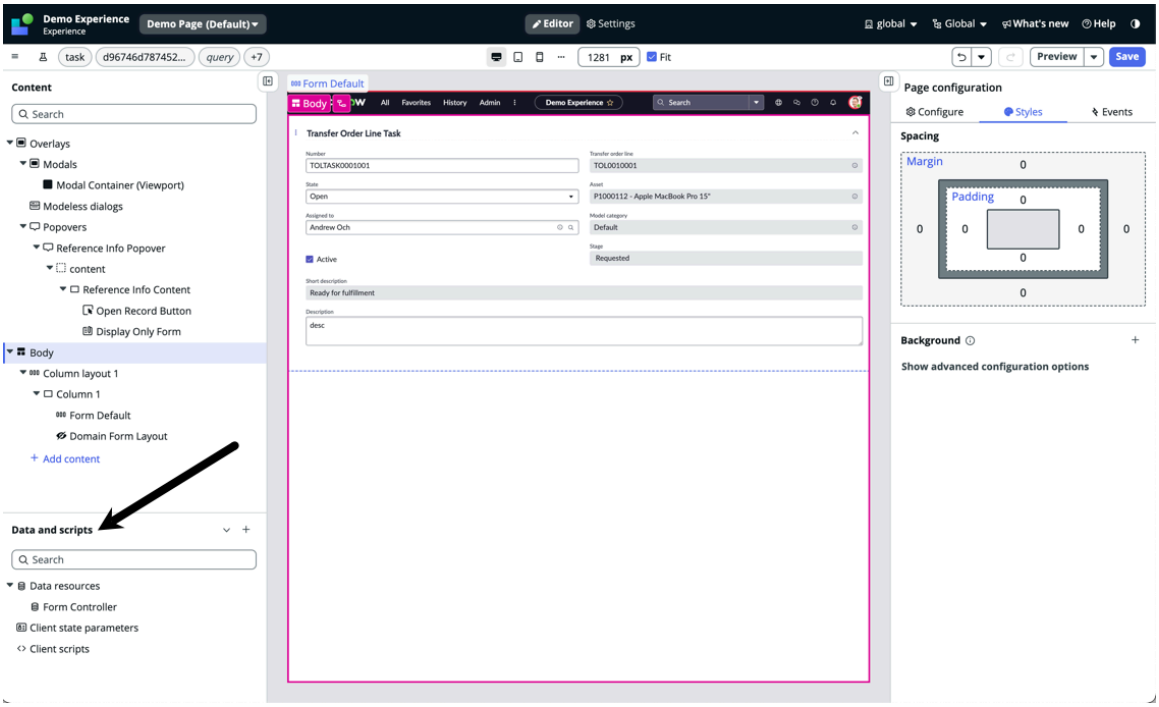
UI Builder syncs ServiceNow tables/data using data resources. A data resource fetches the data that UI Builder uses to display information in a component. UI Builder components use

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

data resources to sync data across different experiences. Data resources make the data in components dynamic, which means that you don't have to recreate data for every page.

Data resources are found in the data resource drawer. The data resource drawer is where you can add and configure a data resource for your page. After you configure the data resource, you can sync the data between components on your page and ServiceNow tables/data.

Data Resource Drawer



The data resource drawer contains three sections:

- Data resources: The data resources that are part of the experience
- Configuration/Events: Configured data resources and events for the experience
- Preview: JSON for the information returned by the data resource

You can bind the configuration properties for components, other data resources, client scripts, client state, and events to these data resources.

Set conditions for a filter in your data resource. For more information, see Connect data to your components.

How data resources work in UI Builder

Data resources fetch data from Glide, GraphQL, and REST APIs, then transform it for use in a component on a UI Builder page.

Components use both inherited and local data resource instances. Inherited data resources are automatically loaded into a UI Builder page, and local data resource instances can be added and configured. Data resources are evaluated based on specified input values to make sure the right data is retrieved. For more information about inherited and local data resources, see Inherited versus local data resources in UI Builder.

Local data resources in UI Builder

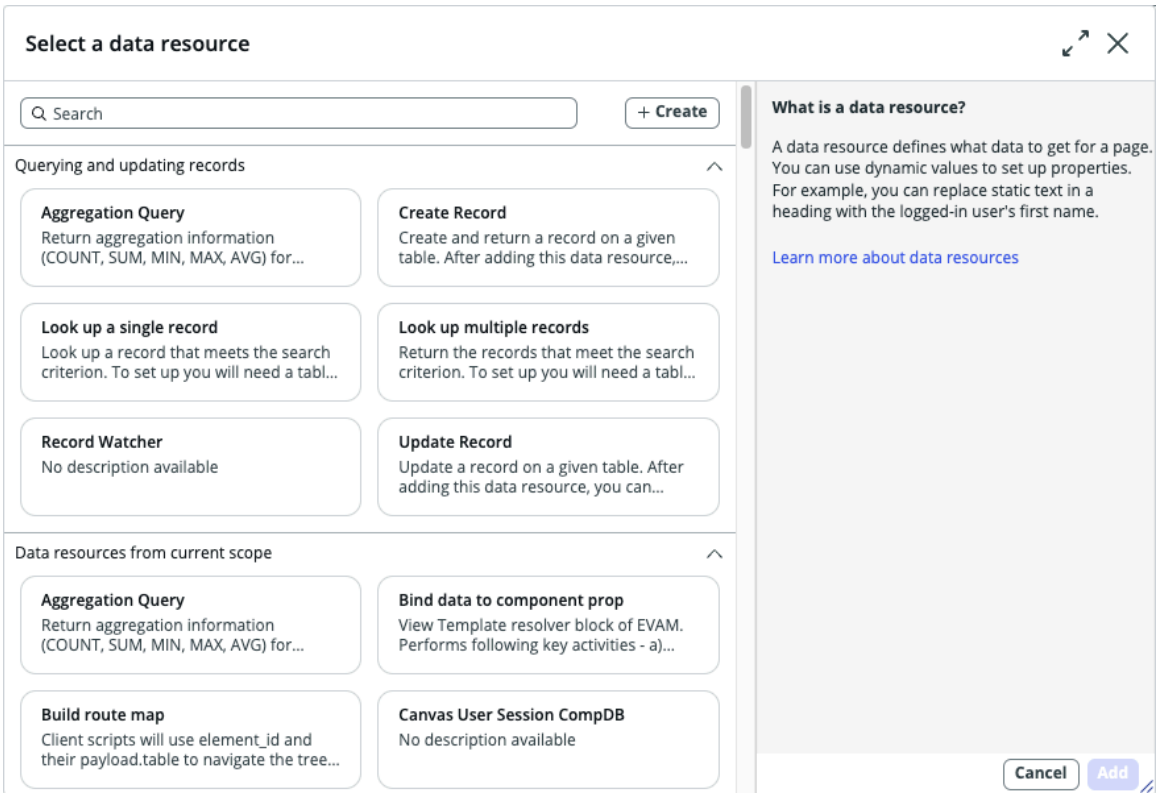
You can select local data resources, such as server data, operations, transforms, or client data, like the gForm API, to bring data to the UI Builder page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: Only one GlideForm is supported per page in UI Builder. For more information about GlideForm, see the ServiceNow Developer site.

Data resources are organized by application in the data resource drawer. They are then further categorized by the data resource type like Server data or Transform. For example, the Global application has several data resources, but the Customer Service Management (CSM) Workspace application has only a few data resources.

Data resources



You see different data resources depending on the application that you are in. If you select the Global application, you will see different data resources under Server data than if you select the CSM Configurable Workspace application.

Using data binding in UI Builder

Data binding enables you to create dynamic pages by syncing pages and components to data resources. You can bind data to a component to retrieve data from the back end, and use field parameters to get properties from the URL. Changing the URL parameters enables you to create dynamic pages that show different data depending on the parameters.

You can bind data to a component in the following ways:

Context binding

Use URL parameters to connect parts of the URL with your page's properties. For example, you can link the table name from the URL to your component by using the @context syntax, like this: `@context.props.table`.

Imagine you have a UI Builder page with a required field named `table`. The URL for your page might look like /demo/page/<table-name>. The <table-name> could be something like `incident`. The data can also come from parent data resources or be local properties specific to the page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

To link your component properties, other data resource properties, or event details to the page property, you use a `@context.props.table` binding. Be sure to either provide a test value in the URL or set a fixed value for that property in your page's configuration if you are using context binding.

Data resource binding

Use data resources to fetch data from the back end of your instance, such as Client state, GraphQL, or a REST API. These data sources have properties that can be linked to elements on a UI Builder page.

For instance, if you are using the Lookup Record data resource, you can utilize it in a button component. You might use a data expression in the "label" property like this: `@data.lookup_record_1.result.number.displayValue`.

Component binding

Use component binding to connect one component to another. Let's say that you have a List Menu component on the UI Builder page. The List Menu reveals the currently chosen list to other components on the same page. These other components can access the data by linking to it using an expression such as `@elements.list_menu_1.selectedListId`.

Client state parameter binding

Use client state parameter binding to connect and synchronize data between a client-side application and UI Builder components. Parameter binding allows the client state to automatically update data in components, and vice versa. Use `@state` syntax to bind a state property to a client state parameter.

Types of data sources available in UI Builder

You can use the following data resource types that are shown in the table.

Data resource types

Data resource type	Description
Controller	Encapsulates data and event logic and enables presets for components.
GraphQL	GraphQL queries and mutations that are executed.
Transform	Script that transforms the input data into another format.
Client state	Client-side data resources that include the client information, domain-specific states or logic, user preferences, and so on.
Composite	Single reusable data resource that contains multiple data resources.
REST	Data resources that are made through REST API requests.

Inherited versus local data resources in UI Builder

Inherited data resources share information from the surrounding parts of a UI Builder page such as an application. Imagine your page is in a large frame, and it gets some information from the frame or other parts around it. You can use this info by connecting it to your page's properties,

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

kind of like linking puzzle pieces together. If you get this information from the frame, you don't have to get it again yourself.

Local data resources are items you add to a UI Builder page yourself. Imagine you're making a page for travel requests. You can sync employee data to a list component so employees can request trips linked to their own info.

To use these data resources, either bring them in from an app or create them in the ServiceNow platform. Then, in UI Builder, make these resources available for your components. Next, tie the data to your component, so it can work with it. For instance, you could have a set of records, expose it in UI Builder, and link it to a component. From there, configure the component to do things with the data, like saving new records.

Once your page is syncing data, you connect it to the part of the page that needs it such as a component. It's like making sure the right puzzle piece fits in the right spot. Then you can tell that part of the page what to do with the data. For example, you might use it to add new travel plans for employees.

UI Builder Data resource properties

When you add a data resource in UI Builder, it's like adding a tool that knows where to get information from. Data resource properties are the instructions that tell a data resource how to bring data into UI Builder pages. These properties tell a data resource which data to get, how to organize it, and what conditions to follow. For instance, you can use these properties to specify which tables to look in, how to sort the data, and more. These properties are what make the data resource work correctly and give you the data you need.

After adding a data resource, you can choose to load it either eagerly or explicitly. Use the descriptions below to determine which approach best fits your page. This decision can impact performance, user experience, and resource efficiency, so it's important to understand the differences.

Immediately (eager evaluation)

Data loads automatically when the page or component loads.

Useful for core content the user expects to see right away.

May slow down loading of pages and components by fetching unused data.

Only when invoked (explicit evaluation)

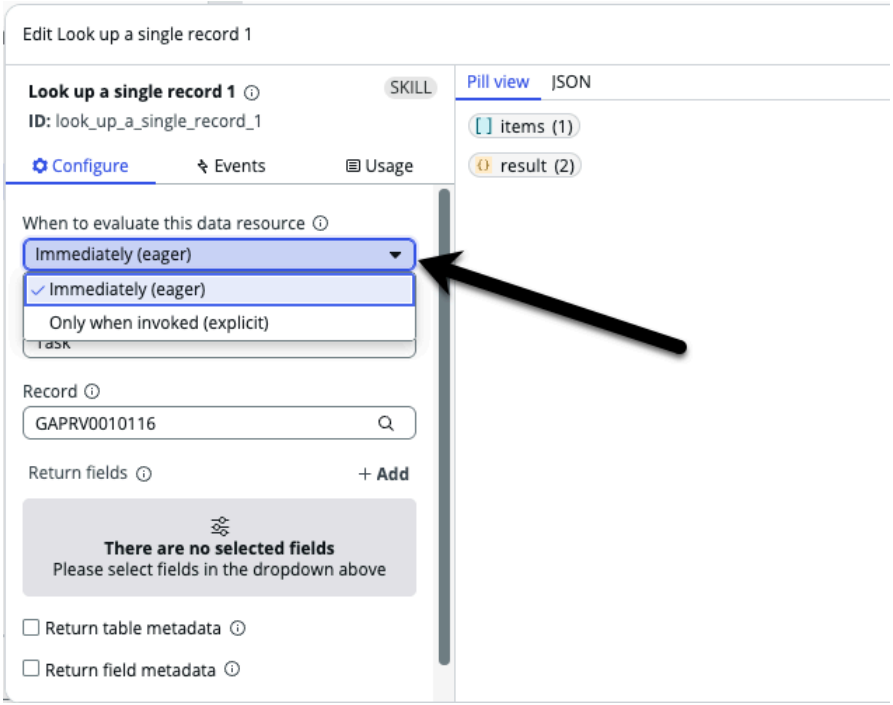
Defers data loading until a user takes a specific action. For example, clicking a button or entering data in a field.

Saves resources by only fetching data when needed.

Speeds up initial page loading.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Data resource evaluation options



UI Builder Data resource scripts

Data resource scripts are special instructions you give UI Builder to return specific pieces of data. For example, if you're dealing with a list of products, a script could tell a data resource to show only products that are available in stock or to arrange them in a particular order. Data resource scripts enable you to customize how UI Builder treats the information, such as adding extra rules or actions.

Add and configure data resources to a page

Add data resources to dynamically expose data from tables, records, or other elements on your page in UI Builder.

Before you begin

Role required: ui_builder_admin

Choose the application containing your data resource. The Global application contains the most common data resources. The data resources relate to Server data, Operations, and Transform. You can add a data resource such as Look Up Records, Table route map, GlideRecord Query, and so on.

About this task

In the data resource section of UI Builder, you can add and configure local data resource instances for your page. Local data resource instances dynamically expose data to components. If you have any inherited data resources, they are listed in the inherited data resource pane and are read-only. You don’t configure inherited data in UI Builder. Data resources from any application can be added to any page unless restricted by security permissions. You can add multiple local instances of the same data resource, if you need different configurations of the same data resource on your page.

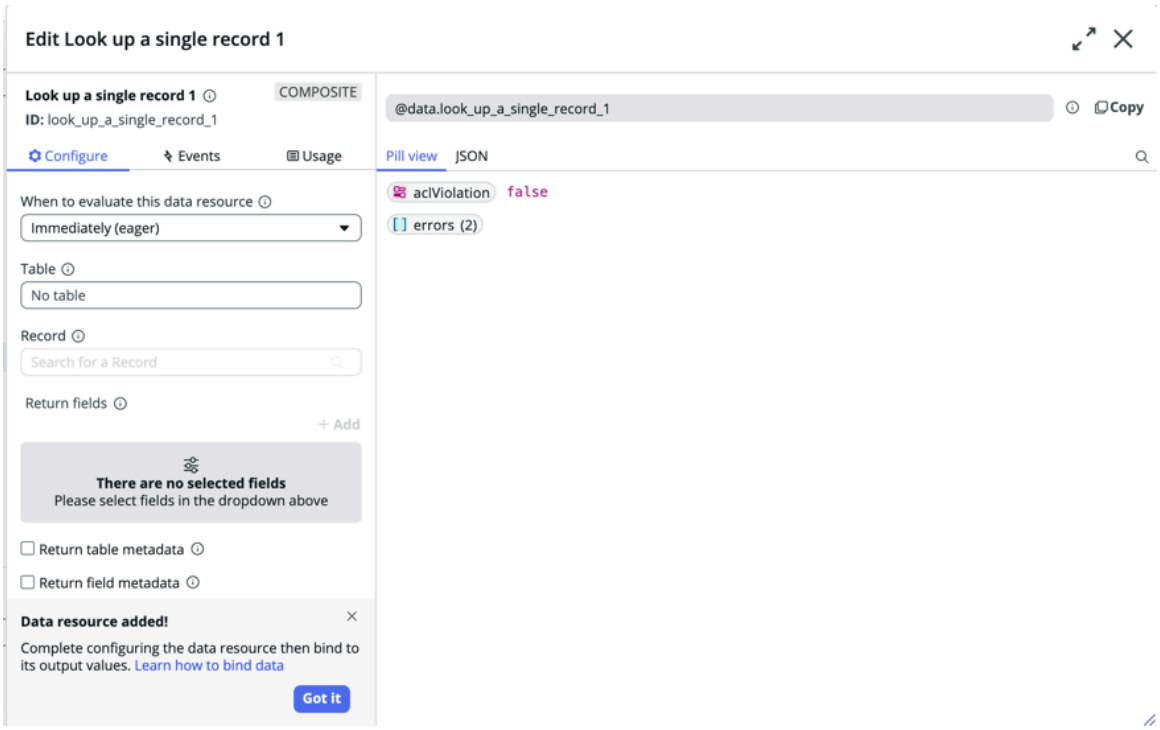
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

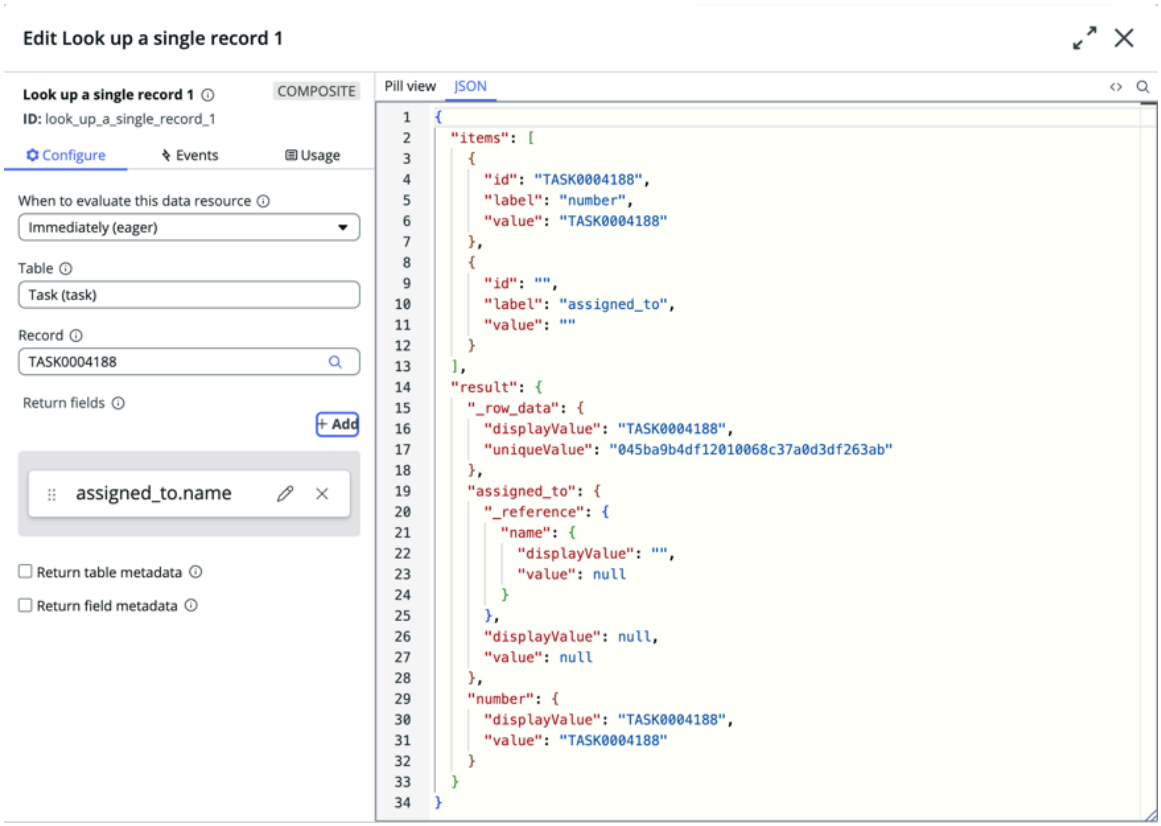
See Configure how users interact with your applications in UI Builder for more information on creating experiences.

3. Create or open a page.
4. Add a component to your page.
   You need a component on your page before you can bind a data resource to it.
5. Select the **+** icon in the data resource drawer.
6. Select **Data resource**.
7. Select a data resource to add to your page.
   For example, you could select **Look up a single record**.
8. Select **Add** to add the data resource to your page.
   The Data Resource configuration modal appears.

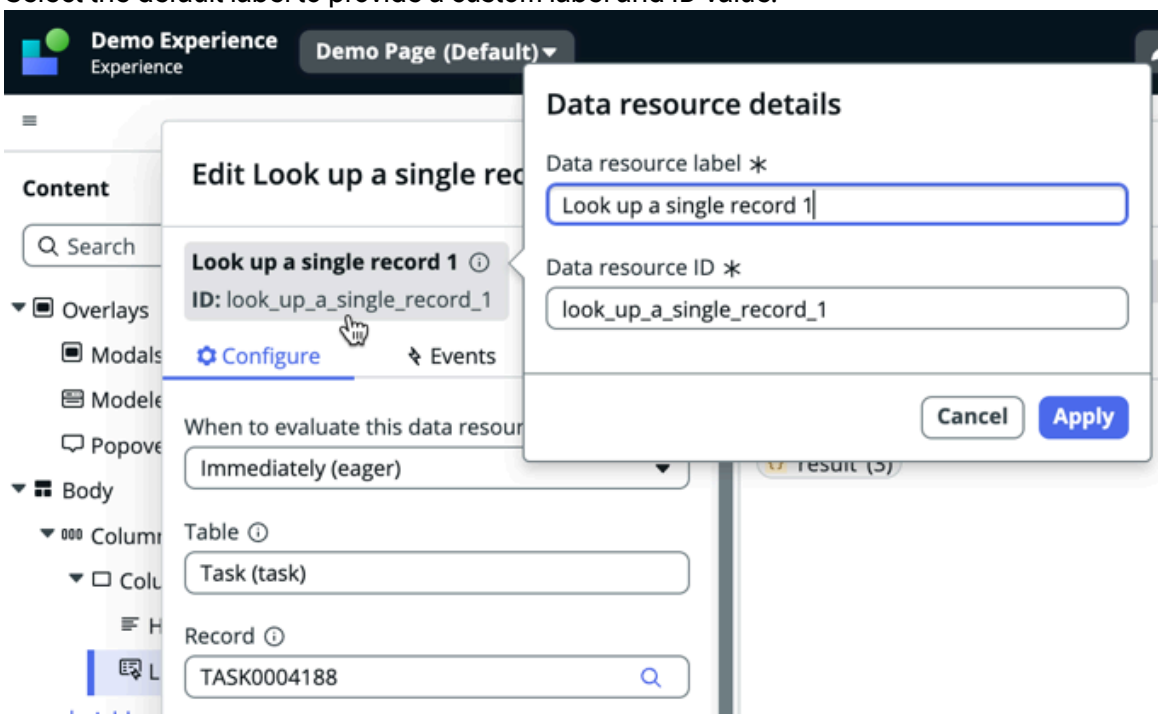


9. Fill in the required information for the data resource in the **Configure** tab to expose the data you want.
   The configuration fields vary depending on which Data Resource instance is selected. The **Configure** panel and preview window may display errors when a Data Resource is added to a page. UI Builder attempts to evaluate Data Resources when they're added to a page. The errors remain until the required Data Resource fields have been filled.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



10. Select the default label to provide a custom label and ID value.



The data resource appears in the data and scripts drawer.

11. Select **Save** in the UI Builder header.

What to do next

Now that you have a data resource connected to your page, you can bind the data to a component or bind an event to the data resource. For more information, see Connect data to your components and Bind an event to a data resource.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add Now Assist skills to your page

Add skills to enhance your UI Builder page with Now Assist's generative AI capabilities.

Before you begin

Role required: ui_builder_admin

Note: Skills must be enabled and activated before they can be added to a page. For more information, see:

- Configure skill deployment settings: Enable the skill to make it available for selection
- Activate a skill: Activate the skill so it can be selected

About this task

You can add Now Assist skills to your page, component, or controller. To trigger a skill, you must create an event to execute the skill while passing the appropriate payload. After configuring the required inputs, the skill is ready to use.

In this example, you will:

1. Add a skill to the page
2. Add an input component to capture user text
3. Configure an event for the input component to pass the entered text to the skill and execute it
4. Add a text component to show the skill's output

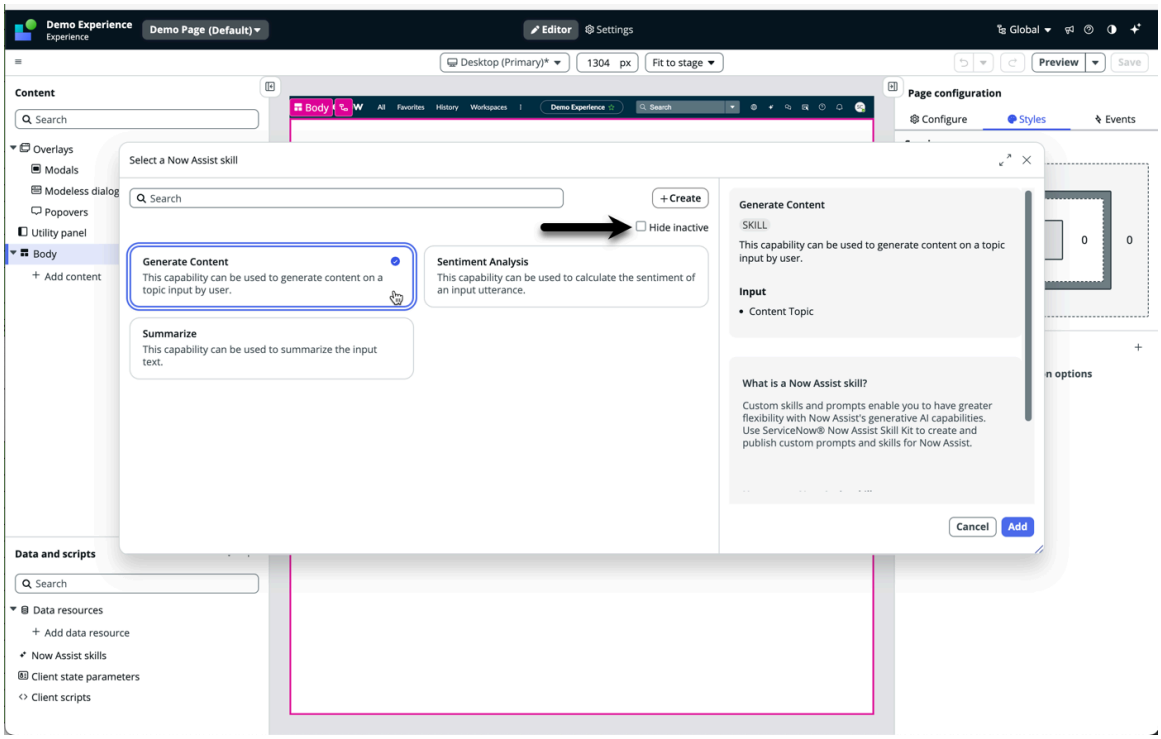
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create or open a page.
4. Hover over **Now Assist skills** in the **Data and scripts** section and select the **+** to the right.
5. Select a skill and select **Add**.
   The following skills are available by default:
   - **Generate Content**
   - **Sentiment Analysis**
   - **Summarize**

   For this example, select **Generate Content**.

   Tip: Select **Hide inactive** to remove inactive skills from view.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Note: Skills that are enabled in AI Skill Kit but not activated in AI Admin Hub appear in the list, but aren't selectable. For more information, see Now Assist Skill Kit and Now Assist Admin console.

6. Select the **X** in the upper right to close the **Edit Generate Content** dialog.
7. Select **+ Add content** in the content tree and add a component from the toolbox.
   In this example, add an **Input** component.
8. Add the **Execute** event handler to the component that you added in the previous step.

   a. Select the **Events** tab in the configuration panel.

   b. Select **Add handler** under **Text entered**.

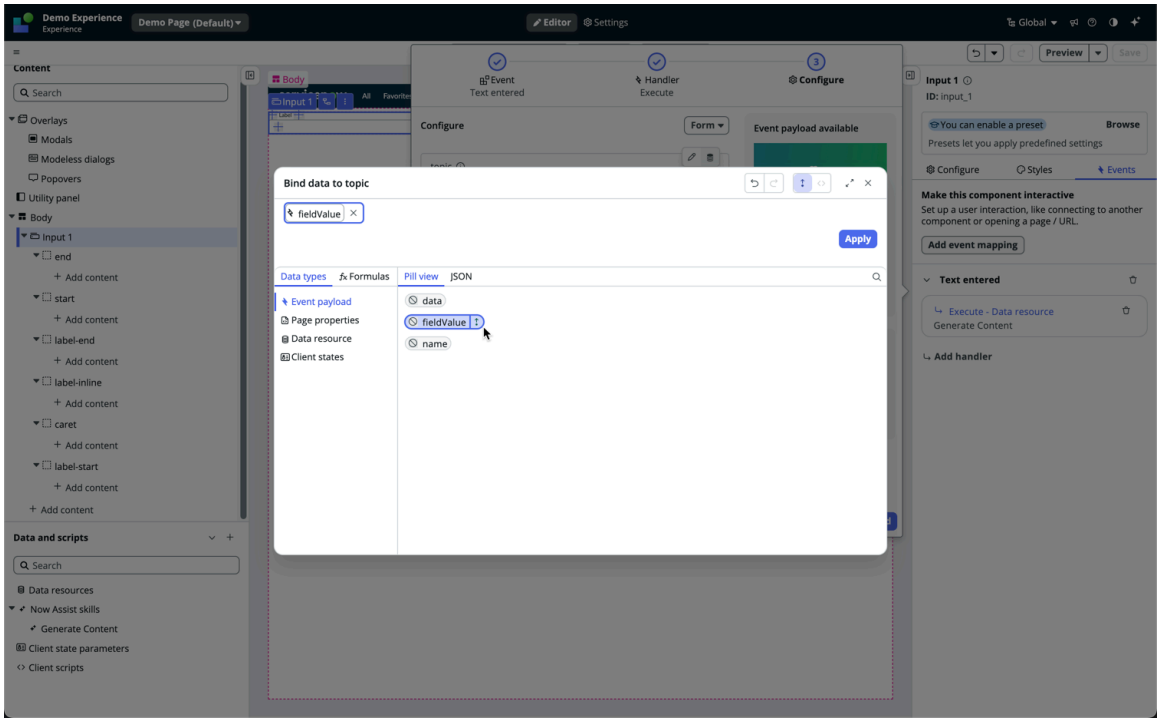
   c. Select **Execute** under **Generate Content (1)** and select **Continue**.

   d. Hover over **topic** and select the bind data icon to configure the input for the skill.

   e. Double-click the **fieldValue** pill to select it as the input source and select **Apply**.

      The **fieldValue** property captures what the user types in the input field. By using this property as the input to the skill, you set the skill to trigger whenever the user enters text.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



f. Select **Add**.

9. Add and configure a second component to display the output for the skill.

a. In the content tree, hover over **Input 1**, select the **Configure** (three vertical dots) icon, then select **Add after**.

b. Add a **Stylized text** component.

c. Select **Cancel** to close the preset window.

d. **Optional:** Change the **HTML tag** property to **Paragraph** (for better readability in this example).

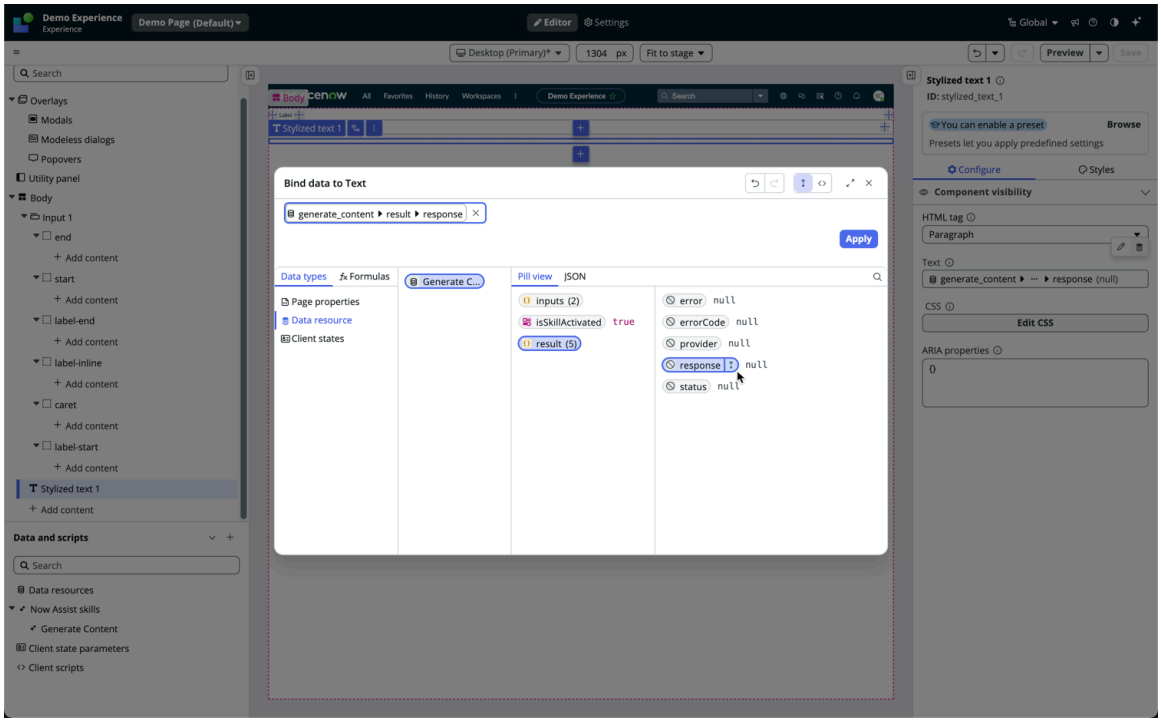
e. Hover over the **Text** field in the configuration panel and select the bind data icon .

f. Select **Data resource** on the left.

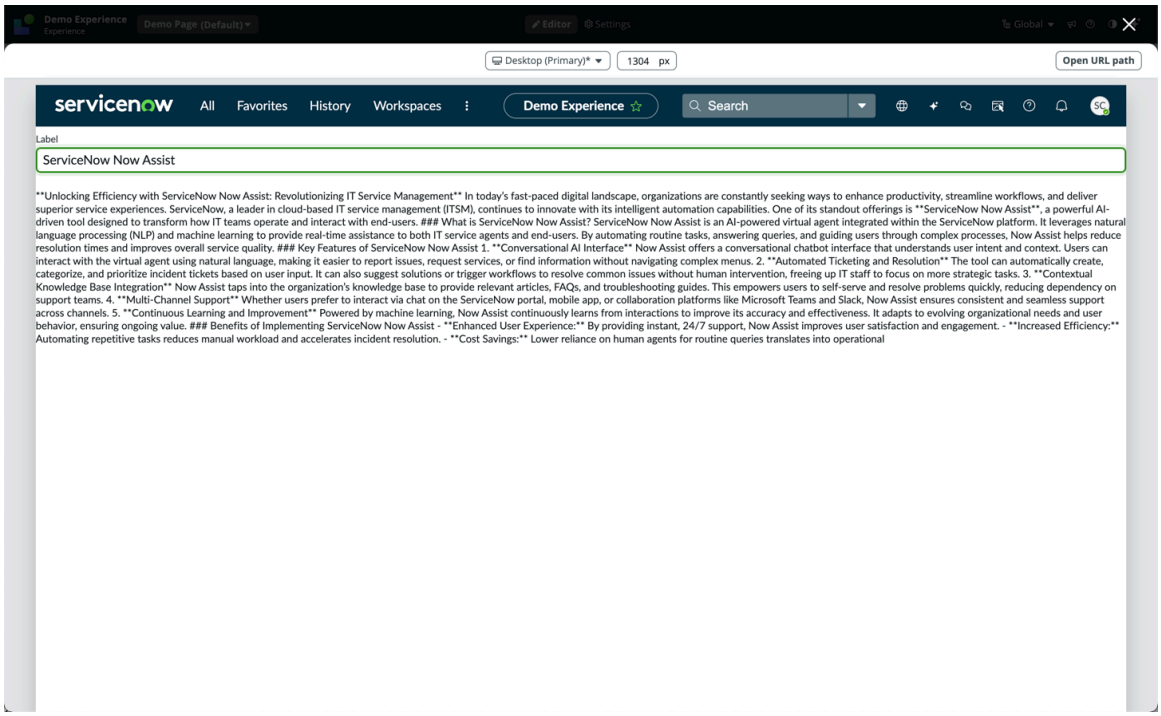
g. Navigate through the pills by selecting **Generate Content**, then **result**, then double-clicking **response** to complete the selection.

h. Select **Apply**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



10. Select **Save**.
11. Select **Preview** to test the skill.



Result

When you view the page and type in the input field, your text input triggers the **Generate Content** skill to execute and return content on the topic in the input field.

Bind data to UI Builder pages using controllers (advanced feature)

Controllers simplify the linking of data and event logic to enable component presets in UI Builder.

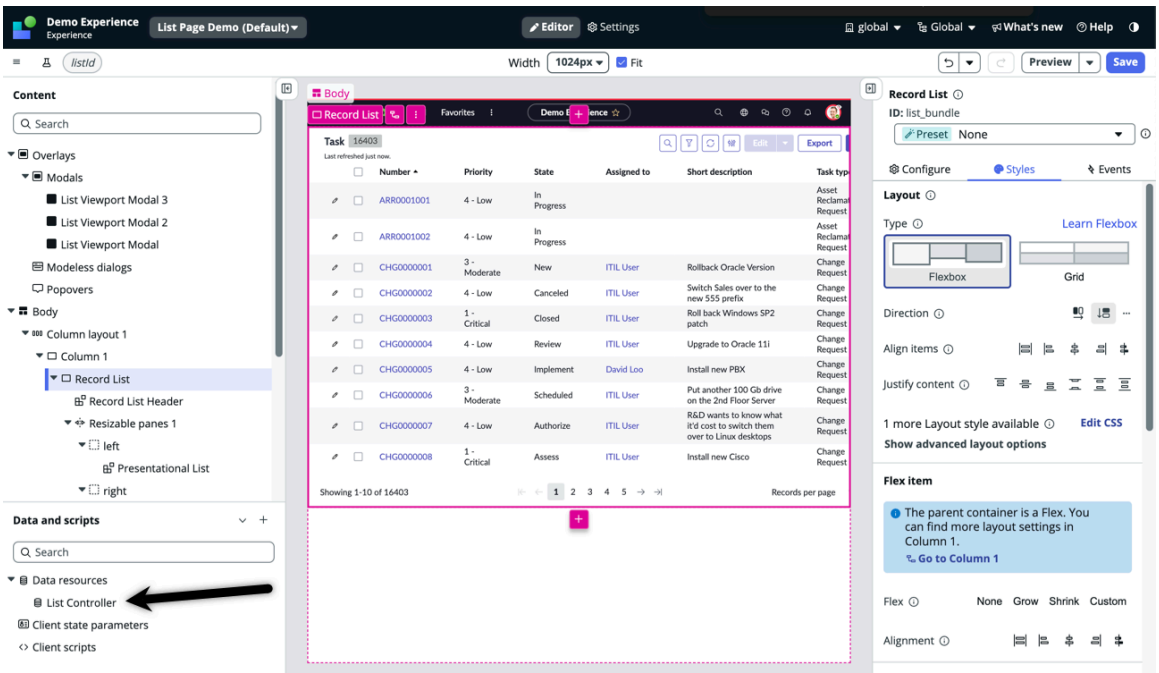
Controllers enable page builders to easily connect data and scripts to their pages in UI Builder. A controller is a type of data resource that component presets use to bind data to components. Controllers differ from other data resources in that they contain data and event logic that enables

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

presets for components. A controller brings data from the server to a component, and it brings updated data back to the server based on interactions with the component. For more information about component presets, see Automatically configure components using presets.

Controllers are added by default when you use a UI Builder page template. You can add controllers to UI Builder pages without a controller within the data resources drawer or by selecting a component preset after adding a component to a page. You can view which controllers are configured on your page by opening the data resources drawer.

Controller in the Data resource drawer



Types of data controllers in UI Builder

Data controllers

Data controllers decide what information should be displayed on UI Builder pages. Data controllers use data resources to sync information in real time and update data/tables when a user inputs information. You can manually add data controllers to a page.

UI controllers

UI controllers are added to pages when using UI Builder page templates and cannot be added manually.

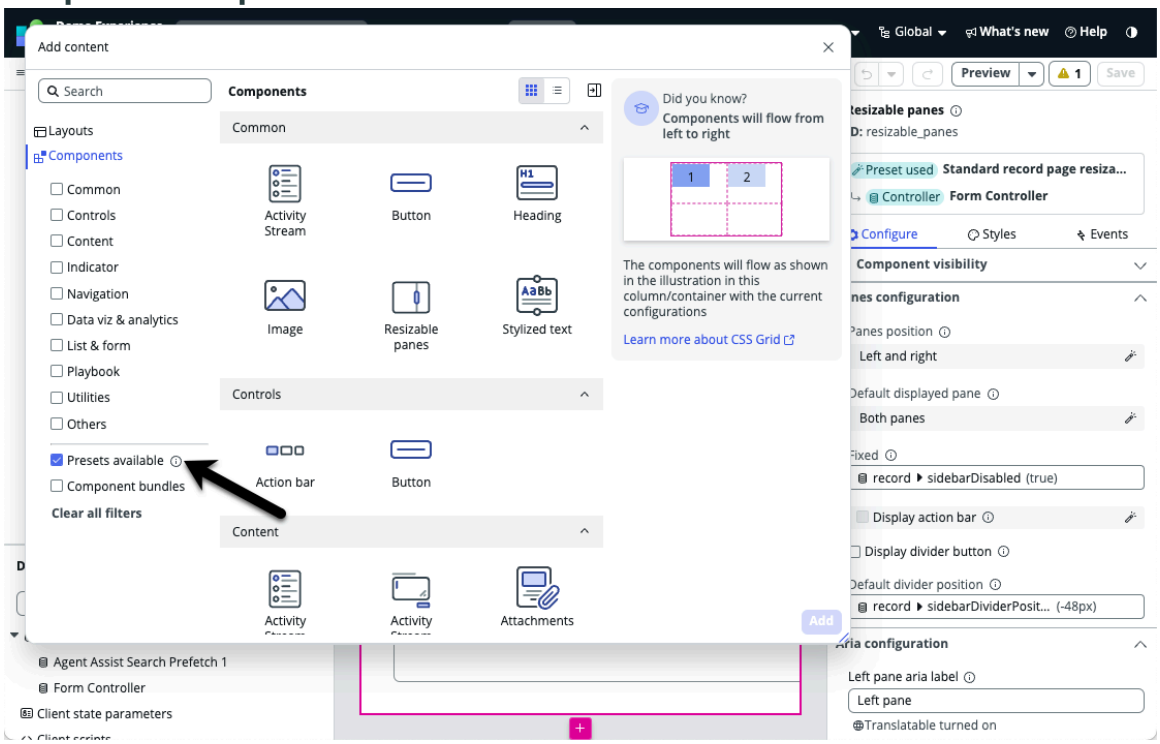
Using controllers with presets in UI Builder

You can add multiple controllers to a UI Builder page, but you can't use the same one twice. If you try to use a component preset that needs a controller, UI Builder prompts you to add it.

Not all components work with controllers, but you can easily see which ones do in the component library. If you have a controller configured on your UI Builder page, you can open the component library to view which components have presets. Components with presets available are highlighted in the component library.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Components with presets



Add a controller

Add a data controller to your page to use component presets.

Before you begin

Role required: admin

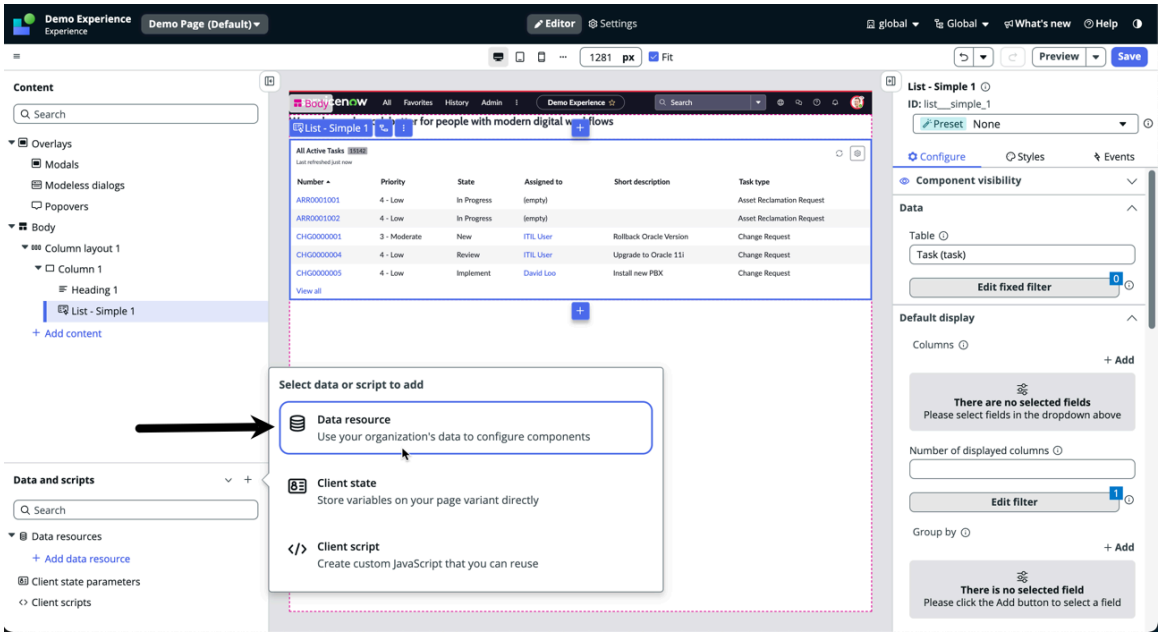
About this task

The record controller is the only controller that you can add to a page in Brazil.

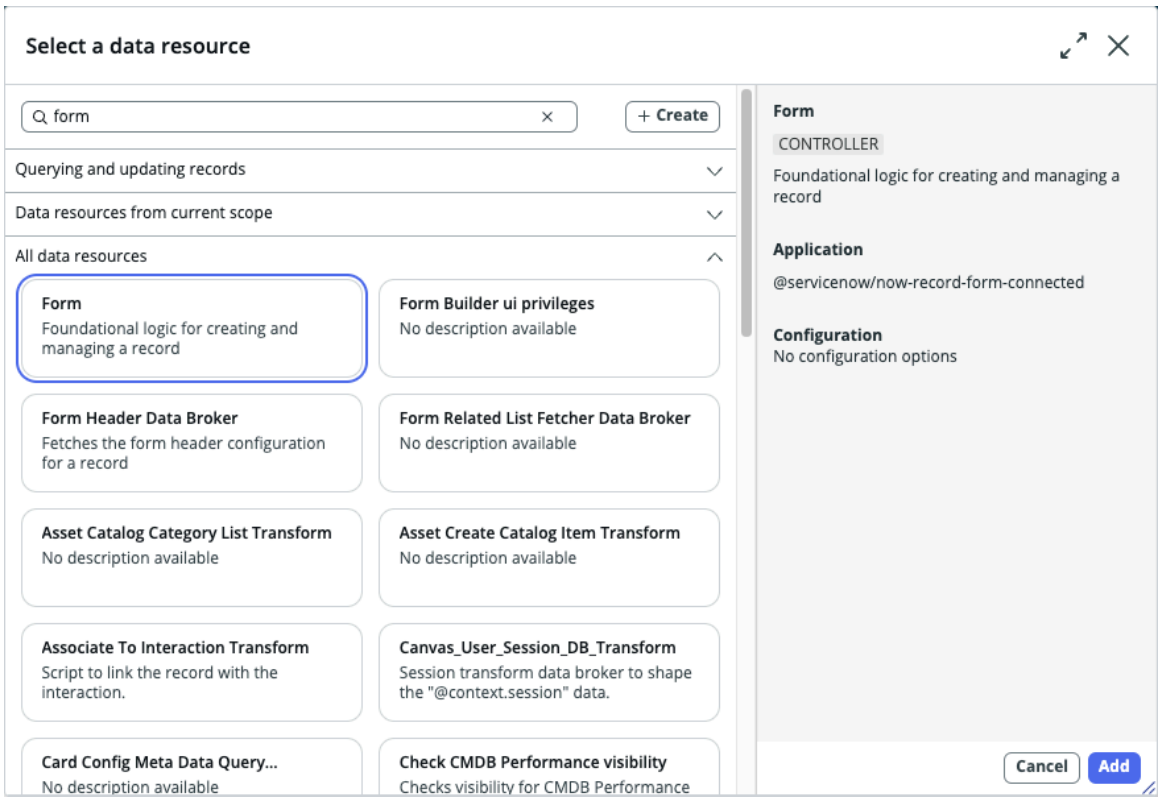
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   For more information, see Configure how users interact with your applications in UI Builder.
3. Create or open a page or page variant.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. Select the **+** icon in the data resource drawer.
5. Select **Data resource**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

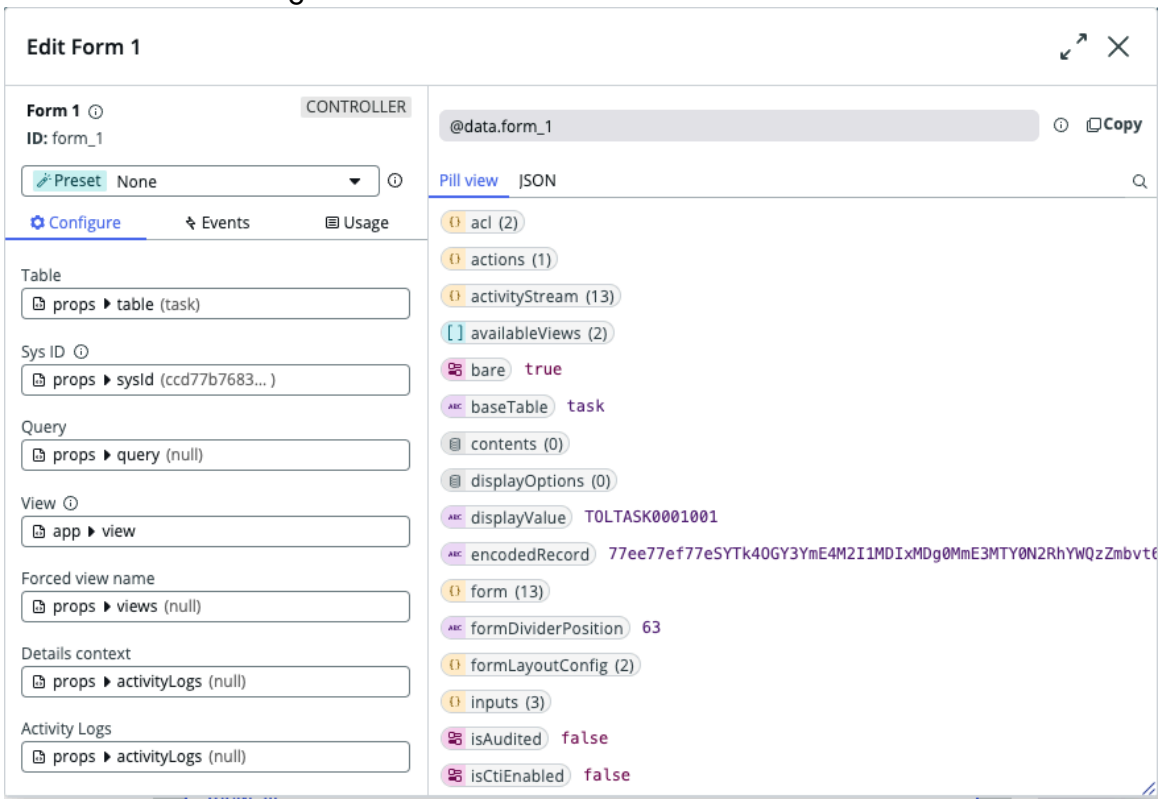


6. Enter `form` in the search field.
7. Select the controller you would like to add to your page.
8. Select **Add**.



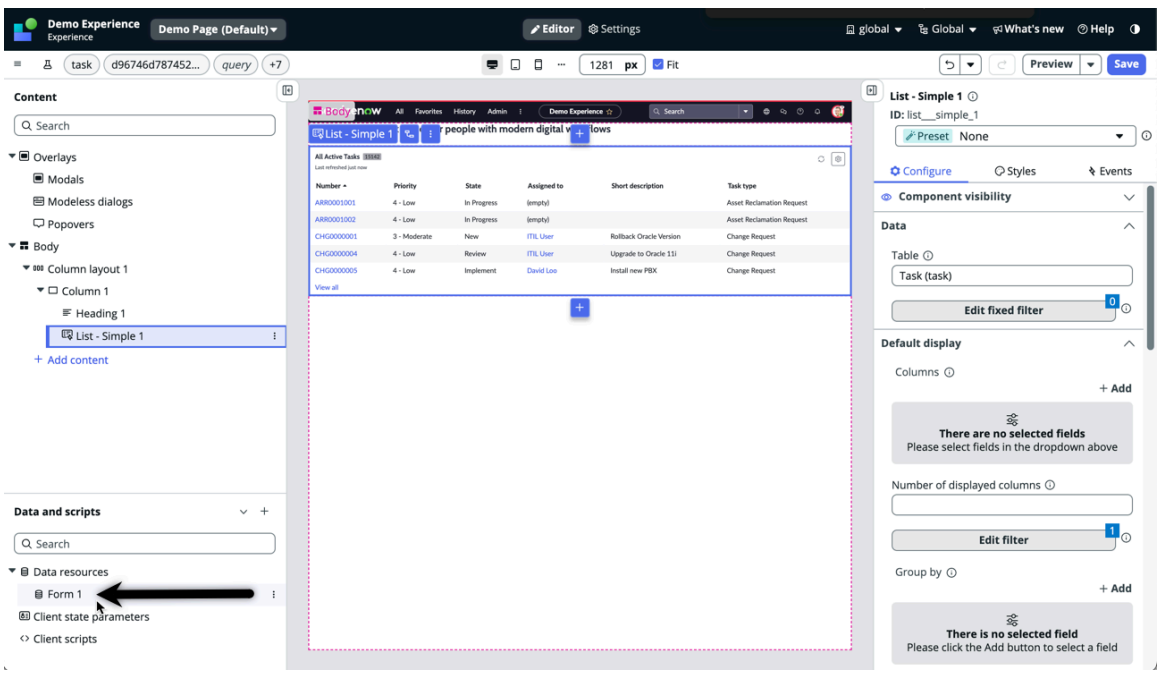
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

9. Fill in the fields to configure the controller.



Result

The form controller displays in the **Data resources** section.



Edit a controller

Configure a controller to pull data from a table.

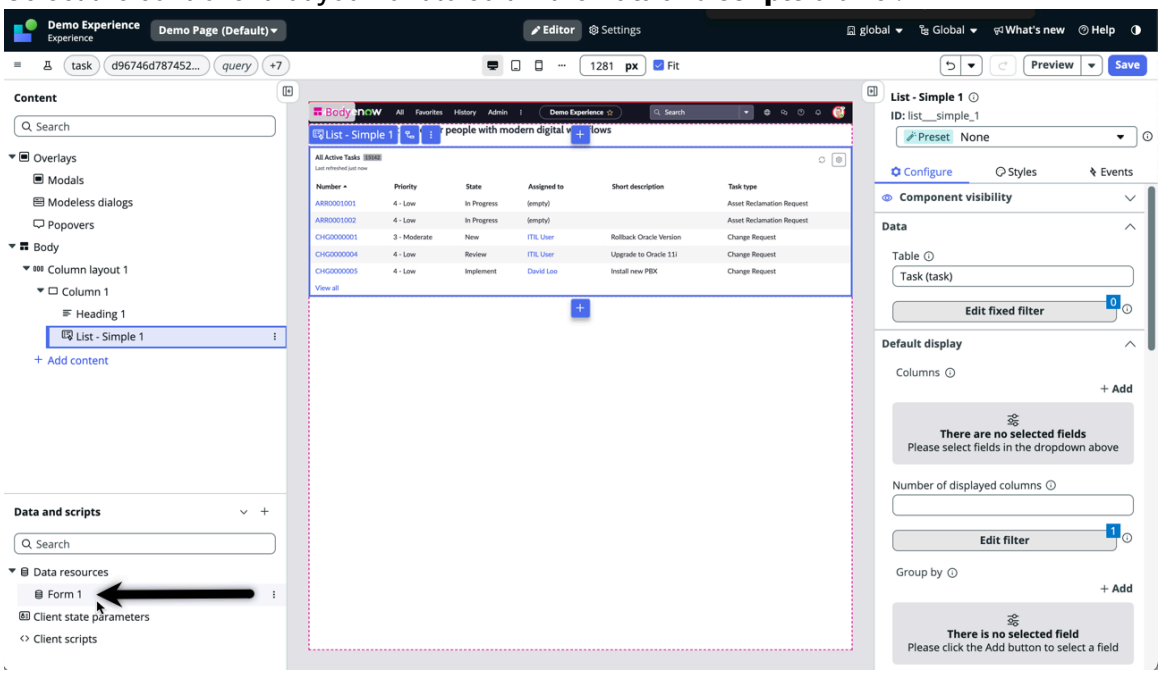
Before you begin

Role required: admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   For more information, see Configure how users interact with your applications in UI Builder.
3. Create or open a page.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. Select the controller that you want to edit in the **Data and scripts** drawer.



5. On the form, fill in the fields.

Form controller form

Field	Description
Type	Controller is predefined as the type for all controllers.
Table	Add a table that you want the controller to pull data from.
Sys ID	Enter the unique identifier for a record, provide a value, or use -1 to generate a new value.

6. Close the edit controller modal.
7. Select **Save**.

Delete a controller

Delete a controller that you no longer need in UI Builder.

Before you begin

Role required: admin

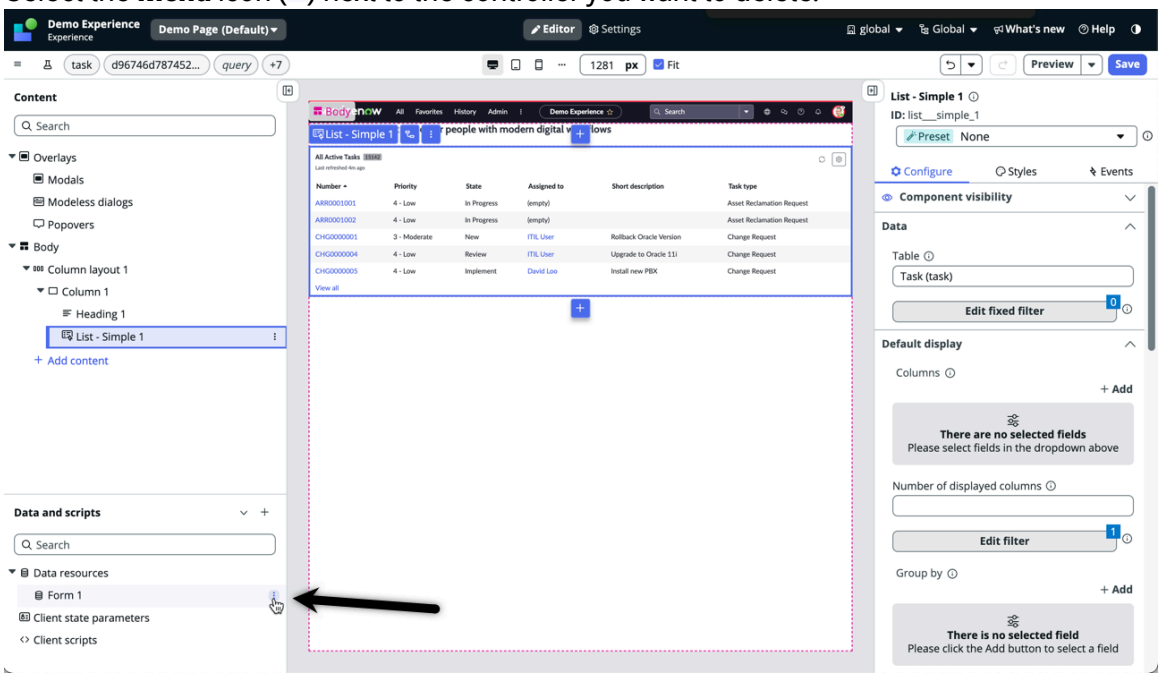
About this task

Controllers cannot be deleted from pages created with a page template.

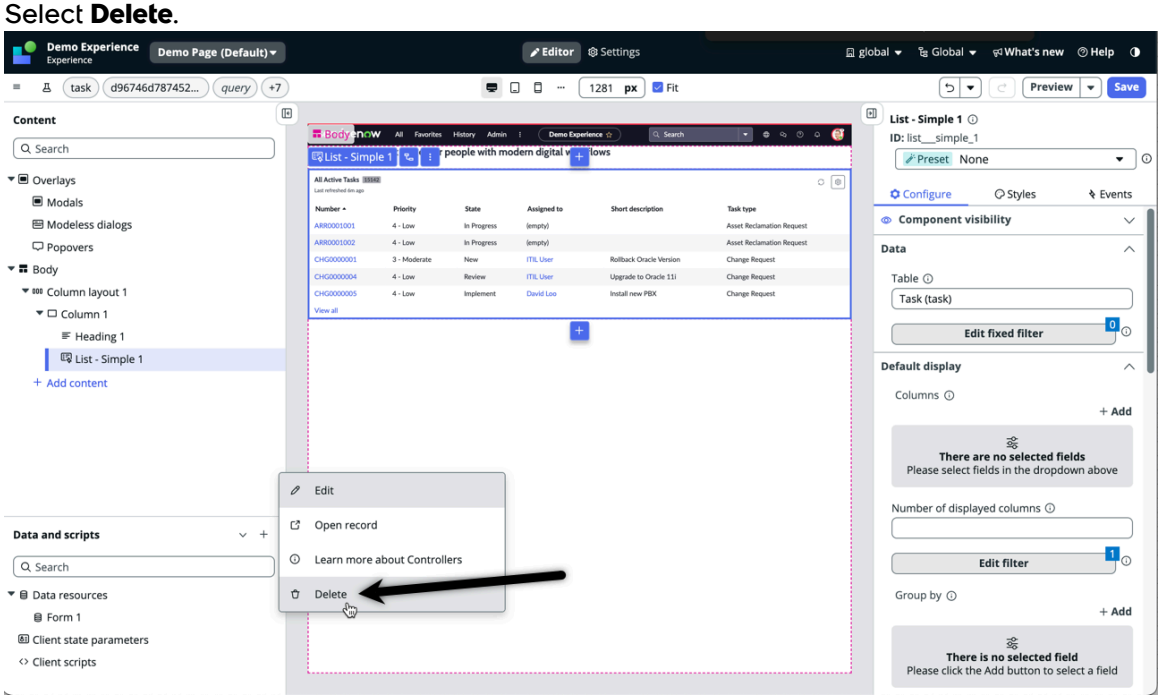
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**. For more information, see Configure how users interact with your applications in UI Builder.
3. Open the page with the controller you want to delete.
4. Select the **Menu** icon (⋮) next to the controller you want to delete.



5. Select **Delete**.



The controller is removed from the local data resource instances.

View properties and events in the Controller API

The controller public API defines the output data that a controller provides to a preset. This includes the property values and handled events used by a component when a preset is selected. Property and event information is available to view in UI Builder.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

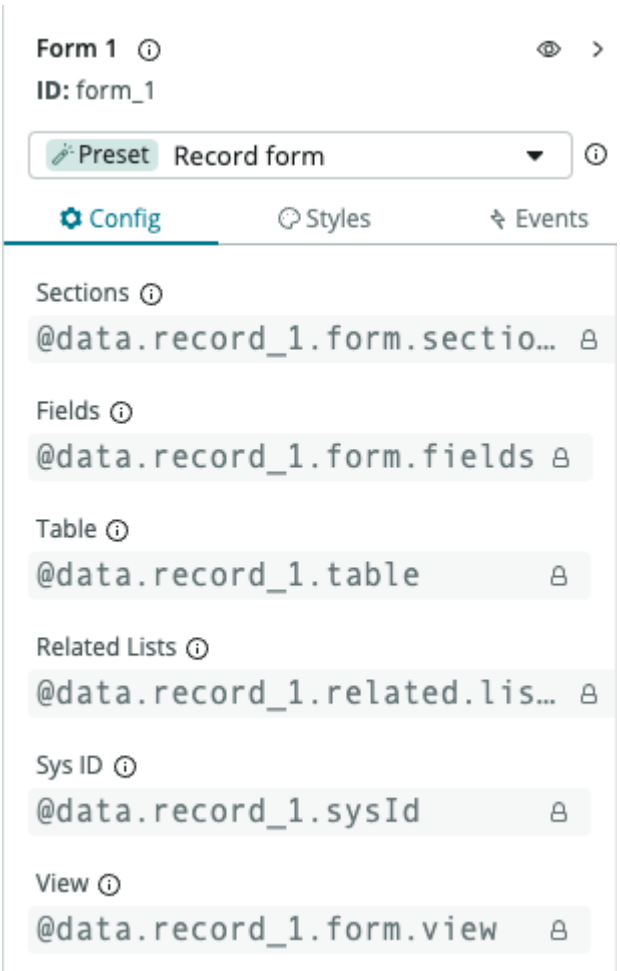
The component property values in a preset can be static values or paths to controller output data. You can use the data resource inspector in UI Builder to display the values from the controller data structure used by a preset. To view the payload carried by an event in the preset, you can look in the event handler picker for that event.

Viewing properties in the Controller API

You can view preset properties in the UIB data inspector. Knowing the available values helps you understand how a component will behave in your design and helps you identify any properties in the preset you might want to override.

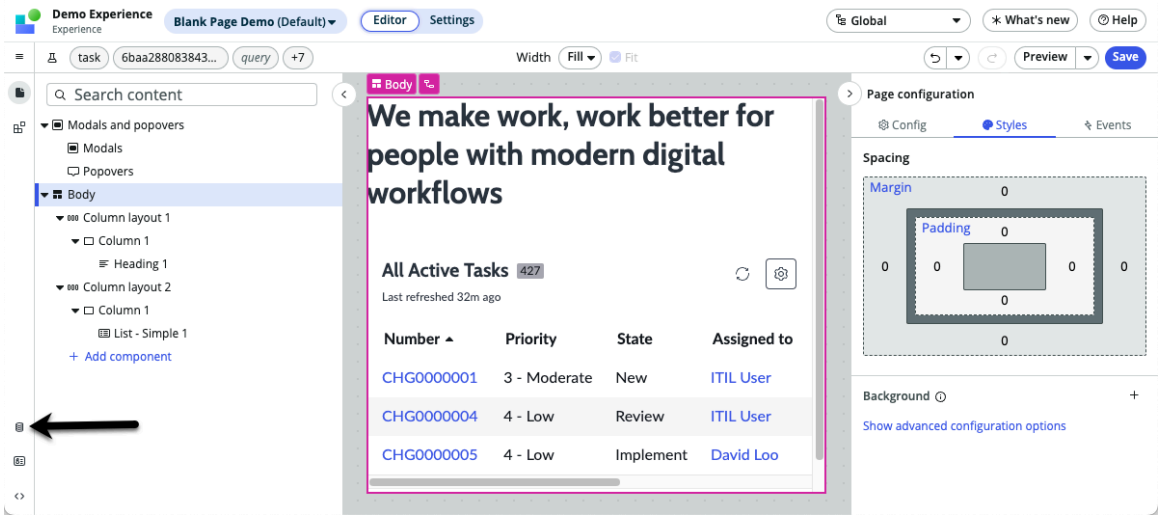
This procedure assumes that you have placed a component with a preset on the page and have configured a controller. For instructions, see Add a controller.

The configuration tab displays preset property values as a path to the controller output. The base data path is expressed as `@data.<controller_name>`. The remainder of the path is built using the contents of the categories within the controller data hierarchy. You can use this path to view the current values for the record for which the controller has been configured.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

1. To view the data resources for a component, select the data icon in the lower left sidebar.



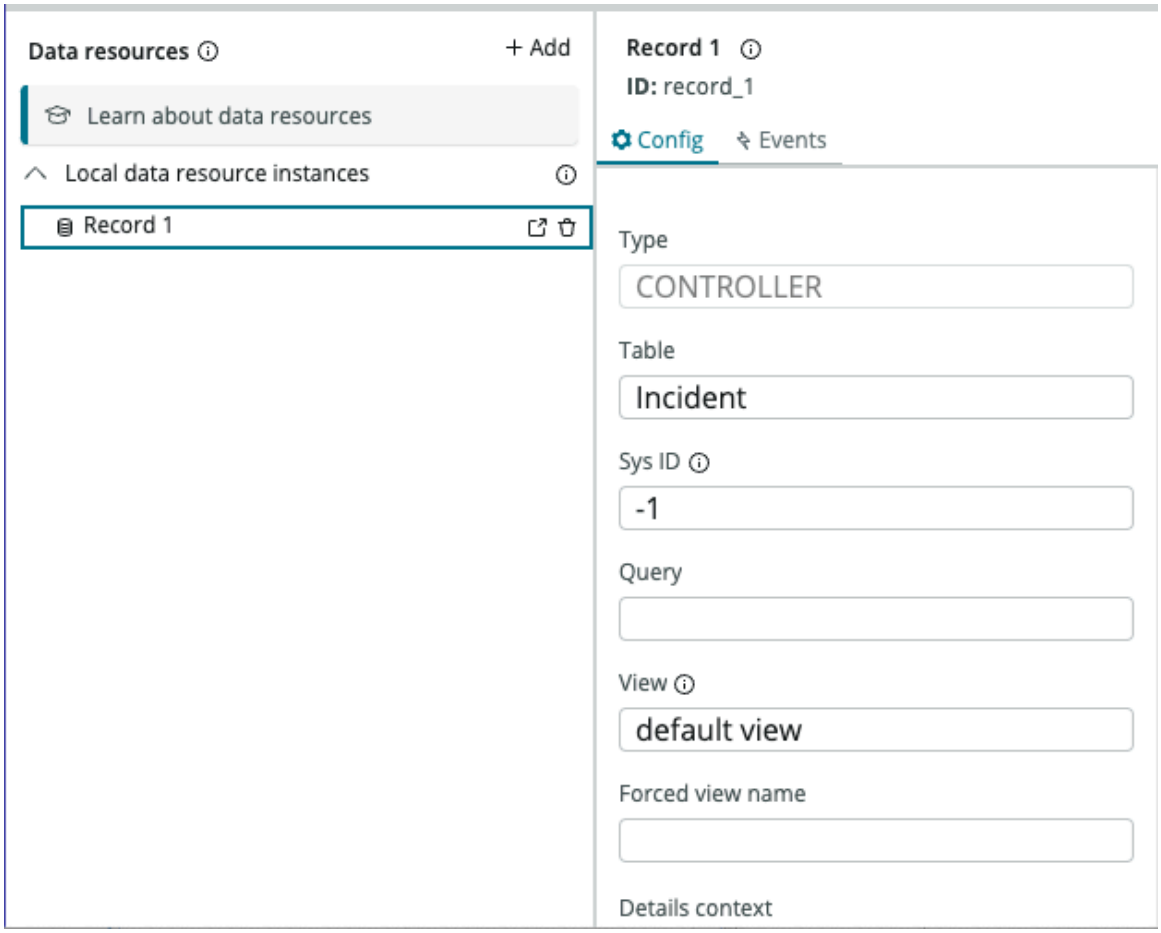
The three column UIB data inspector appears.

2. In the first column where the data resources are listed, select the controller whose output you want to view.

   The configuration data for the selected controller appears in the Config tab of the second column. These are the input properties you entered when you configured the controller for the first component with a preset you placed on the page. You can edit these connection values here if you want.

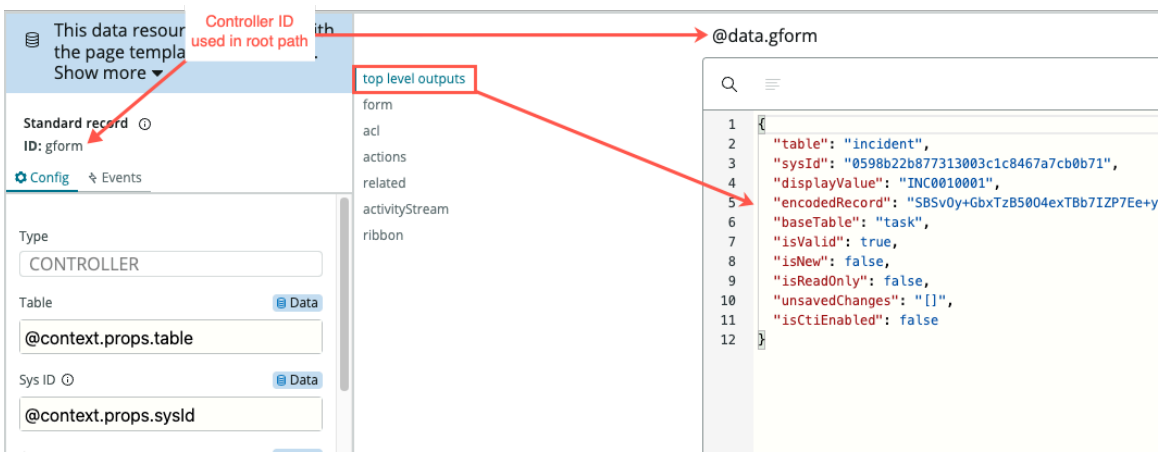
   Note: Configuring **-1** for the value in the **Sys ID** field configures the controller for a new record. The controller then generates a full GUID that you can use to store data against the record before it's saved. This allows you to perform actions in a newly created record such as saving attachments.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



3. Select **top level outputs** in the third column.

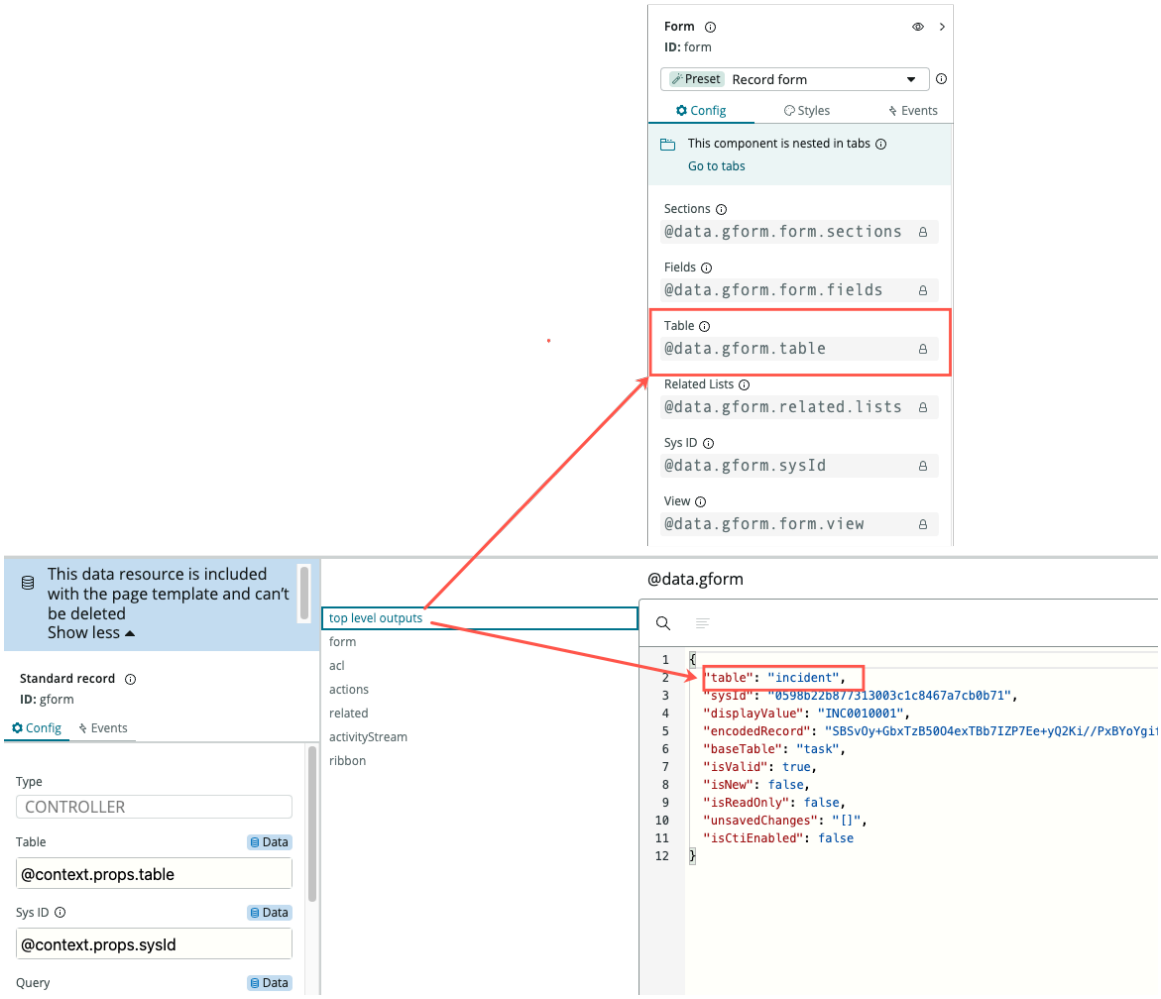
   This is the parent level of the controller data hierarchy, indicated with the prefix `@data.`, followed by the controller ID. This combination forms the root path to the data. The data contained in the top level outputs are displayed in the adjacent code field and are available for presets and scripts to use. Other properties at that level are some common properties that apply to all records.



   When the preset maps a component property to controller output, the path to the output is shown instead of the value. Paths are denoted as **@data.<controller id>.<path from top level to output property>** and can be used as the values or as part of formula expressions.

   In this example, the mapping in the preset results in a path to `@data.gform.table`.

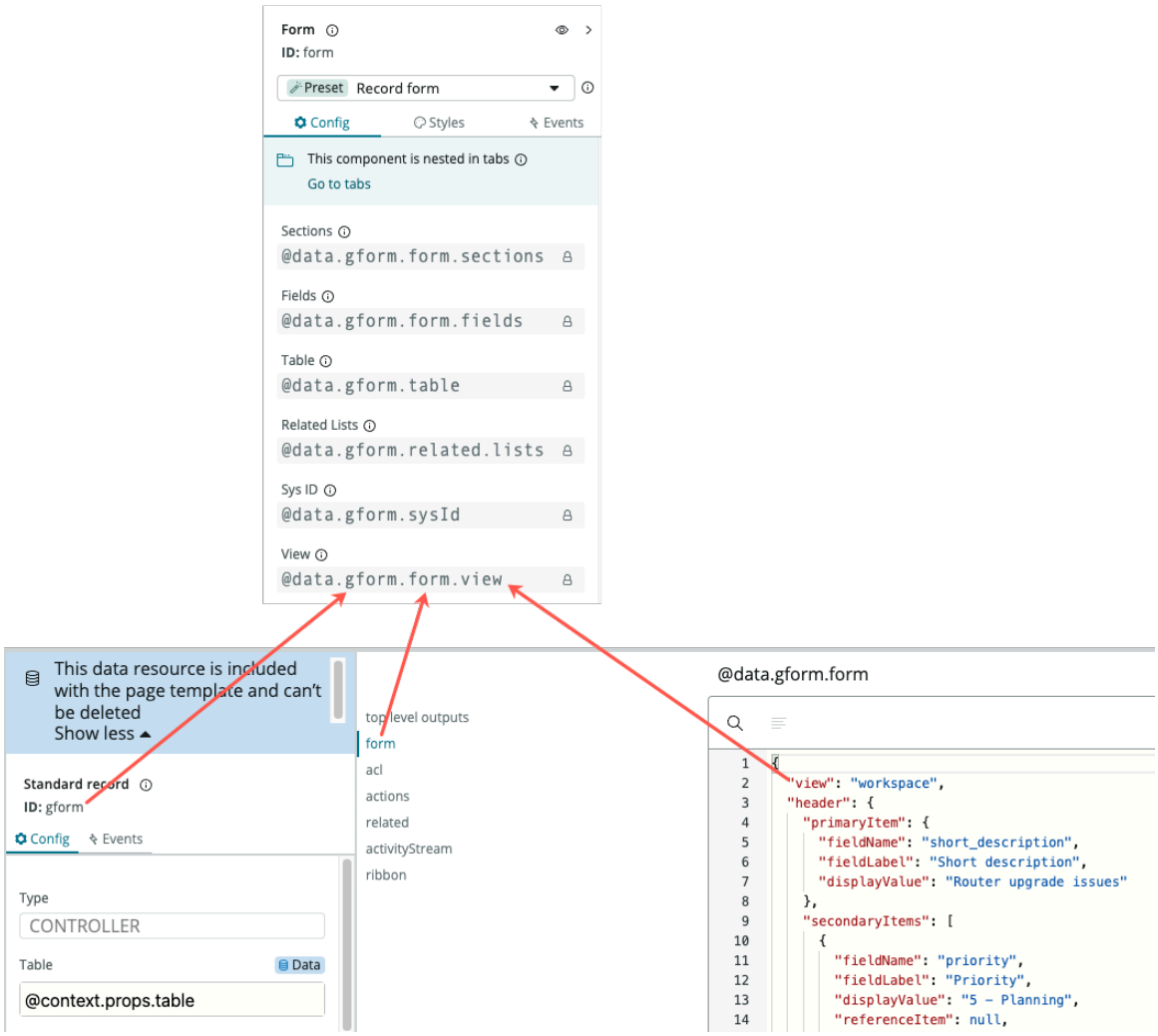
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



4. To view data below the top level, select the child category indicated by the path in the preset property value.

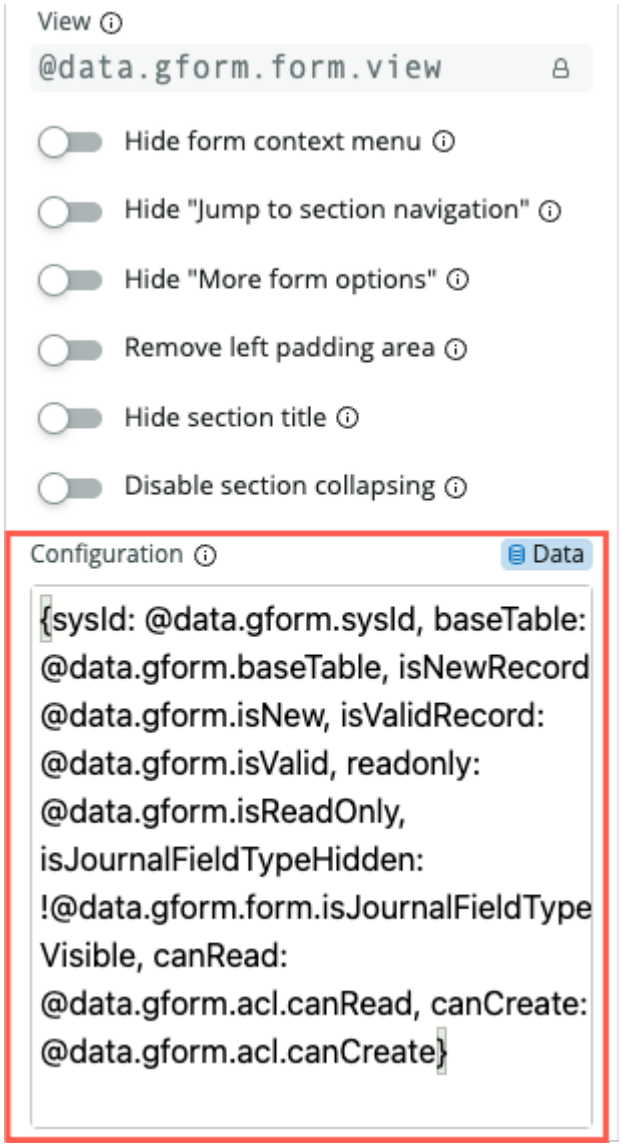
   The name of the child category appears in the path as `@data.<controller ID>.<category>`. The data from that category is then appended to that path. The root path above the code field indicates the child category being used. In this example, the path to the **View** property in the `form` category is expressed as `@data.gform.form.view`. That preset value displays forms in the **workspace** view.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



In this example, multiple controller output properties are used in a formula expression to build an object.

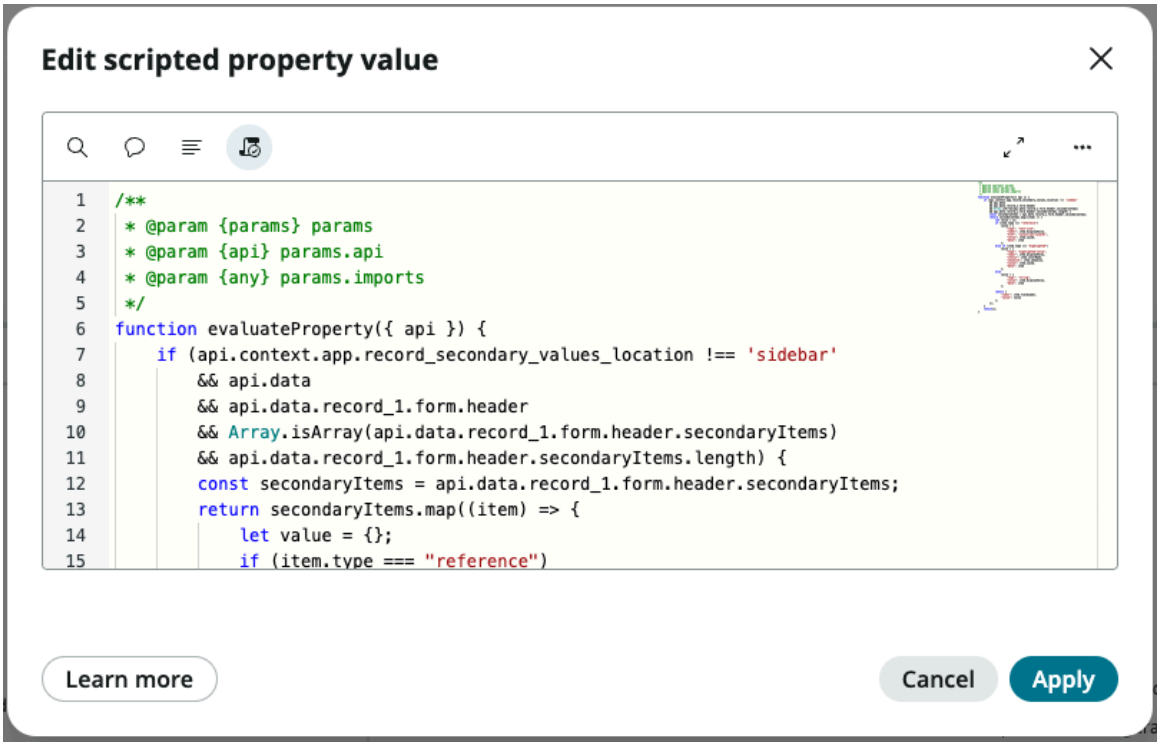
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



5. To view a scripted property value, select the lock icon next to the property.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

This puts the property into edit mode and opens a modal for editing the script.



Viewing events in the Controller API

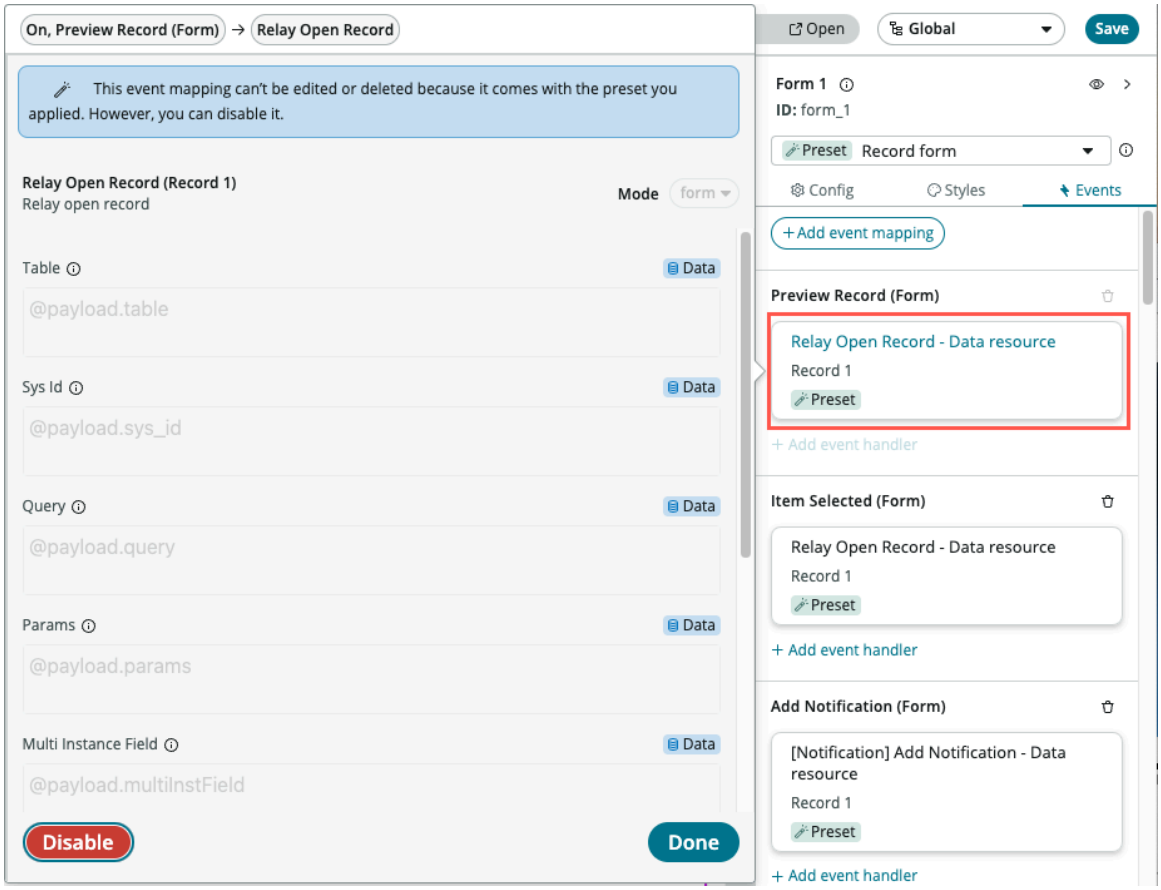
You can view handled events mapped to a component by a preset and their payloads in the Events tab in the UI Builder configuration panel. If the data mapping for an event isn't appropriate for your use case, you can add additional data handlers.

Note: Data handler mappings provided with the preset are not editable.

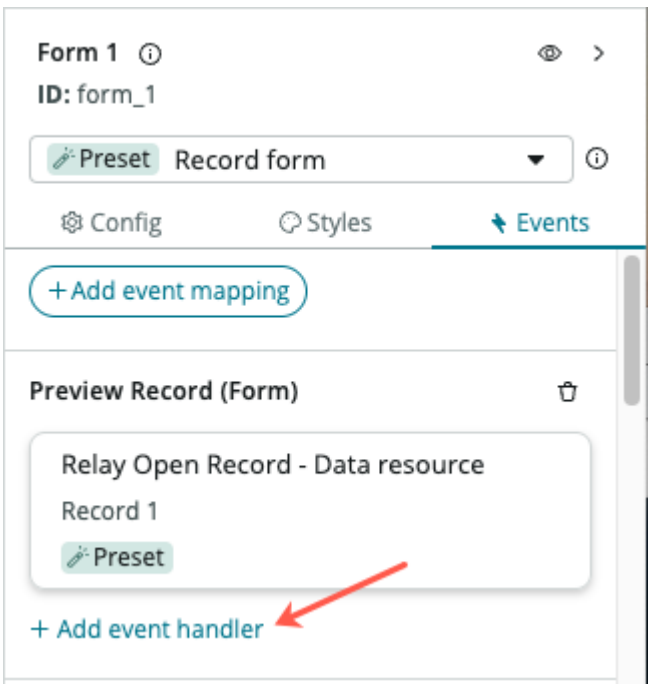
1. Select the Events tab in UI Builder.
2. To view the mapping for a preset event, select the event tile.

   A modal appears showing the payload properties for the preset event. You can **Disable** the event in this modal.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



3. To add an event handler to the event, select **+ Add event handler** under the event tile.

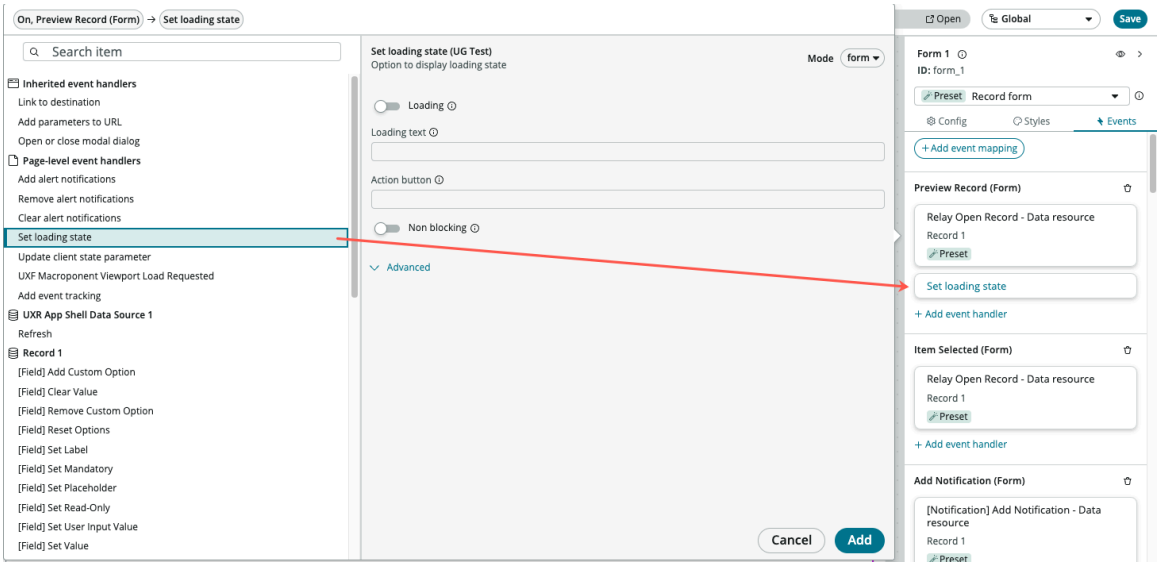


A modal appears, showing a list of available, handled events. You can select any action from the list, including an event handler from one of the controllers listed.

4. Select an event handler from the list and select **Add**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The new handler is listed under the event tile.



Create custom controllers (Advanced feature)

Build custom controllers to use across experiences. Easily connect reusable data, scripts, and events to your page and set up inputs, outputs, and events.

Before you begin

Role required: ui_builder_admin

Procedure

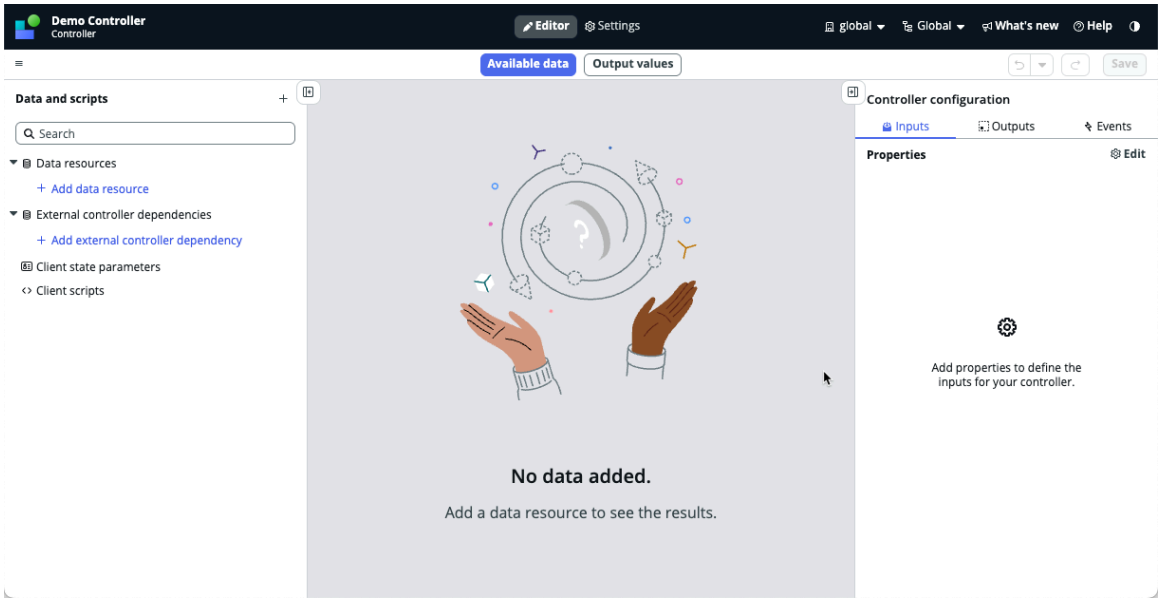
1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Select **Create**.
3. Select **Controller**.
4. On the form, fill in the fields.

Create a page collection form

Field	Description
Name	Name to track your controller internally.
Description	Short description to help find your controller. Write a description that helps page builders understand what content is included in the controller.

5. Select **Create**.
   The controller building window opens.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



6. Add data resources.
   a. Select **+ Add data resource**.
   b. Select a data resource from the list.
   c. Select **Add**.
   d. Configure the data resource.
   e. Select **Save**.
7. **Optional:** Add an external controller dependencies.
   a. Select **+ Add external controller dependency**.
   b. On the form, fill in the fields.

Edit external controller dependency

Field	Description
Name	Name to identify your external controller dependency.
Label	Label to help find your external controller dependency.
Controller	Select a controller for your external controller dependency.

   c. Select **Save**.
8. **Optional:** Add client state parameters.
   a. Select **Client sate parameters**.
   b. Select **+ Add**.
   c. Enter a unique name for the client state parameter.
   d. Select the type from the dropdown.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

e. Enter an initial value in the field.

f. Select **Save**.
For more information about client state parameters, see Client state parameters.

9. **Optional:** Add client scripts.

a. Select the **+** icon next to **Client scripts**.

b. Write your script to perform an action.

c. **Optional:** Add a **Script include** or **Associated components**, which shows up in the `imports` parameter of your client script function.

d. Select **Apply**.

e. Select **Save**.

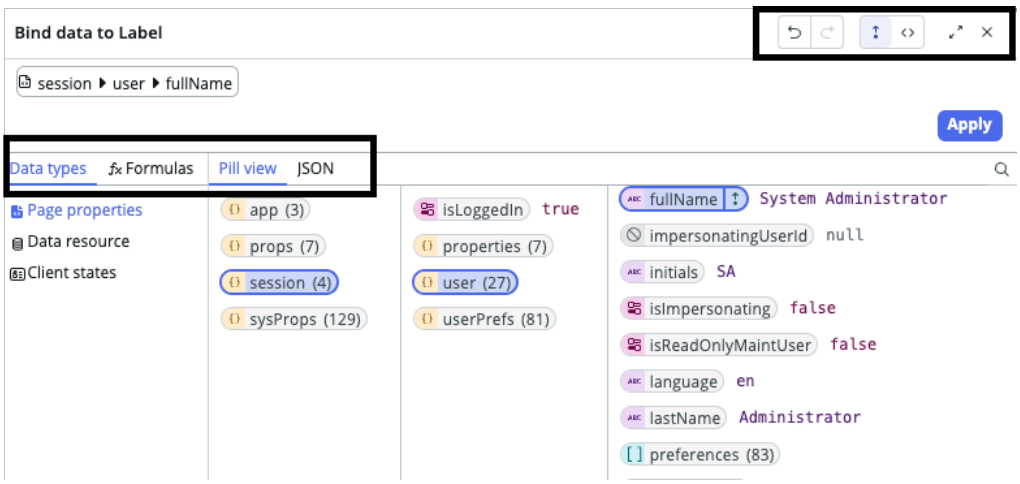
What to do next

Use the custom controller you just created by adding the controller to a page.

Connect data components

Use the data binding modal to associate data exposed by local data resources to components on your UI Builder page.

Data binding modal



Data binding modal options

Option	Description
Undo	Undo the previous change.
Redo	Redo a change that has been undone.
Bind data using drag and drop	Show Pill view.
Use script	Show JSON view.
Toggle full screen	Expand the data binding modal to full screen size. Select again to reduce the data binding modal to smaller size.
Close	Close the data binding modal. If you have unsaved changes, you are given the option to save.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

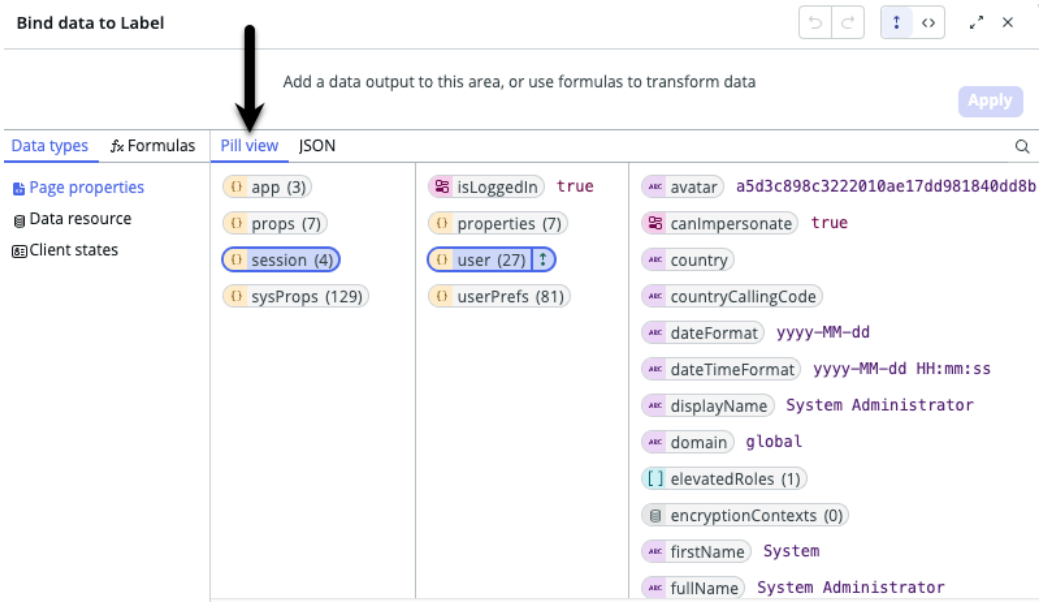
Data binding modal options (continued)

Option	Description
Data types	Lists page properties, data resources, and client states.
Formulas	Lists formulas that can be used. A search field is available.
Pill view	When **Data types** is selected, pill view enables you to specify a property value or display value. You may have to select several pills to get to a specific value.
JSON	When **Data types** is selected, the JSON option enables you to edit the raw JSON code.

There are two methods you can use for binding data: drag and drop, and editing JSON.

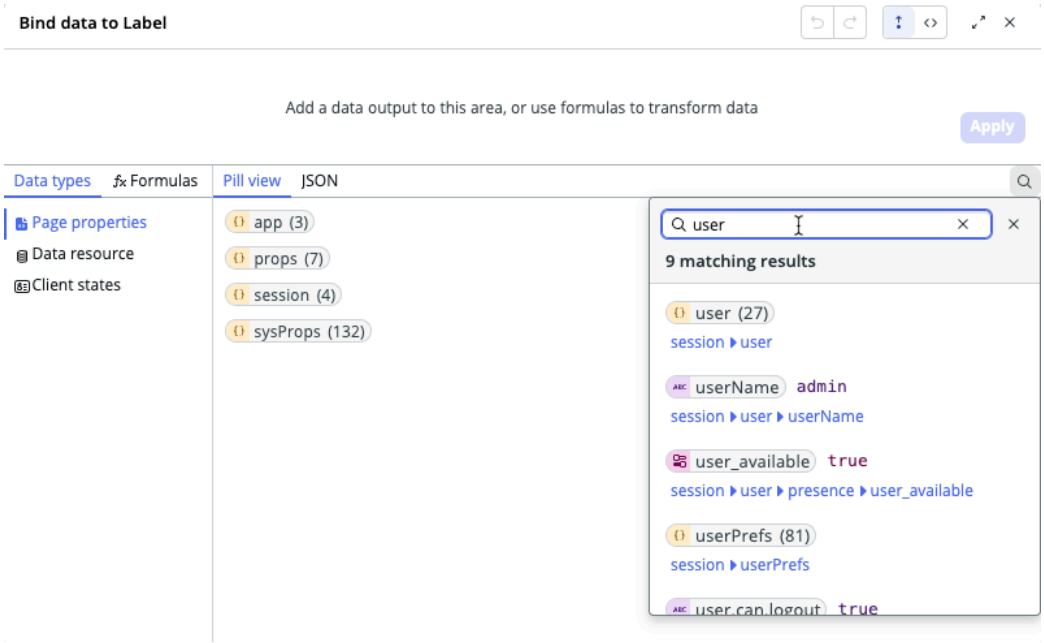
Binding data with drag and drop

Use this method if you want to use a visual process.



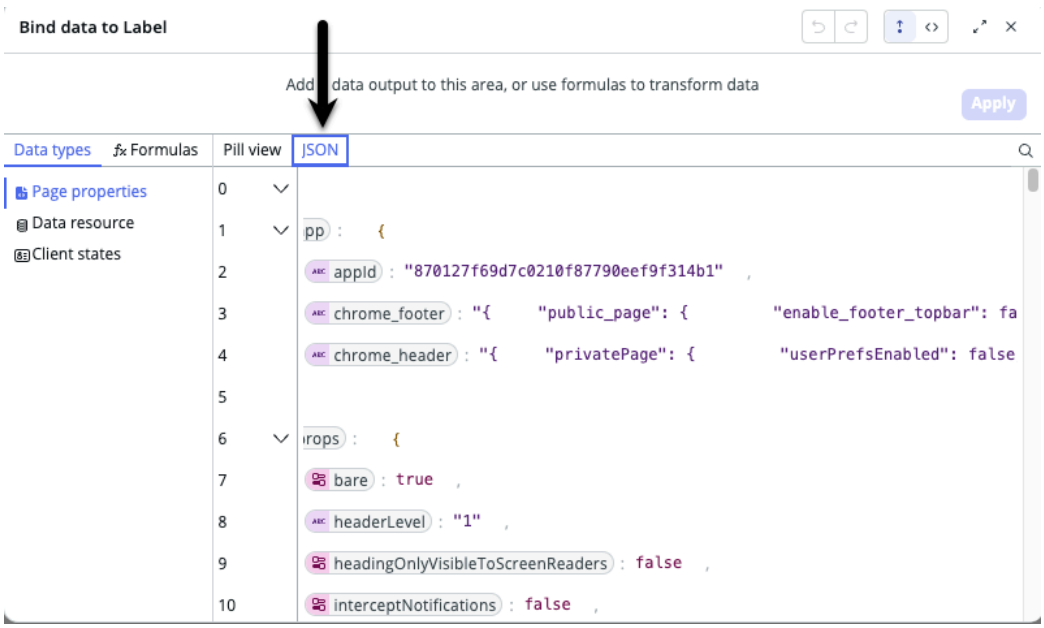
Use the search bar to find the data you would like to bind to your component.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Binding data by editing JSON

Use this method if you are comfortable working in and editing JSON code.



Connect data to your components

Bind data exposed by local data resources to components on your UI Builder page.

Before you begin

Role required: ui_builder_admin

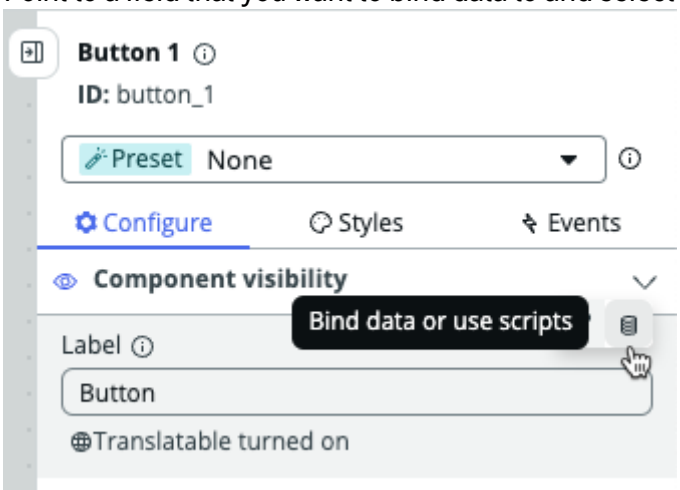
About this task

Data binding is the process of associating data with a UI element that displays the information. Before binding data to components you must add a data resource to your UI Builder page, see Add and configure data resources to a page for more information.

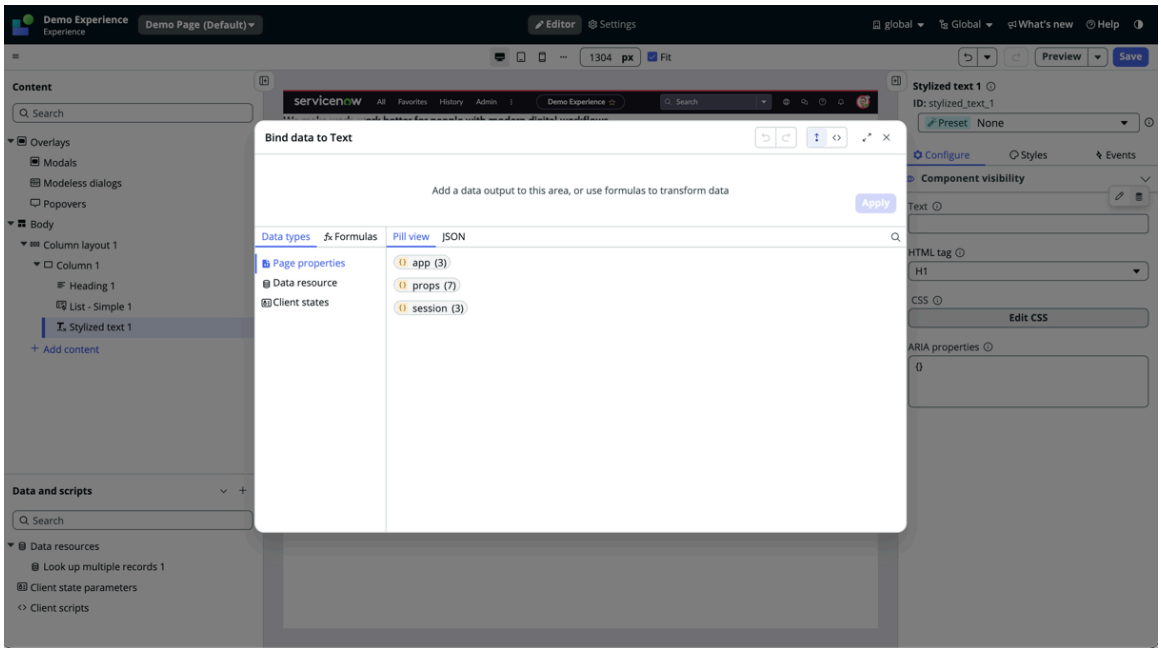
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create or open a page.
4. Add a data resource to your page.
   For more information, see Add and configure data resources to a page.
5. Add a component to your page.
   You need a component on your page before you can bind a data resource to it. For more information, see Customize UI Builder pages using components.
6. Select the **Configure** tab from the configuration panel in UI Builder.
7. Point to a field that you want to bind data to and select the dynamic data binding icon ( ).

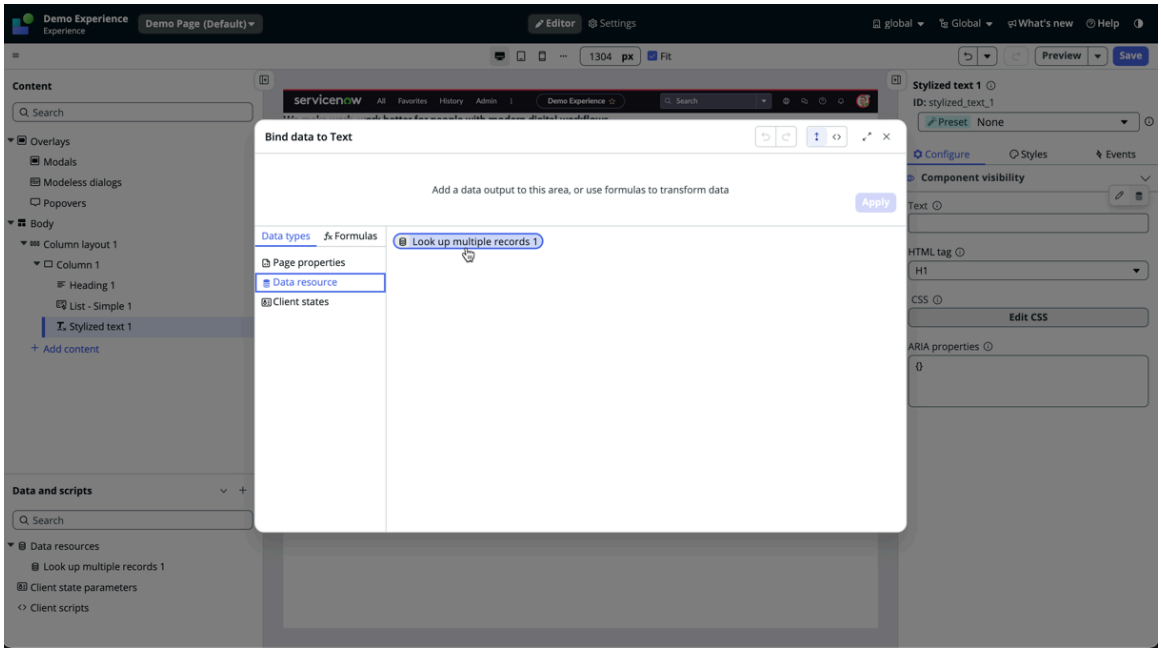


The data binding modal appears.



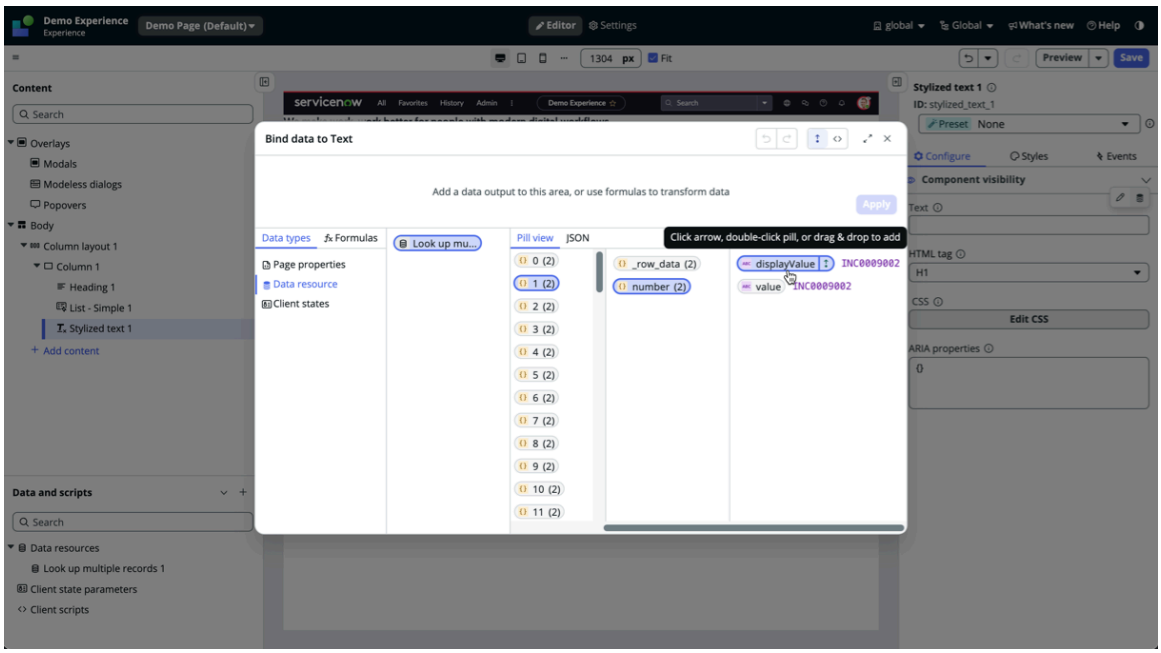
8. Select **Data resource** in the **Data types** tab.
9. Select the data resource that you want to bind to the component.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



10. Select and drag the data you want to bind to the component into the field above.

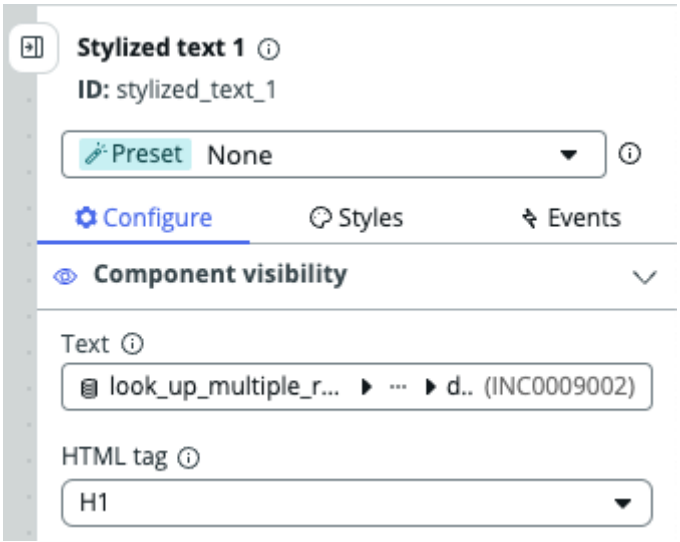
You may have to select several pills to get to a specific property value or display value.



11. Select **Apply**.

The component displays the property value or display value that you've selected.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



12. Select **Save** in the UI Builder header.

Connect data to your components with formulas

Bind data exposed by local data resources to components with formulas on your UI Builder page.

Before you begin

Role required: ui_builder_admin

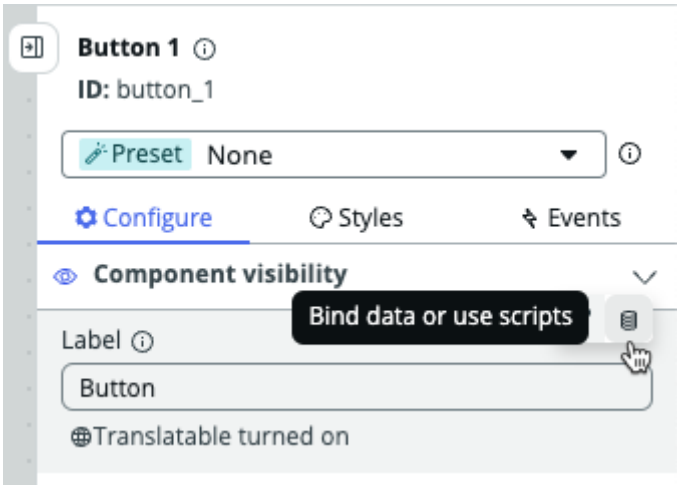
About this task

Data binding is the process of associating data with a UI element that displays the information. Before binding data to components you must add a data resource to your UI Builder page, see Add and configure data resources to a page for more information.

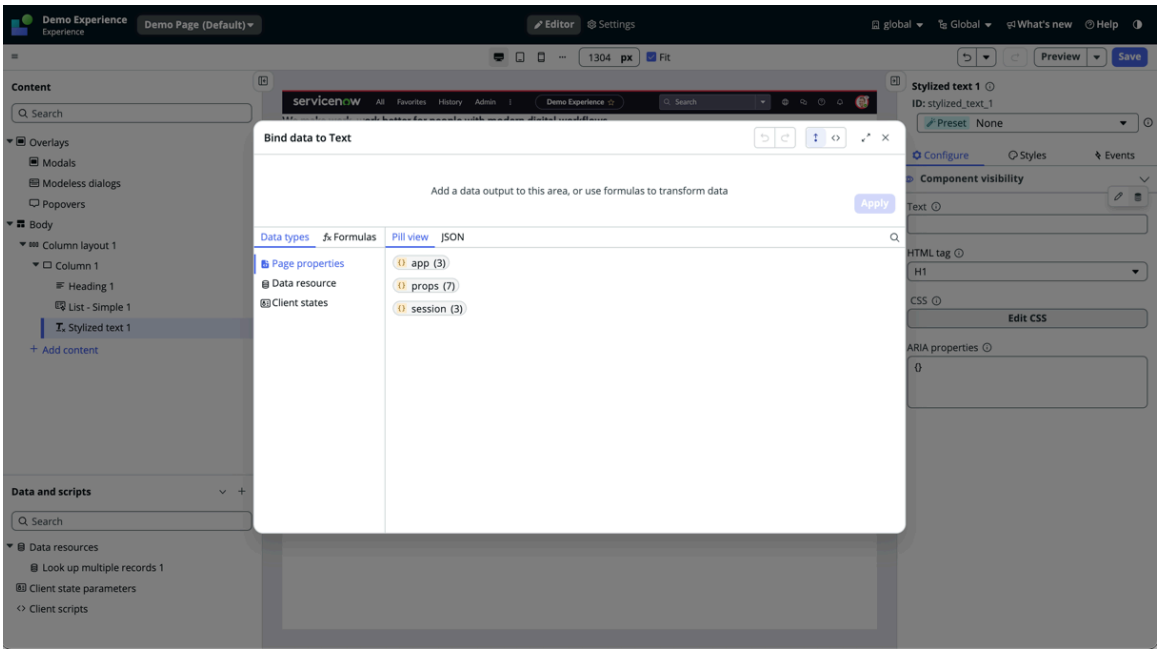
Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create or open a page.
4. Add a data resource to your page.
   For more information, see Add and configure data resources to a page.
5. Add a component to your page.
   You need a component on your page before you can bind a data resource to it. For more information, see Customize UI Builder pages using components.
6. Select the **Configure** tab from the configuration panel in UI Builder.
7. Point to a field that you want to bind data to and select the dynamic data binding icon ( ).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

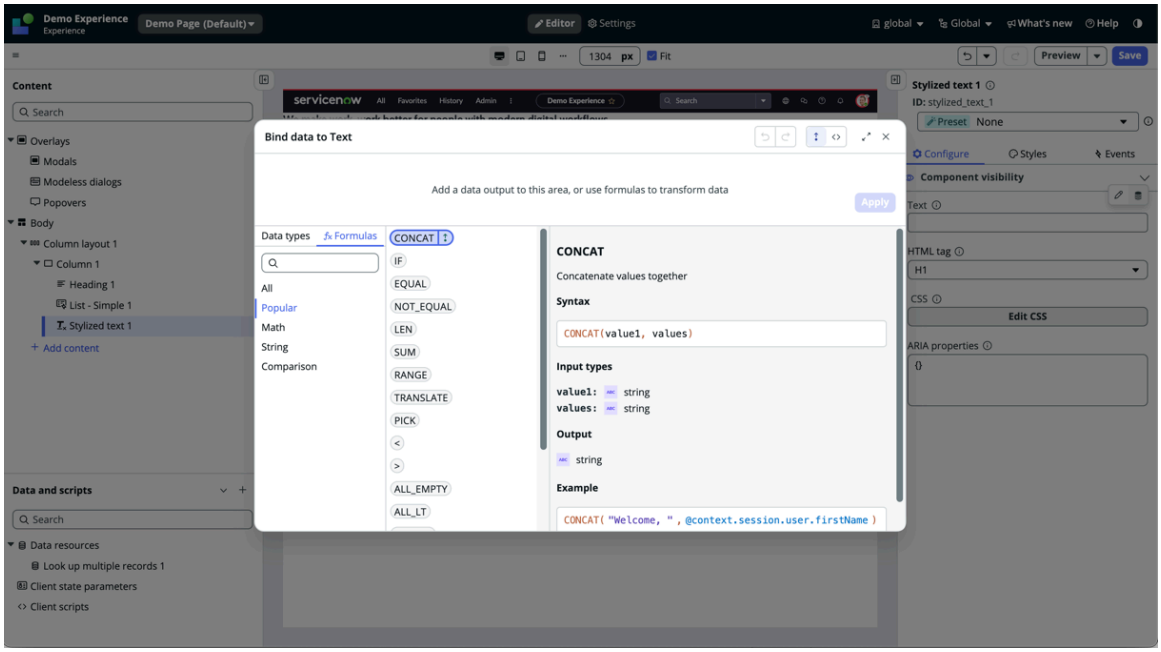


The data binding modal appears.



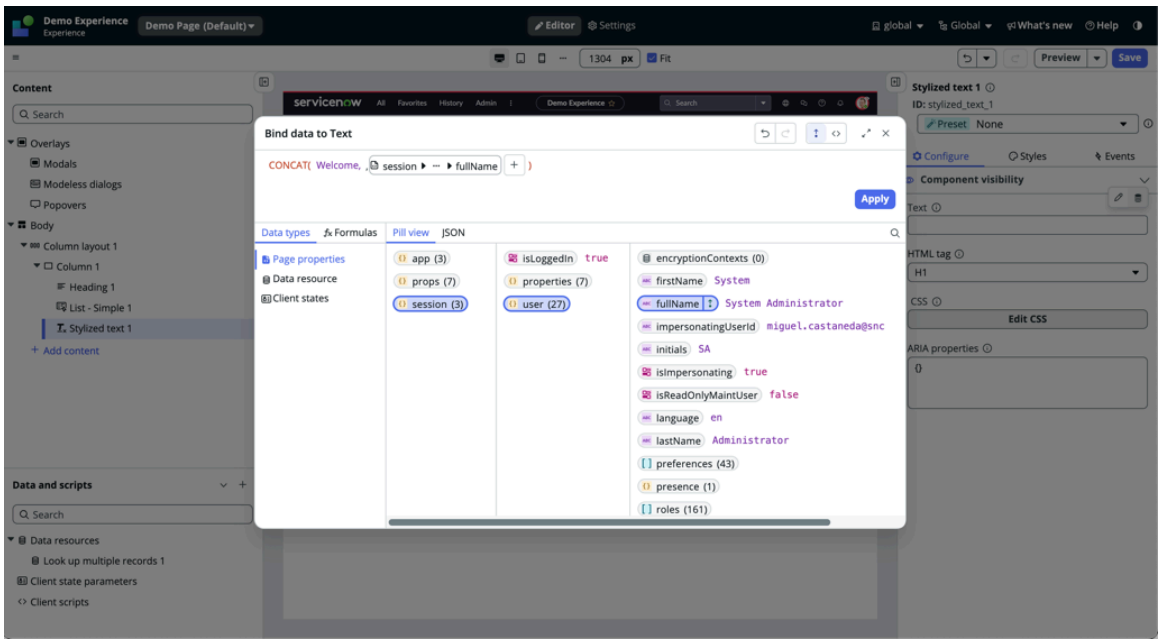
8. Select the **Formulas** tab.
9. Select and drag the formula you want to bind to the component into the field above.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



For more information about formulas, see Supported functions in the UI Builder component formula editor.

10. Fill in the fields of the formula.



11. Select **Apply**.

    The component displays the property value or display value that you've selected.

12. Select **Save** in the UI Builder header.

Client state parameters

Use client state parameters to bind values to component configurations. When the client state parameter's value changes, the component updates to use the new value.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

What is a client-side interaction

In web-based applications, anything that happens in the browser is client-side. Client-side interactions occur when users interact with experience components by clicking. Examples of client-side interactions include:

- Clicking a button
- Applying a filter
- Sorting a list
- Refreshing a list
- Providing inline search results
- Alerting a user to incomplete or incorrect data

Users sometimes interact with pages in order to navigate to other pages. In other cases, users want to make updates to a page's content, appearance, or data. Rather than refreshing the entire page, client-side interactions update only the affected parts of a page.

For example, a user can sort a list by clicking a column header. Sorting redraws only the list and not the entire page.

Creating client-side interactions in UI Builder involves:

- Client state parameters
- Events
- Client scripts

What are client state parameters

Client state parameters are page variables. Define and configure a client state parameter and use the value to configure components. Client state parameters centralize management of configuration values for page components. UI Builder uses client state parameters to enhance user experience and provide personalized content and services.

Example of client state parameters

In the example, a page has a client state parameter called color. Two of the three components are configured to use the color client state parameter. If color is set to orange, the components are orange. If color is set to purple, the components are purple. If color is set to green, the components are green. The client state parameter is a single place to manage a common value for the page. Without the client state parameter, each component that uses a value would need to be updated individually if that value changed.

For example, a web experience stores the primary color used by components in the color client state parameter. When the components are configured to use the client state parameter, changing the value of the client state parameter updates the components to the new value.

Buttons can be added to the experience to allow users to select a color for the page components. With a single click, a user can simultaneously change the color of all components on a page. Store the user’s color choice in a client state parameter, then use the client state parameter to configure the page’s components. When a user interaction causes the client state parameter value to change, the page’s components are updated in real time.

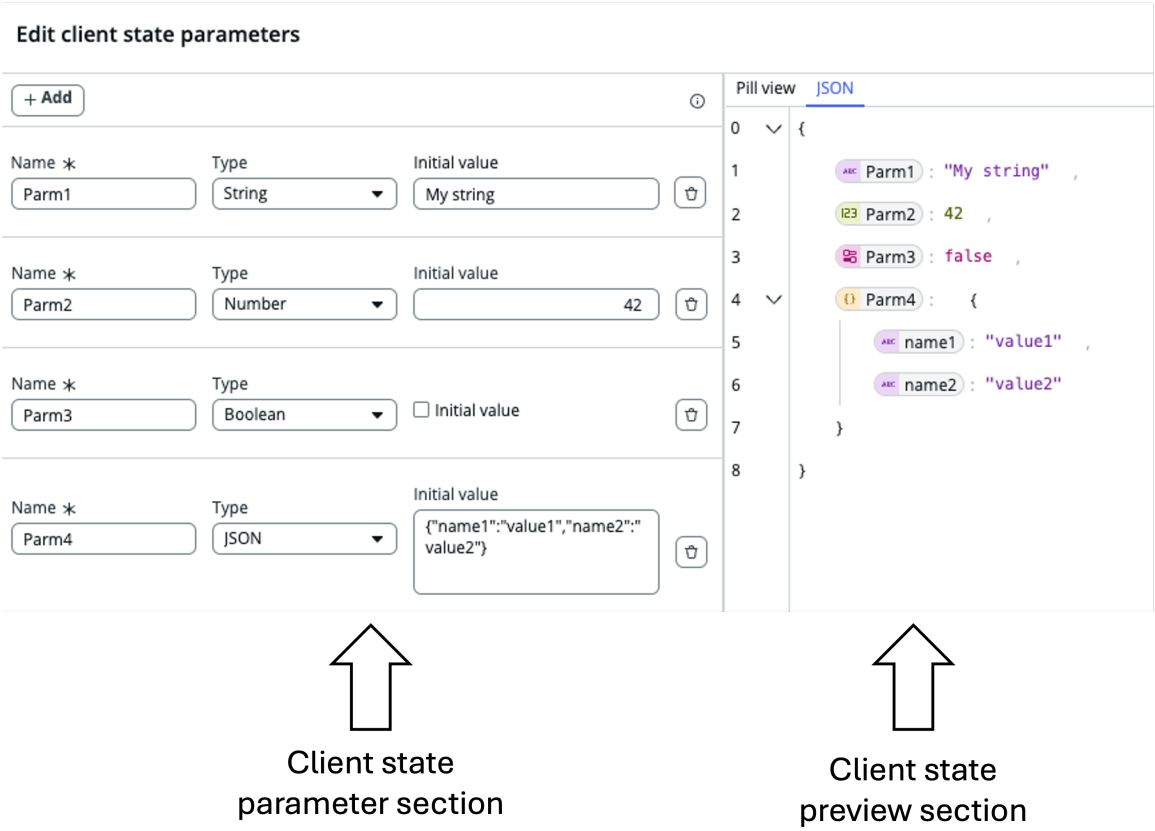
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The client state panel

The client state panel is collapsed by default. Click the client state icon in the left navigation bar to open the client state panel.

The client state panel contains two sections:

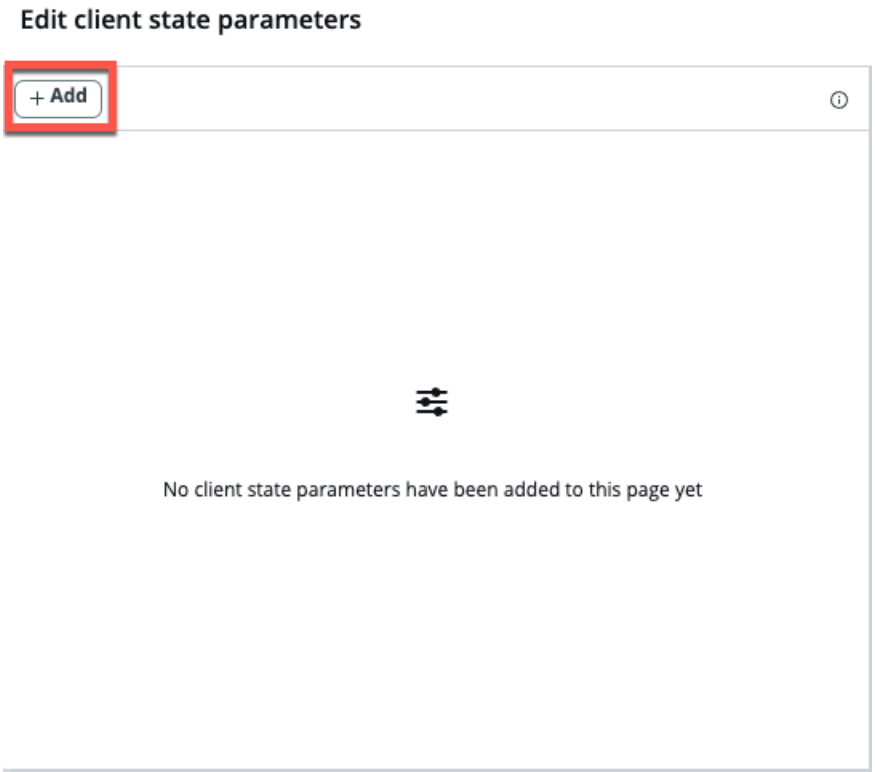
- Client state parameters: The client state parameters for the page
- Client state preview: The JSON for the page's client state parameters



Creating client state parameters

To add a client state parameter to a page, click the **+ Add** button in the client state parameters section.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

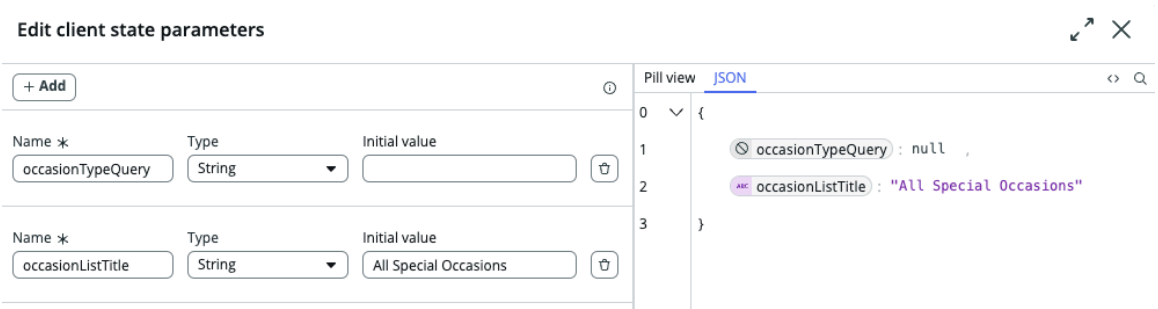


Client state parameters have three configuration fields.

- Name: The name of the parameter. Names should not include spaces
- Type: The parameter's type
  - String
  - Number
  - Boolean
  - JSON
- Initial value: The default value for the parameter

In the example, the client state panel has two client state parameters.

- `occasionTypeQuery` is a string with no default value
- `occasionListTitle` is a string with a default value of `All Special Occasions`



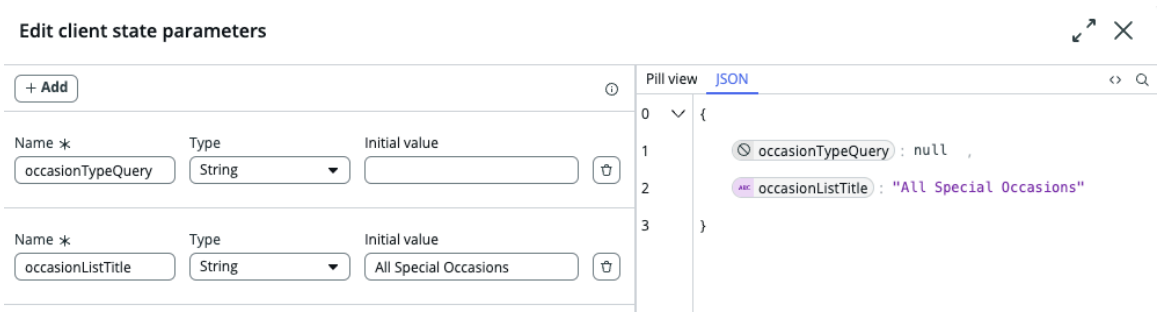
Working with client state parameters

Once you have a client state parameter, what can you do with it? To work with client state parameters, first bind the value of the client state parameter to component configurations. When

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

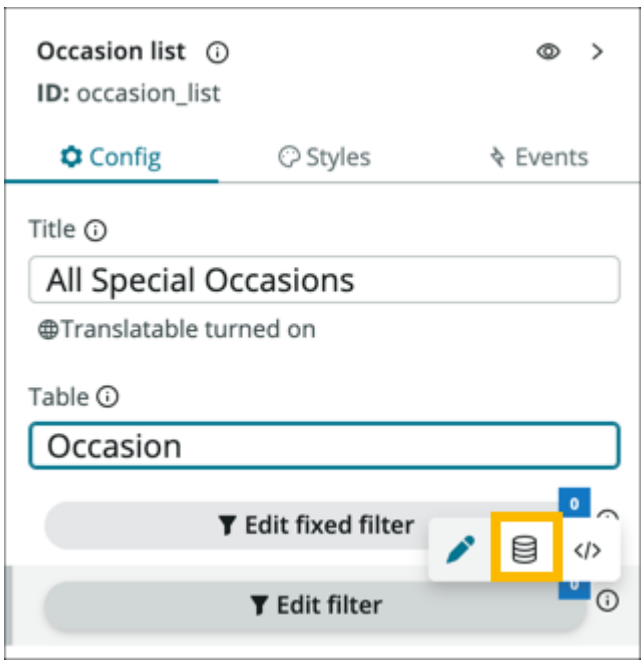
the client state parameter's value changes, the component updates to use the new value. One way to change a client state parameter's value is with event handlers.

In the creating client state parameters page example, two client state parameters were added to the special occasions application: `occasionTypeQuery` and `occasionListTitle`. The `occasionTypeQuery` parameter has no value by default and `occasionListTitle` has the default value `All Special Occasions`. These client state parameters will be applied to the Occasion List component to set the list's title and filter. A button will be used to update the values of these client state parameters to dynamically update the list.



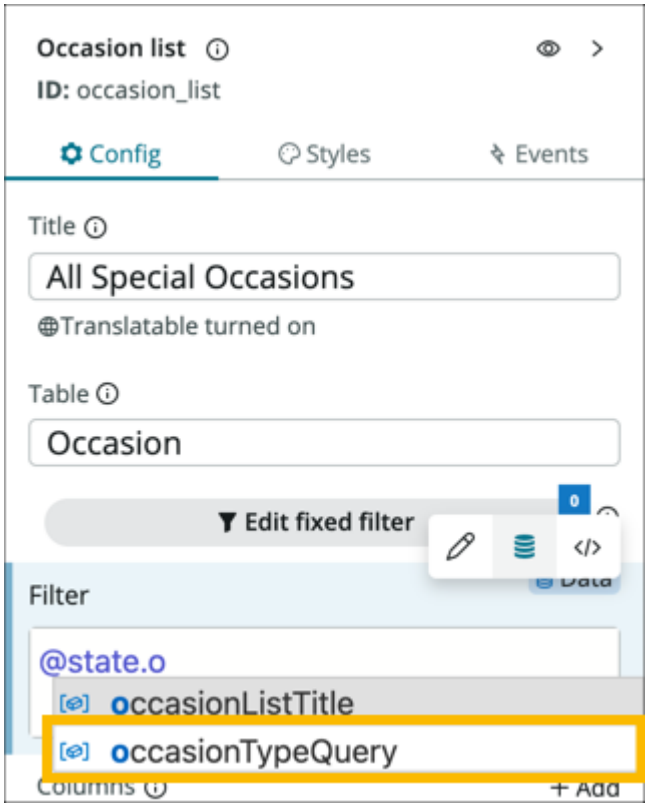
Binding client state parameters

Client state parameter values can be bound to component configuration fields the same way page context, payload, and data resources are. When configuring components, hover over a field, then select the **Dynamic data binding** button ( ) to bind a client state parameter to the field value. In the example, the **Dynamic data binding** button is highlighted for the Occasion List component's filter field.

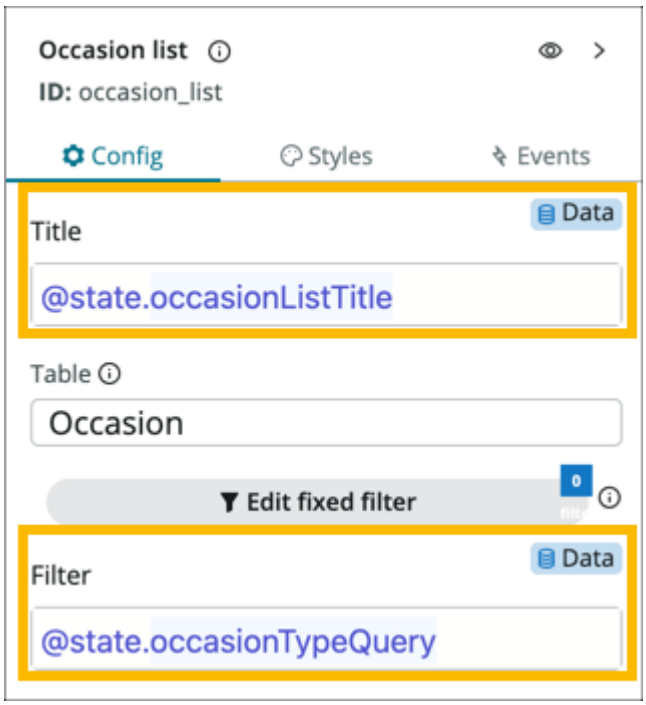


The character @ indicates data binding and the state object contains the client state parameters. Select a client state parameter from the choice list. The example shows selecting the `occasionTypeQuery` client state parameter. The default value for the `occasionTypeQuery` client state parameter is empty, so no filter will be applied by default.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

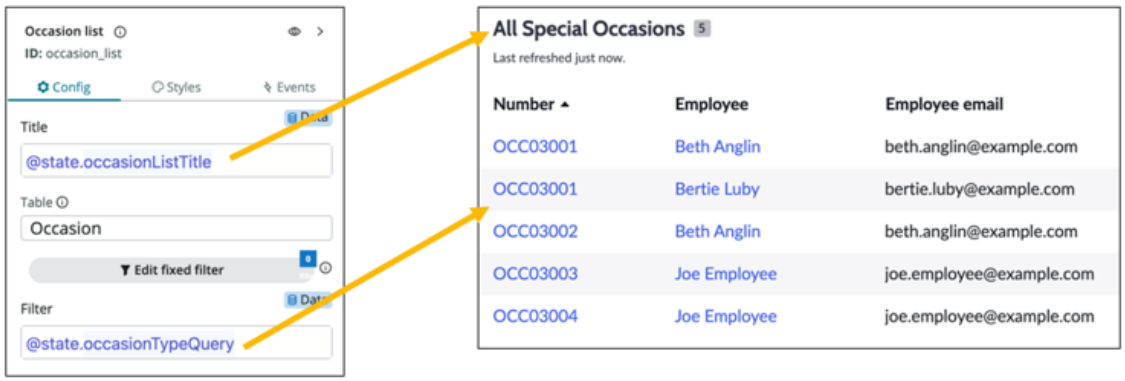


The Title for the component is set in a similar manner: select the **Dynamic data binding** button for the Title field or type @, then select the **state.occasionListTitle** client state parameter from the choice list. The default value for the `occasionListTitle` client state parameter is `All Special Occansions`. The example shows both Title and Filter configured to use client state parameters.



With both client state parameters applied, no filter is applied to the list and the title is `All Special Occasions`.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

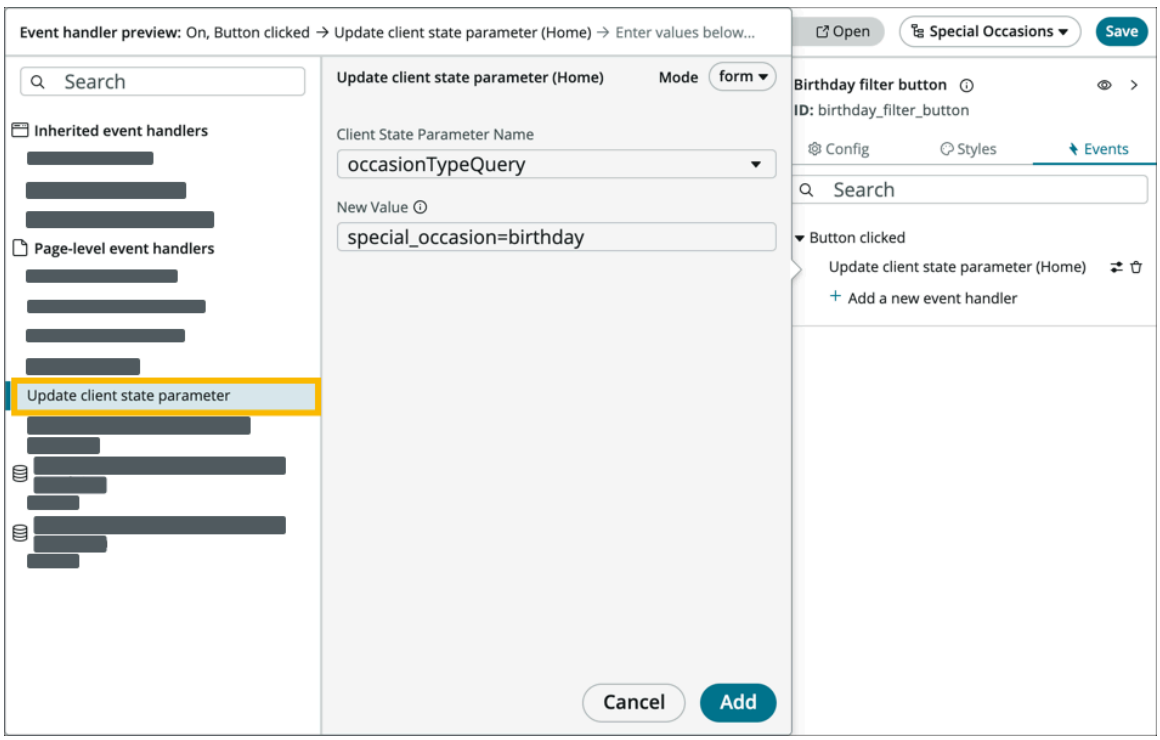


The `occasionTypeQuery` and `occasionListTitle` client state parameter values remain unchanged until a user interacts with a component that changes the value of the client state parameter.

Change client state parameter values

User the **Update client state parameter** event handler to change the value of a client state parameter in runtime. The **Update client state parameter** event handler has two properties: client state parameter to update and the new value to use for the client state parameter.

As an example, a `Birthdays` button is added above the `Occasion List` component in the `Special Occasions` application. The objective of this button is to change the `Title` of the list to `All Birthdays` and to adjust the filter of the list to only show birthdays. To achieve this objective, two event handlers are mapped to the **Button clicked** event for the button, one to change the `occasionTypeQuery` client state parameter and another to change the `occasionListTitle` client state parameter. The image shows the **Update client state parameter** event handler configured to set the `occasionTypeQuery` client state parameter to `special_occasion=birthday`, which will filter the list to only show birthdays.



When the **Button clicked** event is mapped to event handlers to update both the `occasionTypeQuery` and `occasionListTitle` client state parameters, clicking the

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

button updates the client state parameters, which automatically updates the `Occasion List` component with the new values.



Using client state parameters in UI Builder

Create a simple counter by adding the stylized text component and two buttons to an experience page. Use a client state parameter to implement the functionality so that when the buttons are selected the count increases or decreases.

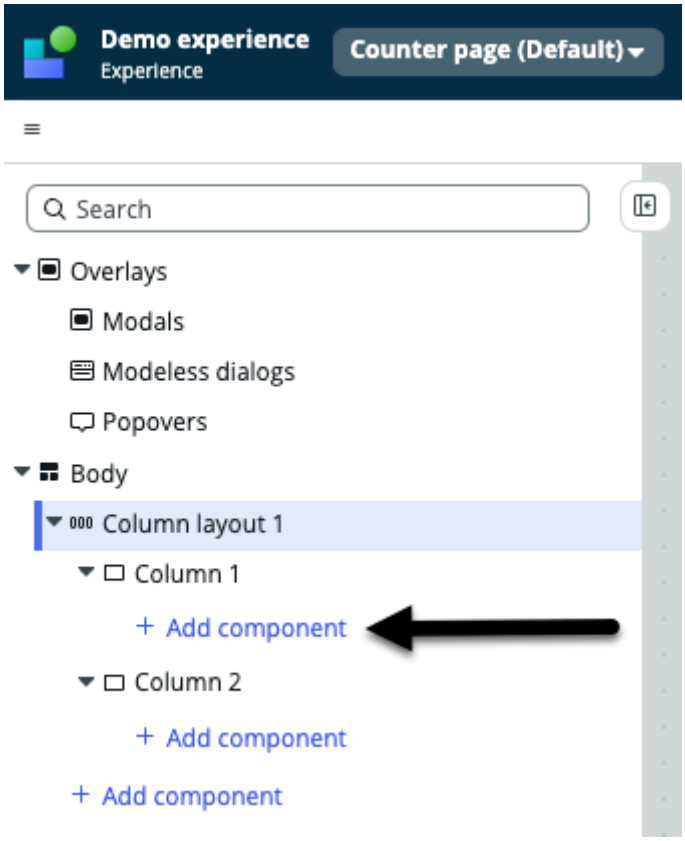
Before you begin

Role required: ui_builder_admin

Procedure

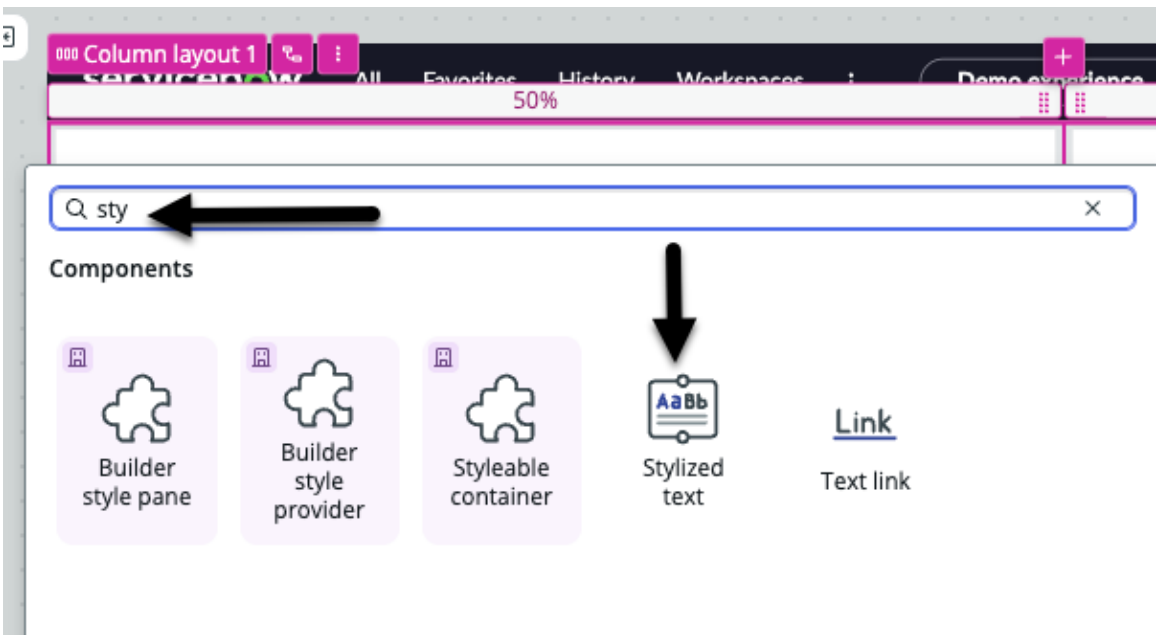
1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create > Experience**. See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create a page in UI Builder or open a page.
4. Add a column layout by selecting **+ Add content** in the content tree under **Body**.
5. On the **Layouts** tab, select **Two columns**.
6. Add the first component by selecting **+ Add content** in the content tree under **Column 1**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



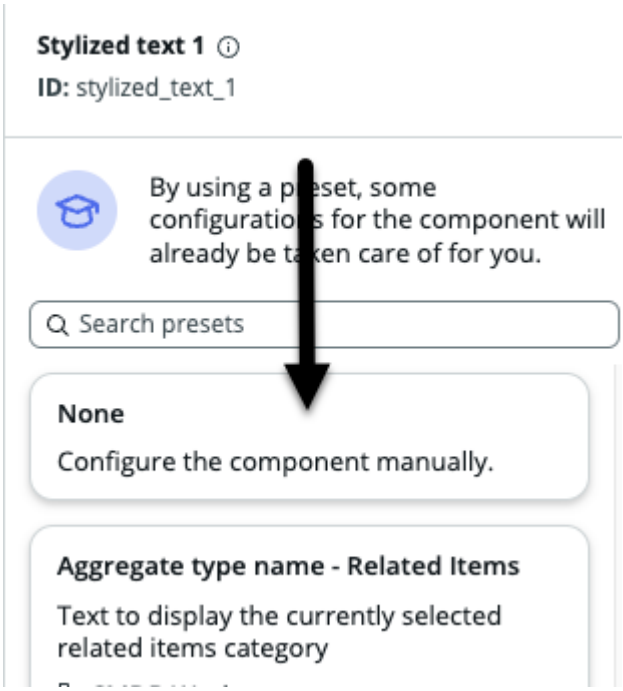
a. In **Search**, type `sty`.

b. Select the **Stylized text** component.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

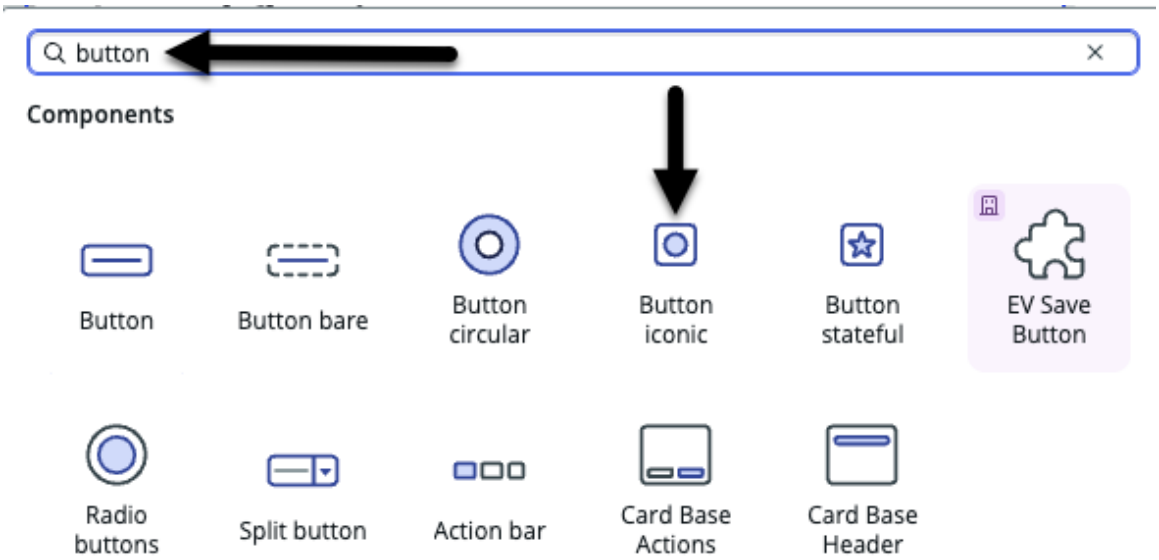
c. In the configuration panel, select **None - Configure the component manually**.



7. Add the second component by selecting **+ Add content** in the content tree under **Column 2**.

a. In **Search**, type `button`.

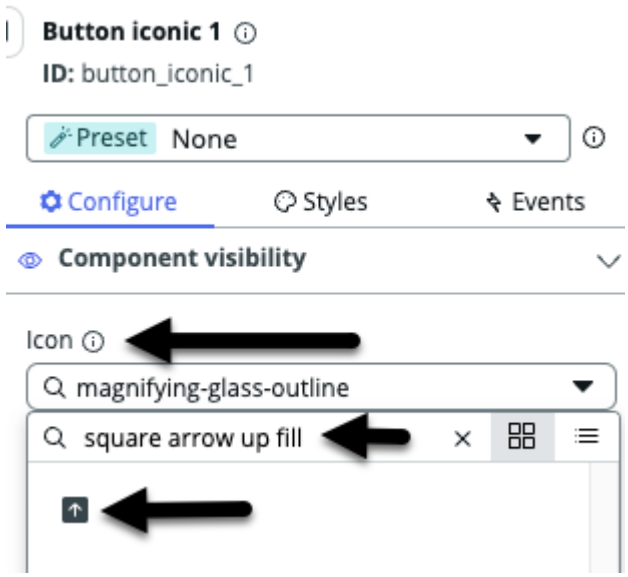
b. Select **Button Iconic**.



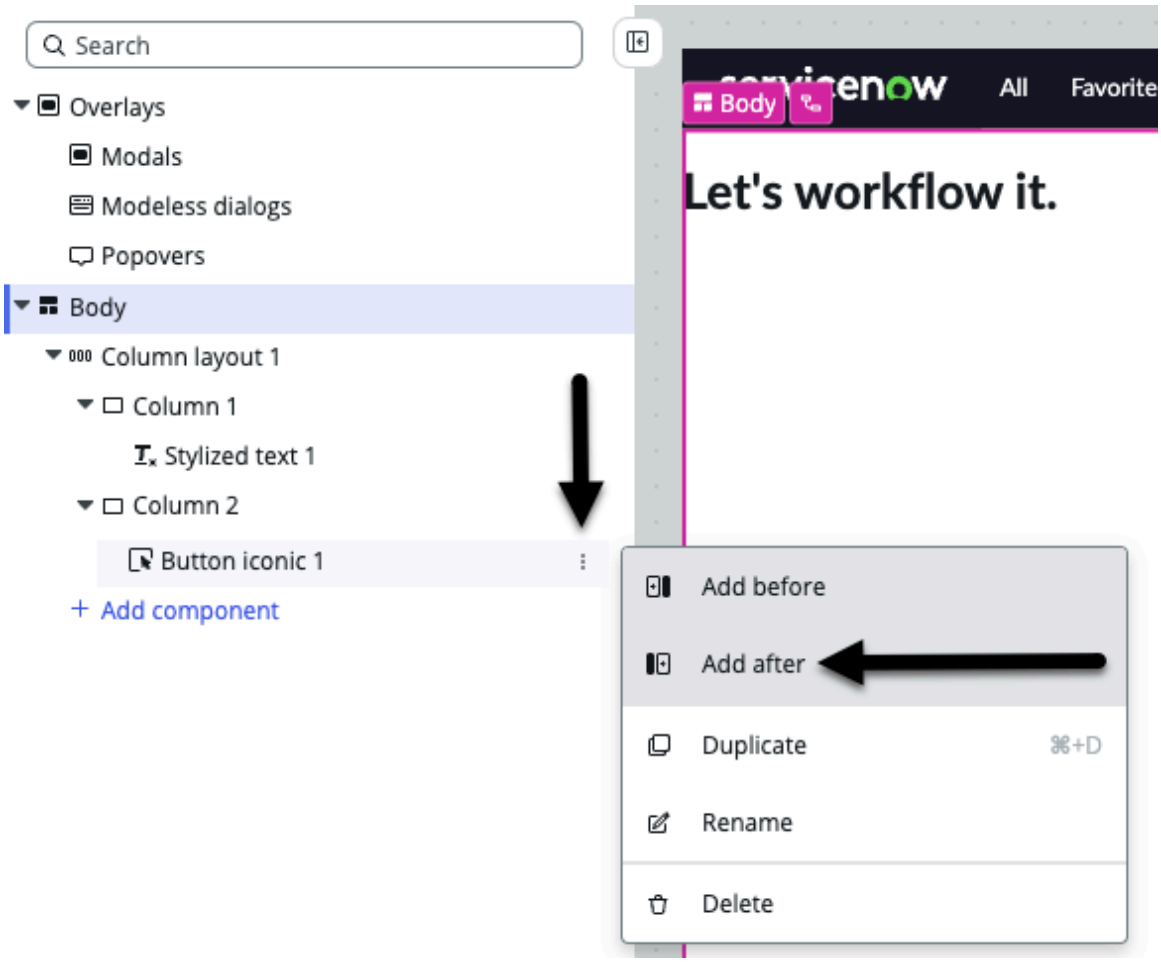
c. In the configuration panel, select **None - Configure the component manually**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

d. In **Icon**, select **Square Arrow Up Fill**.



8. Add the third component by pointing to **Button iconic 1** in the content tree, selecting the **Menu** icon, and selecting **Add after**.



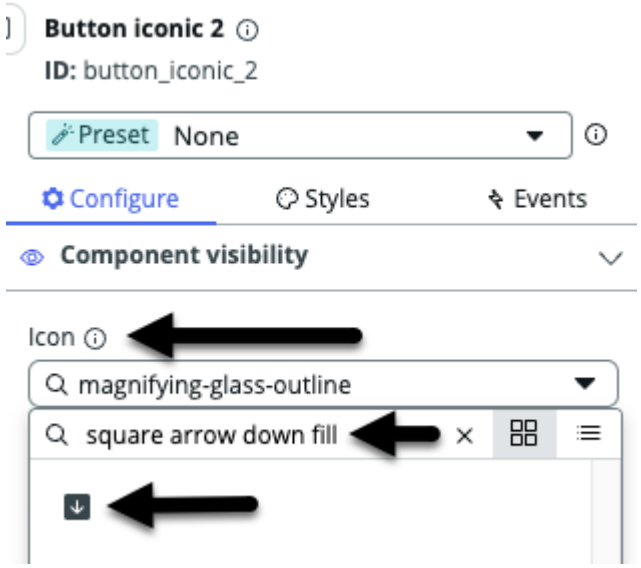
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. In **Search**, type `button` as you did in the previous step.

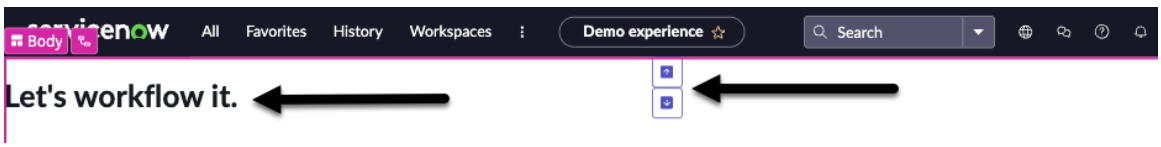
b. Select **Button Iconic** as you did in the previous step.

c. In the configuration panel, select **None - Configure the component manually** as you did in the previous step.

d. In **Icon**, select a different icon this time, the one named **Square Arrow Down Fill**.



9. Select **Save**.
The stylized text component and the two button iconic components are saved and appear on the stage.



10. Add a client state parameter by going to **Data and scripts**, pointing to **Client state parameters**, and selecting the **Add new (plus)** icon.

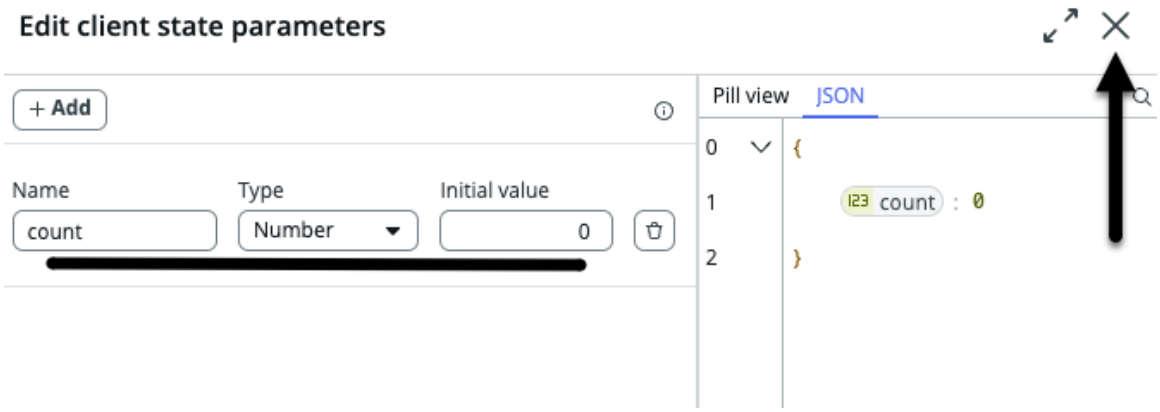
a. Change the **Name** by entering `count`.

b. Change the **Type** to **Number**.

c. Set the **Initial value** by entering the number 0.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

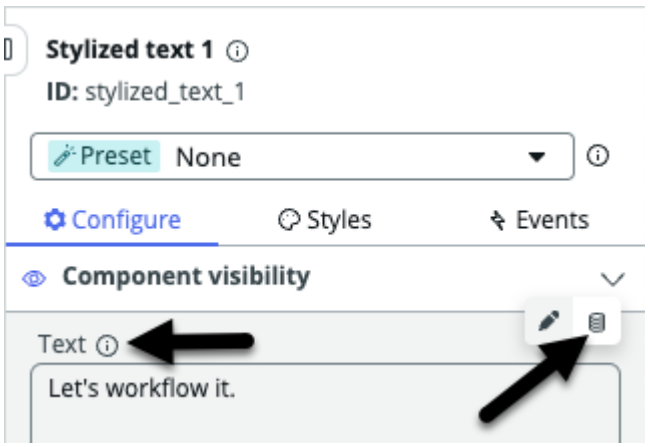
d. Select the **X** to close the window.



11. Bind the stylized text component to the client state parameter.

a. Select the stylized text component on the stage.

b. In the configuration panel, point to the **Text** field and select the **Bind data or use scripts** icon.



c. On the **Data types** tab, select **Client states**.

d. Double-click on the **count** pill.

e. Select **Apply**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



The **Text** field changes to **count**.

12. Now configure the buttons to increase or decrease the number.

a. Select the top button on the stage.

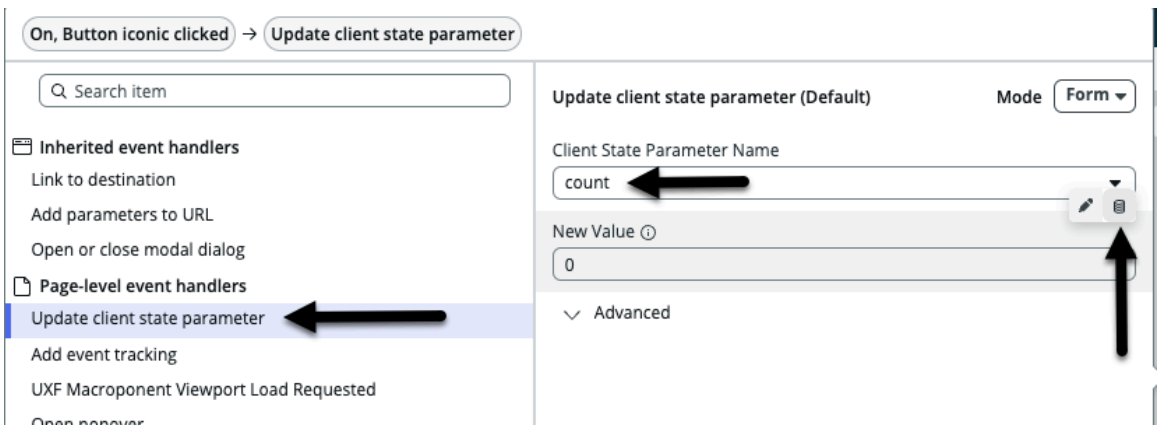
b. In the configuration panel, select the **Events** tab.

c. Select **+ Add event handler**.

d. Under **Page-level event handlers**, select **Update client state parameter**.

e. In **Client State Parameter Name**, select **count**.

f. Point to the **New Value** field and select the **Bind data or use scripts** icon.



g. On the **Data types** tab, select **Client states**.

h. Double-click on the **count** pill.

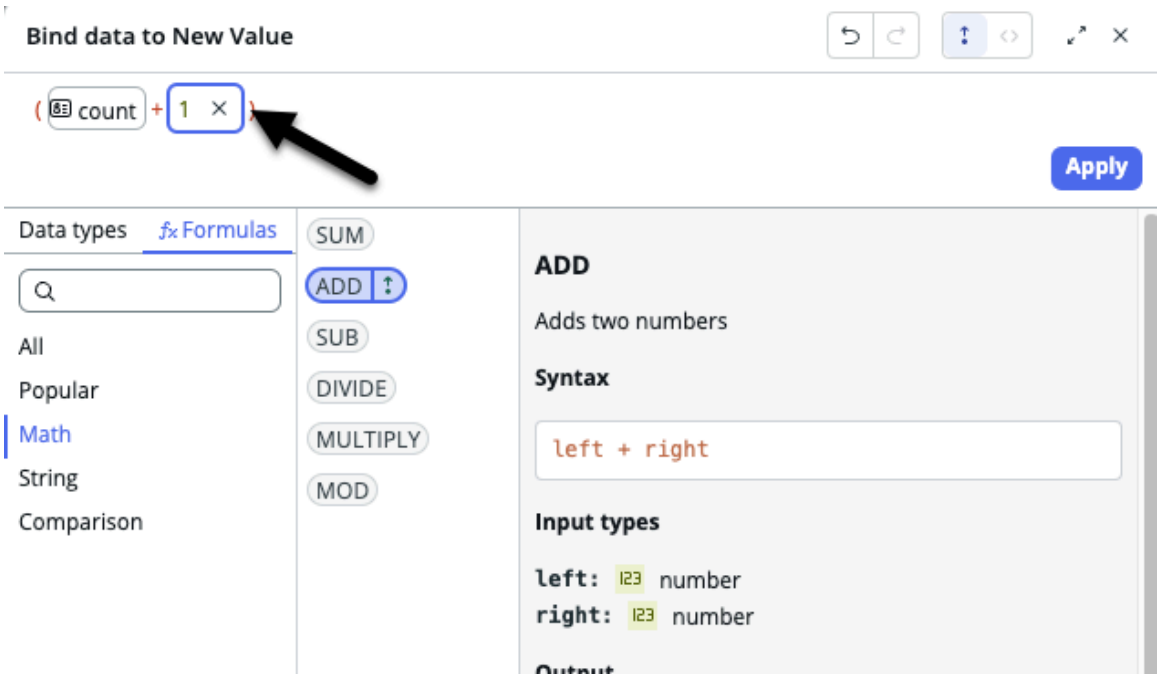
i. Select the **Formulas** tab.

j. In the list, select **Math**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

k. Double-click on the **ADD** pill.

l. In the **right** pill at the top, remove the text and enter `1`.



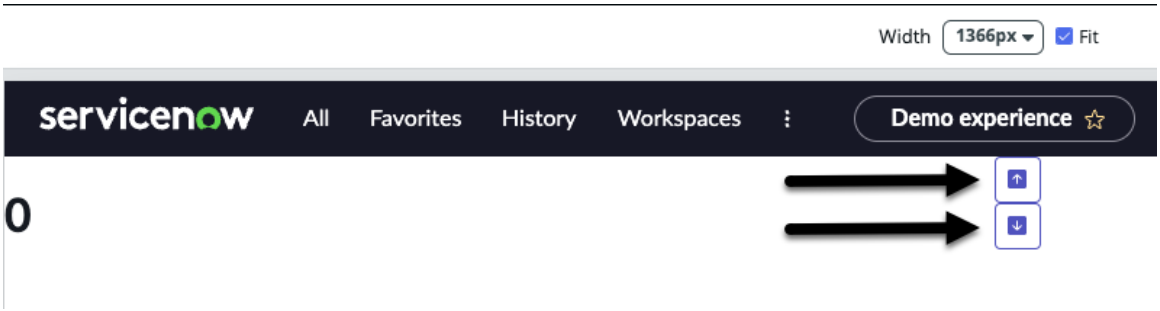
m. Select **Apply**.

n. Select **Add**.

13. Now configure the second button by following the same process that you used for the first button, but select the SUB pill instead of the ADD pill.
    a. Select the bottom button on the stage.
    b. In the configuration panel, select the **Events** tab.
    c. Select **+ Add event handler**.
    d. Under **Page-level event handlers**, select **Update client state parameter**.
    e. In **Client State Parameter Name**, select **count**.
    f. Point to the **New Value** field and select the **Bind data or use scripts** icon.
    g. On the **Data types** tab, select **Client states**.
    h. Double-click on the **count** pill.
    i. Select the **Formulas** tab.
    j. In the list, select **Math**.
    k. Double-click on the **SUB** pill.
    l. In the **right** pill at the top, remove the text and type `1`.
    m. Select **Apply**.
    n. Select **Add**.
14. Test the counter by selecting **Preview**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

15. Select the up arrow button to increase the count and the down arrow button to decrease the count.



Result

You added a button component to increase the count by one and another button component to decrease the count by one. You added the stylized text component to show the count as it increased and decreased.

For detailed, technical information about the button component, see Button Overview & API.

For detailed, technical information about the stylized text component, see Stylized Text Overview.

Update a component's state using client state parameters

Create and bind a client state parameter value to a component in UI Builder. By adding custom values to your components, these components can then be automatically updated through a script.

Before you begin

Role required: ui_builder_admin

About this task

You can add two components to a page and then connect them by using client state parameters and scripts. Start by creating a client state parameter and then binding the client state parameter to one of the components, such as a label. Next, you create a client script and bind it to a second component, such as a button, using an event handler. When you create the event handler, bind the client state parameter value to it to connect the two components. If you click one component, it changes the state of the other component. Client states are useful because you can add custom values to your components that can be automatically updated through a script. Think of a client state as a bucket for storing information that is specific to the page.

For example, you can add a button and label component to your page. The button changes the value of the label, such as changing the text color.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create or open a page.
   For more information, see Manage UI Builder pages and page variants.
4. Add two components to your page.
   For more information, see Add and configure components.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

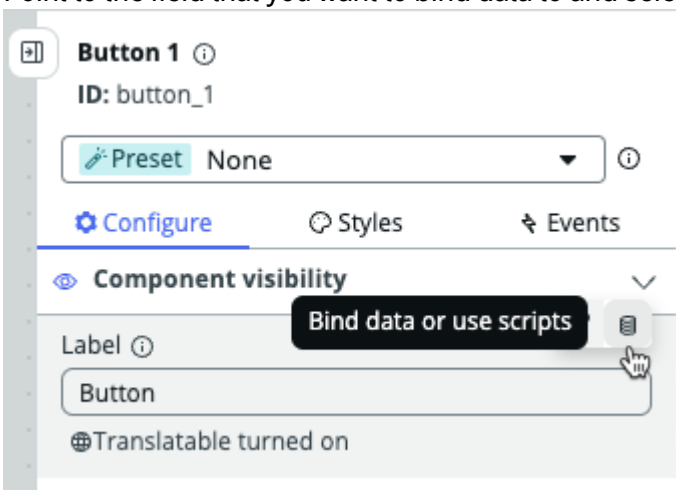
5. To define a client state parameter with an associated value, select Client state parameters in the lower-left panel.
6. Select **+ Add**.
7. Enter a name for the client state name, type, and initial value.

   The client state supports Strings, Numbers, Boolean, and JSON. For example, you could enter the name as `dynamic_label_value`, the type as `String`, and the initial value as `Initial Value`.
8. Bind the value of the client state parameter to your component.

   a. Select the component that you want to bind the client state parameter to.

   b. Open the configuration panel and select the **Configure** tab.

   c. Point to the field that you want to bind data to and select the dynamic data binding icon ( ).



   The data binding modal appears.

   d. Select **Client states** in the **Data types** tab.

   e. Double-click the client script that you want to bind to the component.

   f. Select **Apply**.
9. Select **Save**.
10. Bind the client state value to one component and create a script to connect the second component to the first.

    a. In the left pane, select the **+** icon next to **Client scripts**.

    b. Enter a script name that describes the task.
    For example, you could enter `Update Label Value` because that is what you want the script to do.

    c. In the Now Code Editor, add your script.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Let's say that you choose an API to call, and the parameters for the API, such as a state and a value. For example, you could call the `api.setState`, and include the `dynamic_label_value` as the first parameter, and a `NEW VALUE` as the second parameter.

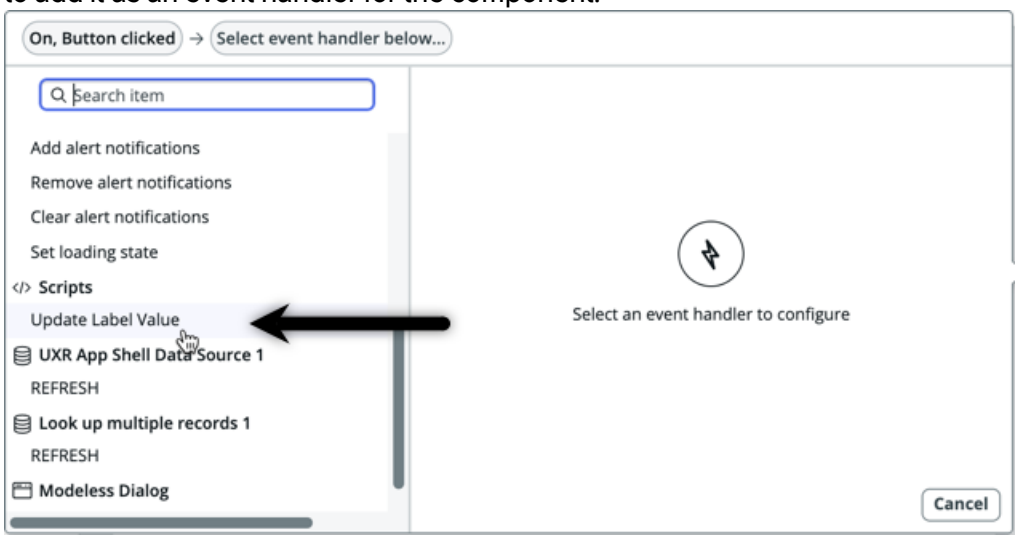
11. Add an event handler to your second component to call the new client script that you created.

    a. Select the component on your page that you want to bind the script to.

    b. Select the **Events** tab.

    c. Select **+ Add a new event handler**.

    d. Under the Scripts section of event handlers, select the script that you created, and select **+** to add it as an event handler for the component.



12. Select **Save**.
13. To preview your page and test the components to ensure they're connected, select **Preview**. When you select one component, it should change the state of the second component. For example, select the **Button** component to change the **Label value** component text from **Initial Value** to **NEW VALUE**.

Result

What you did in this procedure:

- Added two components to your page.
- Defined a client state parameter that had an initial value.
- Bound the client state parameter to the first component.
- Created a client script that updates a value that is defined by the client state parameter.
- Created an event handler for the second component to call the new client script that you just created
- Selected the second component so that it changed the state of the first component.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Define and bind client scripts to components

Add and edit client scripts in UI Builder so that you can update the client state through events. You can bind these scripts to any component by using an event handler.

Before you begin

Role required: ui_builder_admin

About this task

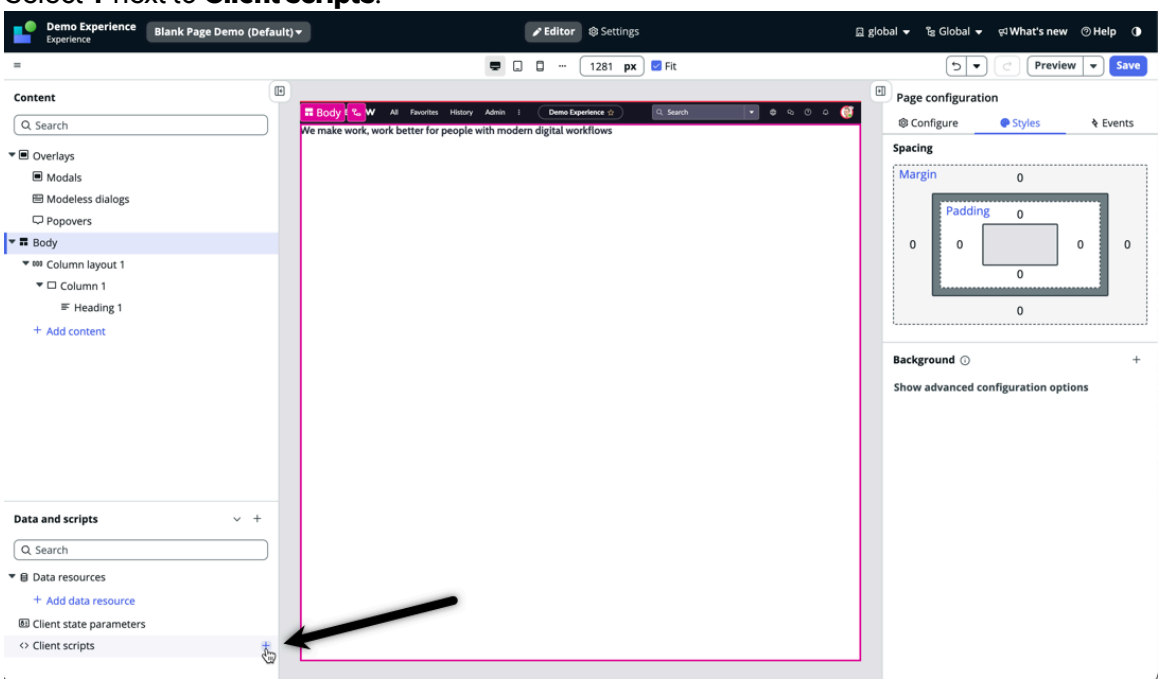
You can create JavaScript client scripts in UI Builder by using the Now Code editor. Then, you can add the client script as an event handler to update the client state, emit a handled event of your page, or execute a data resource operation. For example, you could write a script to increment a date or counter, and bind the script to a component event, like a button click. For more information about the Now Code Editor, see Edit code with the Now Code Editor (advanced feature).

With these scripts, you can do the following actions:

- Get available data, manipulate the data, and store it in the client state.
- Access data resource results.
- Execute data resource operations.
- Dispatch events.

Procedure

1. Navigate to **All** > **Now experience framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   For more information, see Configure how users interact with your applications in UI Builder.
3. Open or create a page.
   For more information about how to create a page in UI Builder, see Create a page in UI Builder.
4. **Optional:** If you do not have any components on your page, add a component to your page.
   For example, you can add a Heading component. For more information, see Add and configure components.
5. Select **+** next to **Client scripts**.

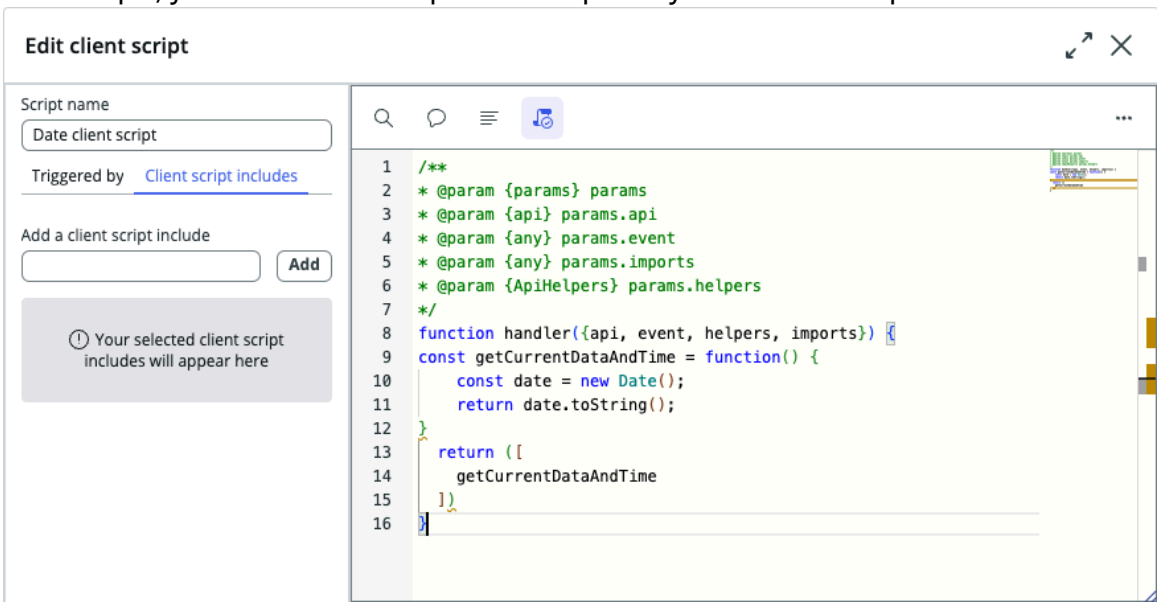


© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

6. Name your script.

   A descriptive name helps you know what the script does. It also makes it easier to find the script later when you add it to an event handler. The following example shows a simple date client script.

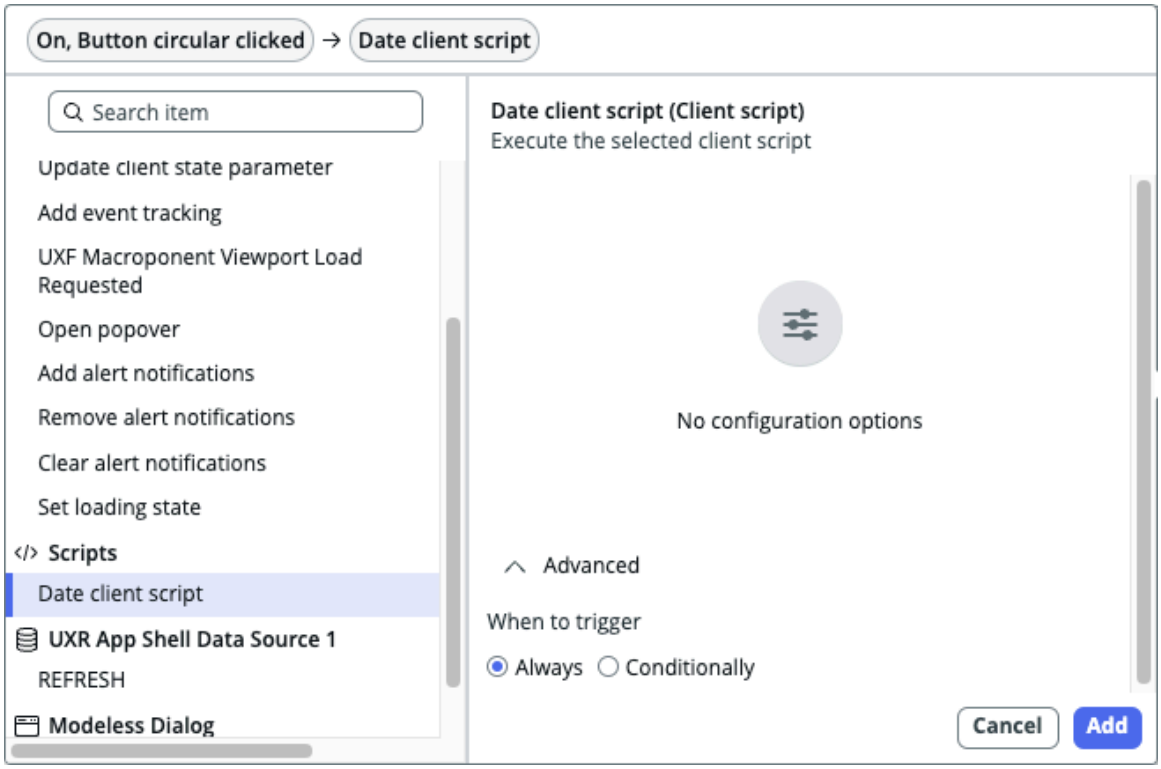
7. Write your script to perform an action.
   For example, you could write a simple date script that you bind to a component.



8. **Optional:** Add a **Script include** or **Associated components**, which shows up in the `imports` parameter of your client script function.
9. Select the component on your page that you want to bind the script to and then select **Events**.
10. Select **+Add a new event handler**, select the script that you created under **Scripts**, and then select **Add**.

    The following example shows a date client script.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



11. Select **Save**.
12. To preview your scripted component, select **Preview**.

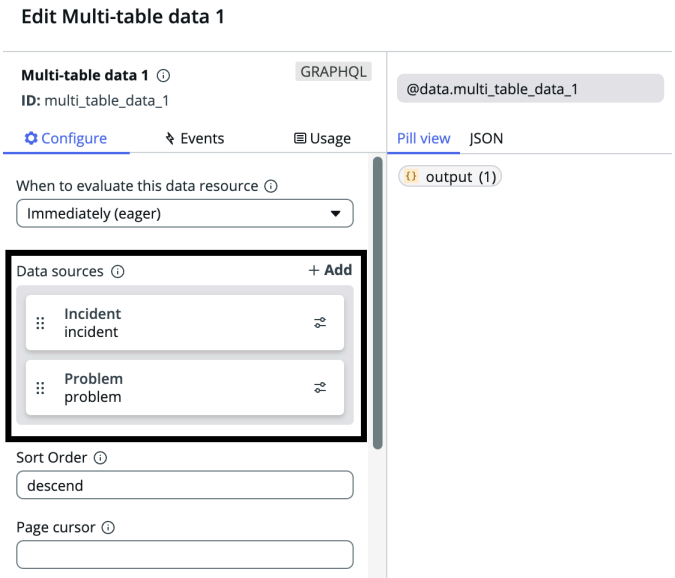
Multi-table data configuration

Present data from multiple tables using components and control the layout and styling.

A data resource fetches the data that UI Builder uses to display information in a component. Most often, a component is mapped to data contained in one table. However, there are numerous use cases where there's a need to map a single component to information in multiple tables, for example:

- Place the card component in a repeater and use multi-table data configuration to pull data from different sources to display on the cards.
- Create a portal page that displays a list of tasks pulled from different locations.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Note: You can configure more than one multi-table data resource on a single page to display different sets of data.

Multi-table data configuration and Entity View Action Mapper

The Entity View Action Mapper (EVAM) can also be used to configure cards and lists using different data sources. The EVAM is an option for users who understand and are comfortable working with JSON. The multi-table data configuration option gives you greater control over the presentation of data, makes it easier to configure the data, and enables you to remain in UI Builder. Use the option that you prefer. For more information about EVAM, see Work with the Entity View Action Mapper for UI Builder.

Fetch data from multiple sources

Learn how to obtain data from different sources to use in a single component.

Before you begin

Role required: ui_builder_admin

About this task

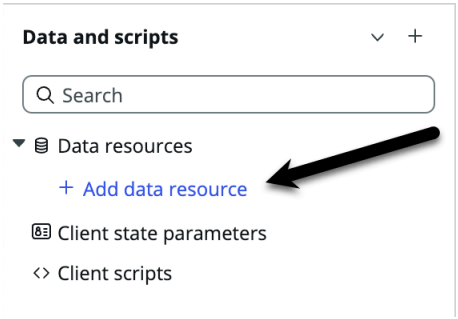
Place the card base container component in a repeater and use multi-table data configuration to present all active incidents and problems created in the last two years.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create > Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Create a page from scratch.
   For more information about how to create a page, see Create a page in UI Builder.
4. Add a multi-table data resource.

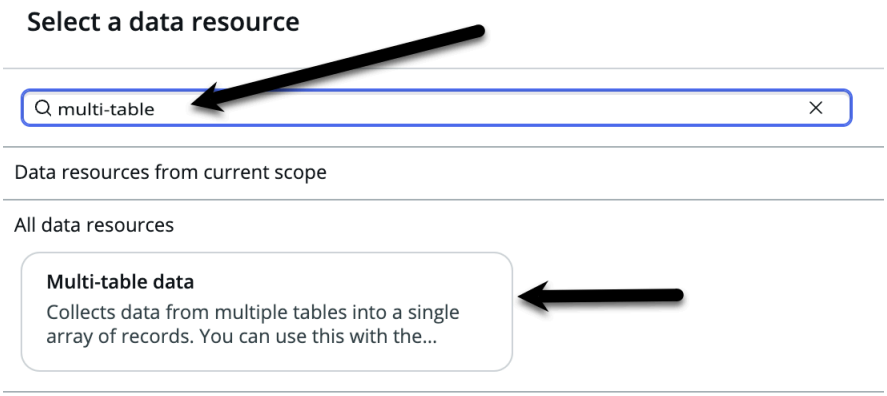
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. In the data shelf, select **+ Add data resource**.



b. In **Search**, enter `multi-table`.

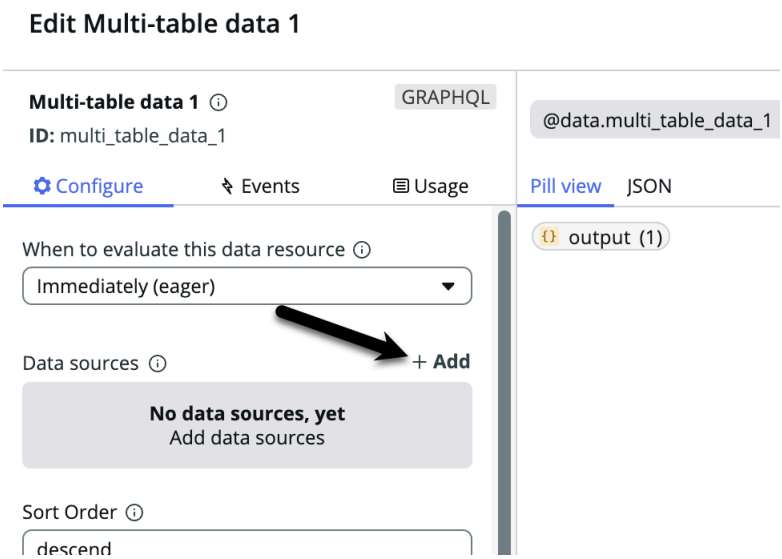
c. Select **Multi-table data**.



d. Select **Add**.

5. Configure the first data source.

a. In **Data sources**, select **+ Add**.



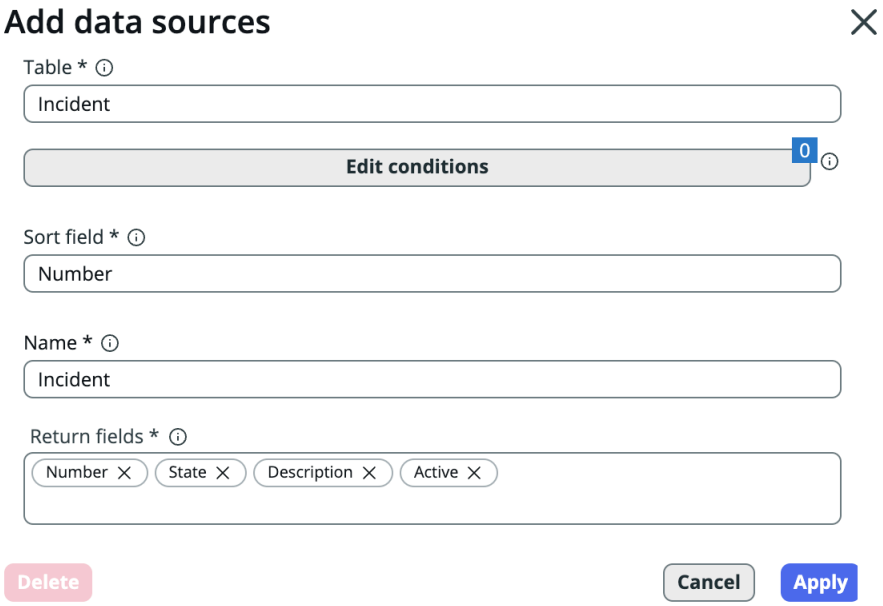
b. In **Table**, enter `incident` and select the **Incident** table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

c. In **Sort field**, enter `number` and select **Number**.

d. In **Name**, enter `Incident`.

e. In **Return fields**, add **Number**, **State**, **Description**, and **Active**.

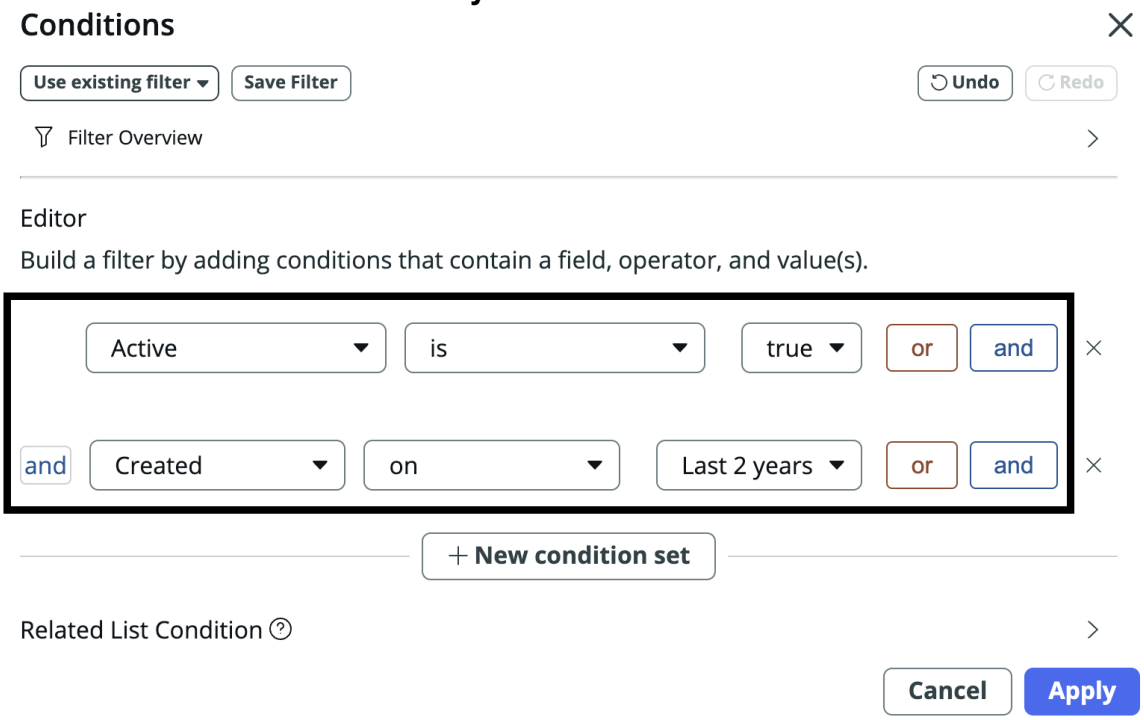


f. Select **Edit conditions**.

g. Build the condition **Active is true**.

h. Select **and**.

i. Build the condition **Created on Last 2 years**.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

j. Select **Apply**.

k. Select **Apply**.

6. Configure the second data source.

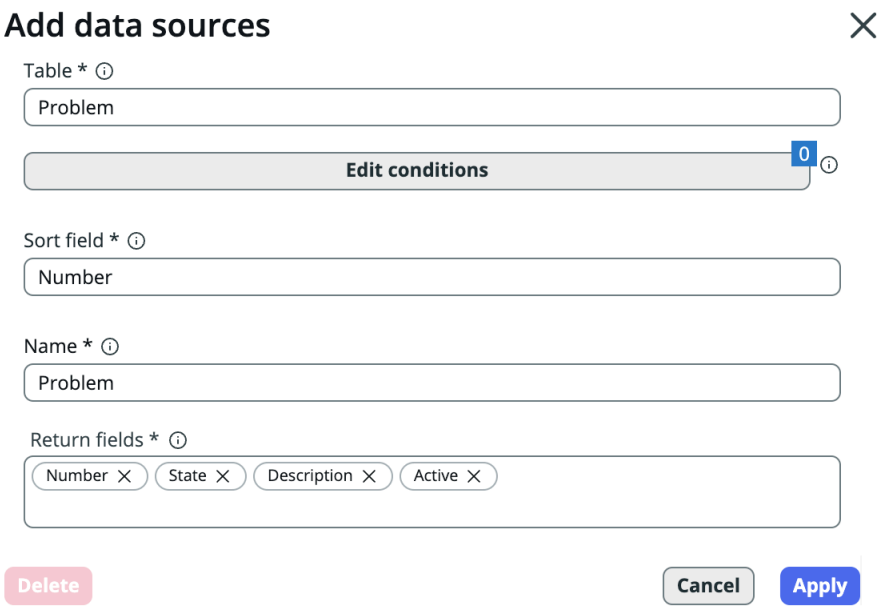
a. In **Data sources**, select **+ Add**.

b. In **Table**, enter `problem` and select the **Problem** table.

c. In **Sort field**, enter `number` and select **Number**.

d. In **Name**, enter `Problem`.

e. In **Return fields**, add **Number**, **State**, **Description**, and **Active**.
When creating a multi-table data resource, each data source should have the same return fields specified and in the same order. This helps ensure that the data displayed on the page is consistent and accurate.



f. Select **Edit conditions**.

g. Build the condition **Active is true**.

h. Select **and**.

i. Build the condition **Created on Last 2 years**.

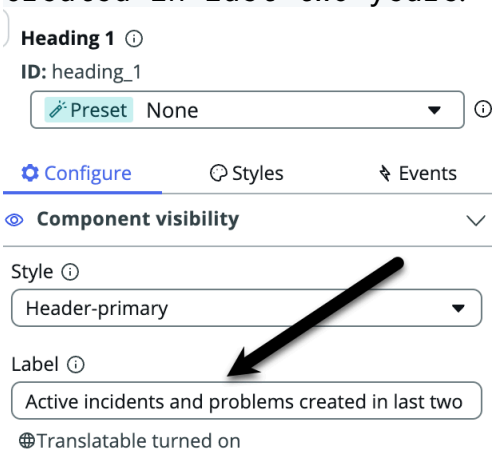
j. Select **Apply**.

k. Select **Apply**.

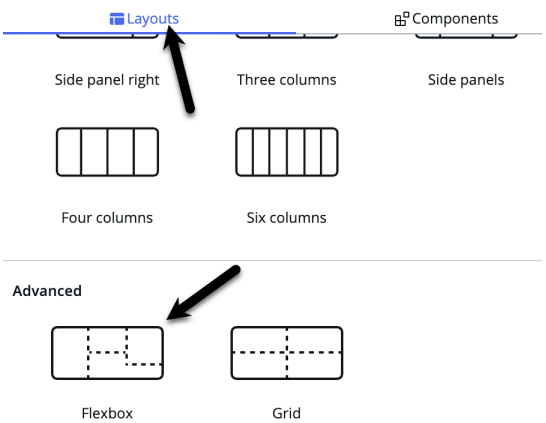
l. Select the **X** to close the **Edit Multi-table data** pop-up.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

7. Select **Save**.
8. Add the heading component.
   a. In the content tree, select **+ Add content** under **Body**.
   b. In **Search** enter `heading`.
   c. Select the **Heading** component.
   d. In the configuration panel, on the **Configure** tab, select **None - Configure the component manually**.
   e. In **Label**, delete the default text and enter `Active incidents and problems created in last two years`.



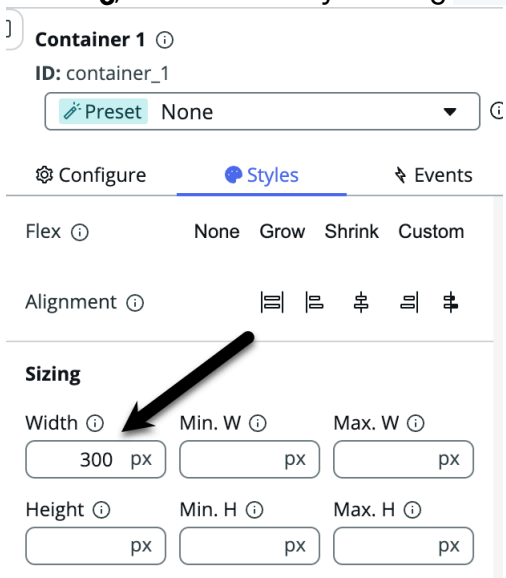
   f. Select **Save**.
9. Add a container.
   a. In the content tree, select **+ Add content** under **Heading 1**.
   b. On the **Layouts** tab, in the **Advanced** section, select **Flexbox**.



   c. In the configuration panel, on the **Configure** tab, select **None - Configure the component manually**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

d. In **Sizing**, set the **Width** by entering `300`.



e. Select **Save**.

10. Add the repeater component.

a. In the content tree, select **+ Add content** under **Container 1**.

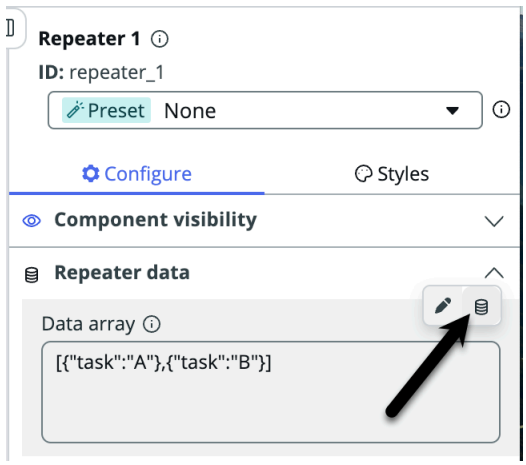
b. In **Search** enter `repeater`.

c. Select the **Repeater** component.

d. In the configuration panel, on the **Configure** tab, select **None - Configure the component manually**.

e. Select **Save**.

f. In the configuration panel, on the **Configure** tab, hover over the **Data array** field and select the bind data icon.



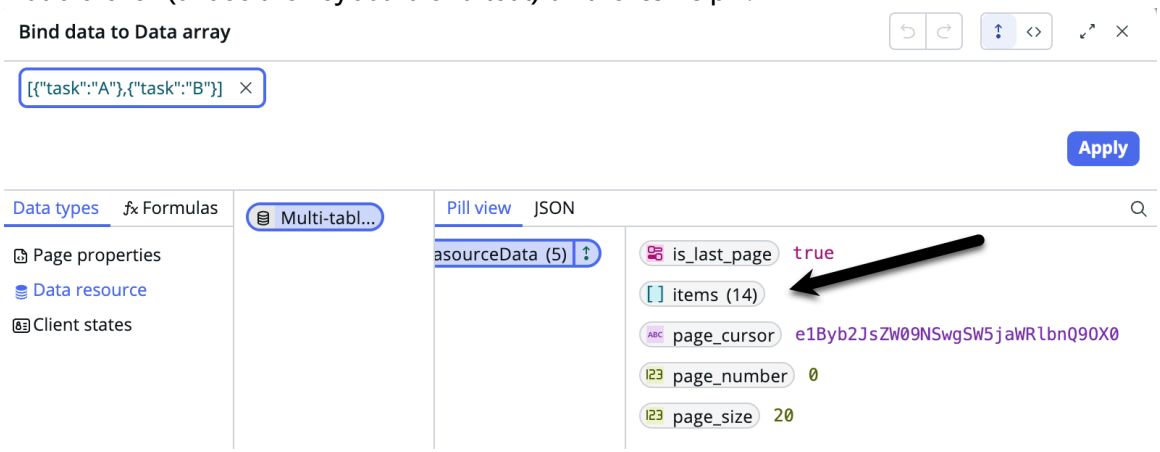
g. Under **Data types** select **Data resource**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

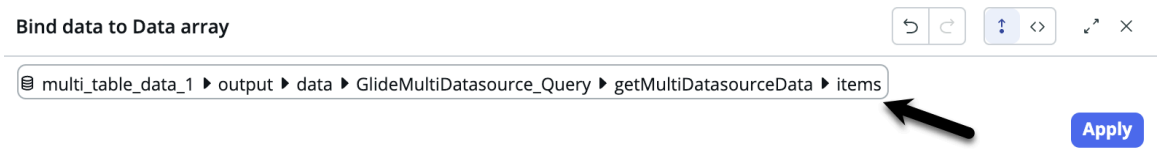
h. Select **Multi-table data 1**.

i. Select **output** > **data** > **GlideMultiDatasource_Query** > **getMultiDatasourceData**.

j. Double-click (or use the keyboard shortcut) on the **items** pill.



Check that the top section is accurate.



k. Select **Apply**.

l. Select **Save**.

11. Add the card base container component within the repeater.

a. In the content tree, select **+ Add content** under **Repeater 1**.

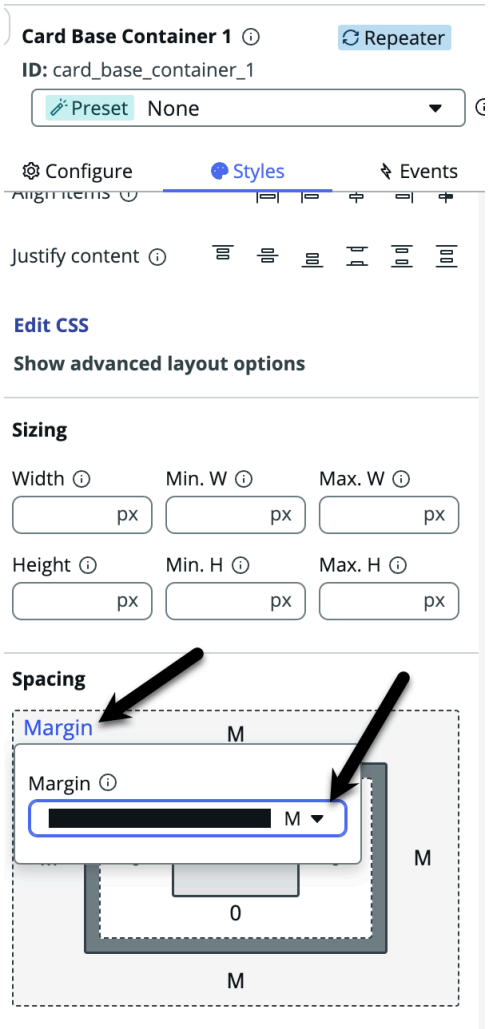
b. Locate and select the **Card Base Container** component.

c. In the configuration panel, on the **Configure** tab, select **None - Configure the component manually**.

d. Select the **Styles** tab.

e. To add a little space around the cards, go to **Spacing**, select **Margin**, and then select **M** (for medium).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



f. Select **Save**.

12. Add the heading component within the card base container.

a. In the content tree, select **+ Add content** under **Card Base Container 1**.

b. Locate and select the **Heading** component.

c. In the configuration panel, on the **Configure** tab, select **None - Configure the component manually**.

d. In the configuration panel, on the **Configure** tab, delete the default text in **Label**.

e. Point to the **Label** field and select the bind data icon.

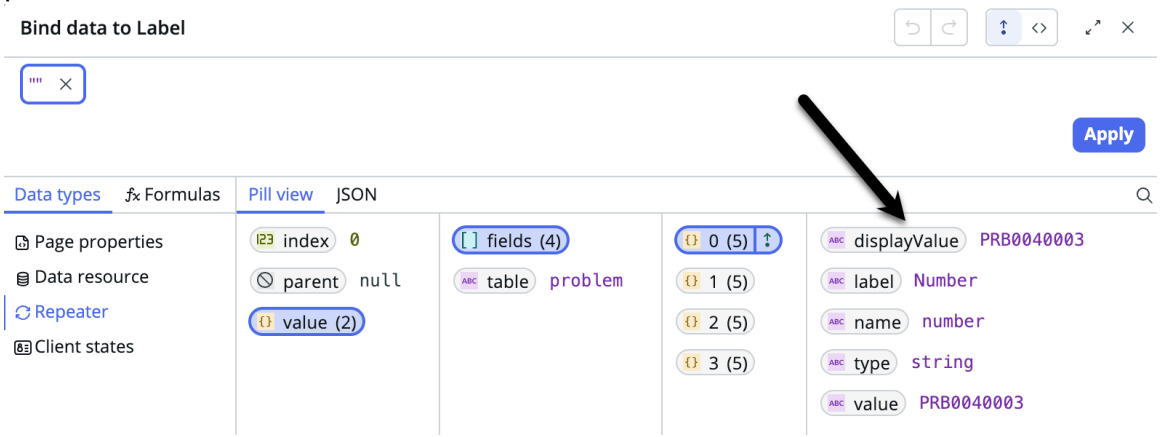
f. Under **Data types** select **Repeater**.

g. Select **value** > **fields**.

h. Select the top pill and in the last column check for **displayValue** with an incident or problem number.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

i. Double-click (or use the keyboard shortcut) on the **displayValue** pill with an incident or problem number.

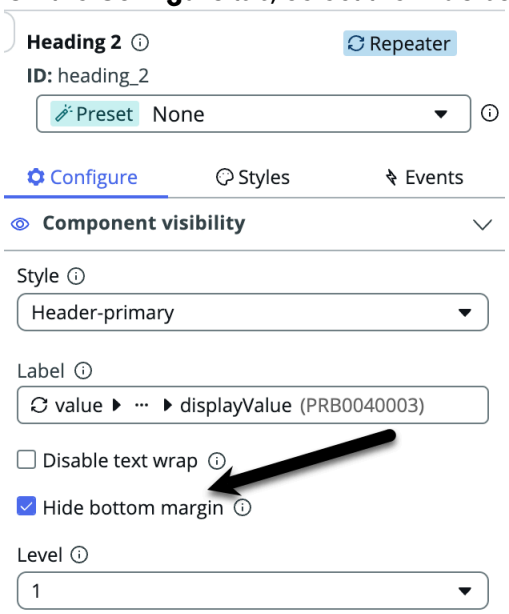


Check that the top section is accurate.



j. Select **Apply**.

k. On the **Configure** tab, select the **Hide bottom margin** option.

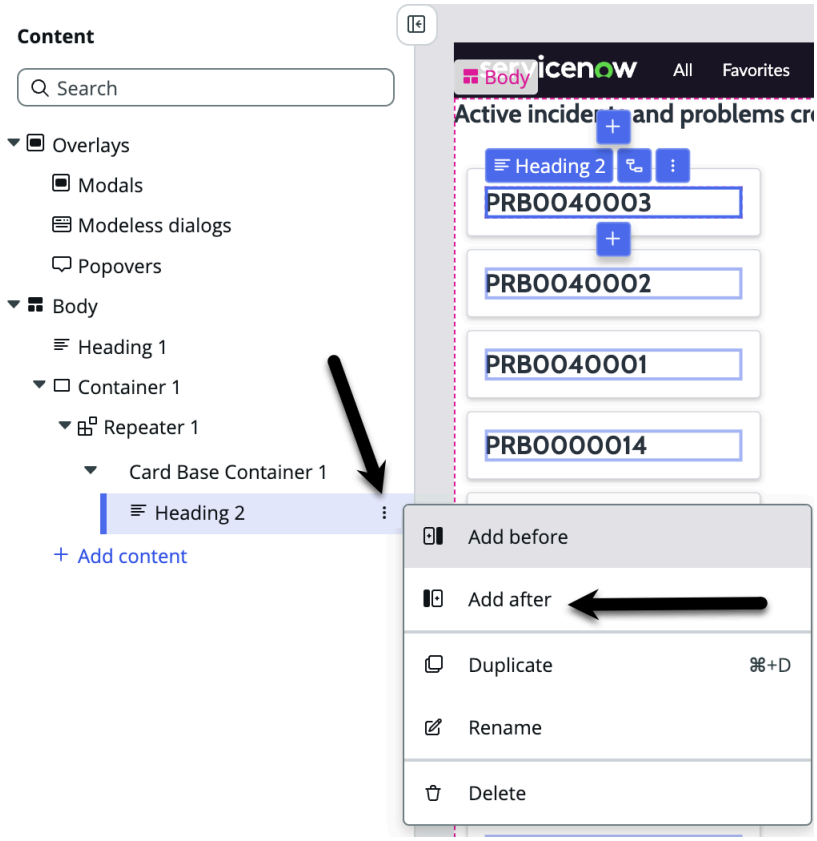


l. Select **Save**.

13. Add the first stylized text component to show the description of the incident or problem.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. In the content tree, point to **Heading 2**, select the menu (three vertical dots) icon, and select **Add after**.



b. Locate and select the **Stylized text** component.

c. In the configuration panel, on the **Configure** tab, select **None - Configure the component manually**.

d. In the configuration panel, on the **Configure** tab, delete the default text in **Text**.

e. Point to the **Text** field and select the bind data icon.

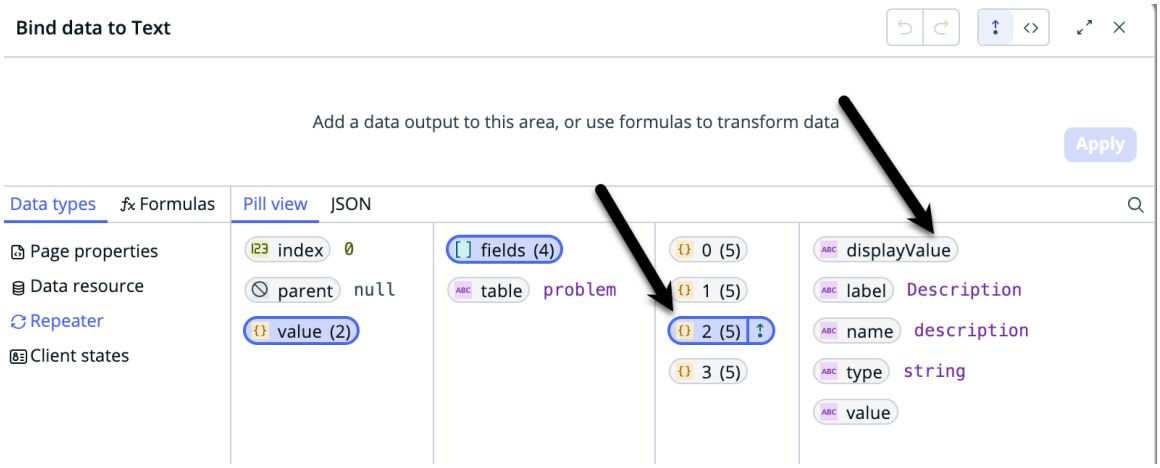
f. Under **Data types** select **Repeater**.

g. Select **value** > **fields**.

h. Select the third pill in the list and in the last column check that the **displayValue** contains no information.

i. Double-click (or use the keyboard shortcut) on the **displayValue** pill with no value.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Check that the top section is accurate.



j. Select **Apply**.

k. To reduce the text size, go to the **Configure** tab and change the **HTML tag** to **Paragraph**.

l. Select **Save**.

14. Add the second stylized text component to show the state of the incident or problem.

a. In the content tree, point to **Stylized text 1**, select the menu (three vertical dots) icon, and select **Add after**.

b. Locate and select the **Stylized text** component.

c. In the configuration panel, on the **Configure** tab, select **None - Configure the component manually**.

d. In the configuration panel, on the **Configure** tab, delete the default text in **Text**.

e. Point to the **Text** field and select the bind data icon.

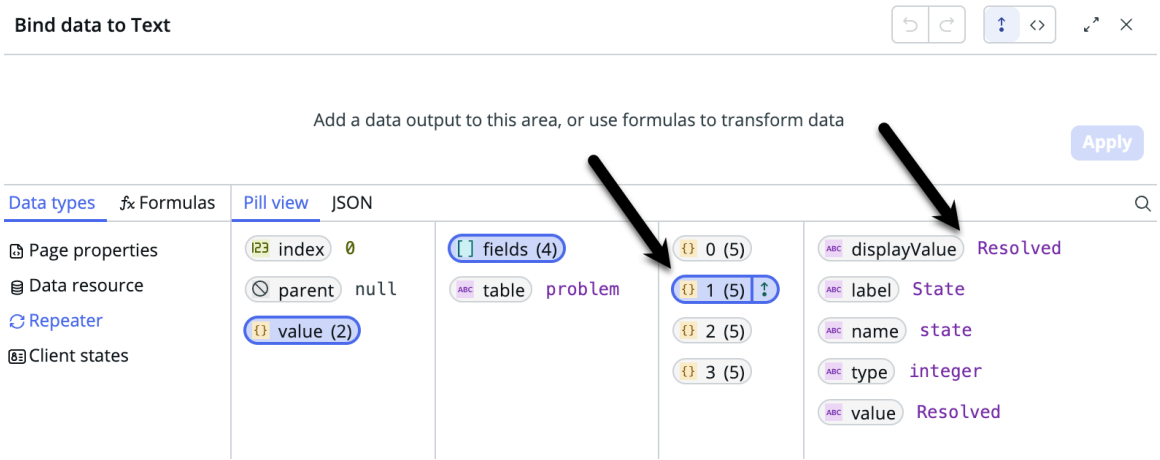
f. Under **Data types** select **Repeater**.

g. Select **value** > **fields**.

h. Select the second pill in the list and in the last column check that the **displayValue** contains a state such as **Resolved** or **In Progress**.

i. Double-click (or use the keyboard shortcut) on the **displayValue** pill with a state value.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Check that the top section is accurate.



j. Select **Apply**.

k. To reduce the text size, go to the **Configure** tab and change the **HTML tag** to **Paragraph**.

l. Select **Save**.

15. Select **Preview**.
    The page heading is at the top. Each card contains a heading with the incident or problem number. The cards also show the description (if one is available) of the incident or problem and the state. Data is being pulled successfully from two sources: the incident table and the problem table.
16. Select the **X** to close the preview overlay.

Work with the Entity View Action Mapper for UI Builder

With UI Builder, you can use the Entity View Action Mapper (EVAM) application to standardize how the data sources in your components are displayed in your cards and lists.

EVAM Overview

Entity View Action Mapper (EVAM) is an application that standardizes how different data sources are displayed in cards and lists. UI Builder uses EVAM data sources to show information in a component as a card grid view or as a list of information. You can add a toggle to your component to let users switch between card grid and list views.

EVAM consists of the following components:

- Entity (data source). Associated data that you intend to display, such as a community post or a user.
- View. How a card displays data and actions.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Actions. Action that it performs on the card. For example, you can activate a user into your system.
- Map. A process that maps the data source to generic fields that are displayed on the card. You can also associate actions that trigger from the card view.

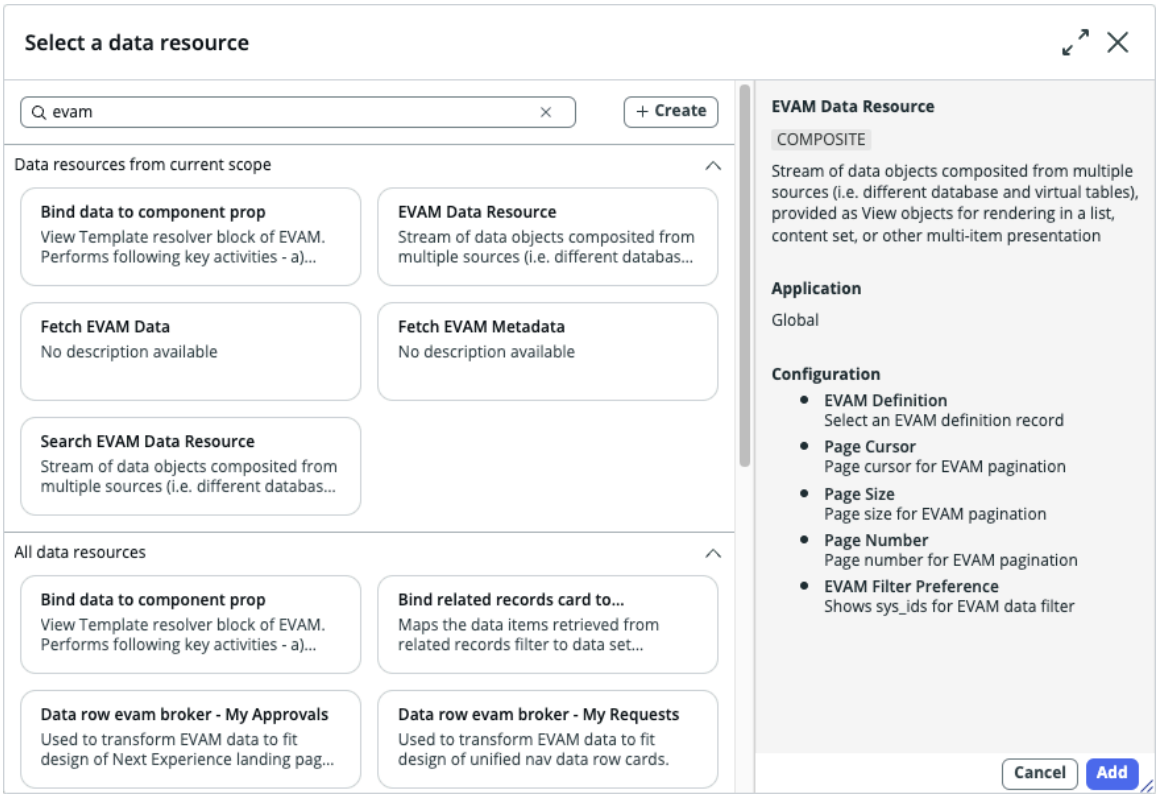
EVAM and Multi-table data configuration

The EVAM is an option for users who understand and are comfortable working with JSON. The multi-table data configuration option is an alternative that gives you greater control over the presentation of data, makes it easier to configure the data, and enables you to remain in UI Builder. Use the option that you prefer. For more information, see Multi-table data configuration.

EVAM data sources

You add EVAM data sources in UI Builder and bind them to a component.

EVAM data resources

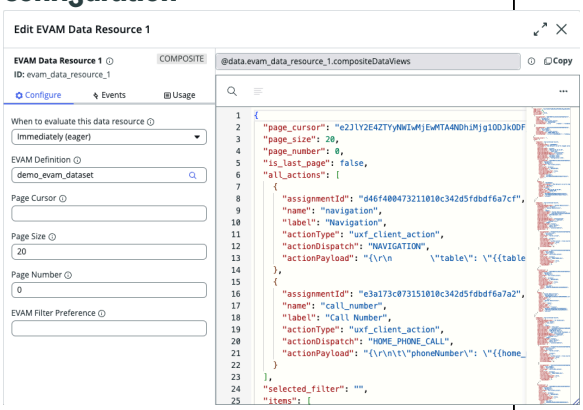


EVAM data sources

EVAM data source	Description
EVAM Data Resource	Add an EVAM definition and other information about the data source. • Type: Composite. • When to evaluate: In the **When to evaluate this data resource**, select **Immediately** to have the EVAM data resource instance evaluated on page load or select **Only when invoked** to use an event handler to evaluate the EVAM data resource.

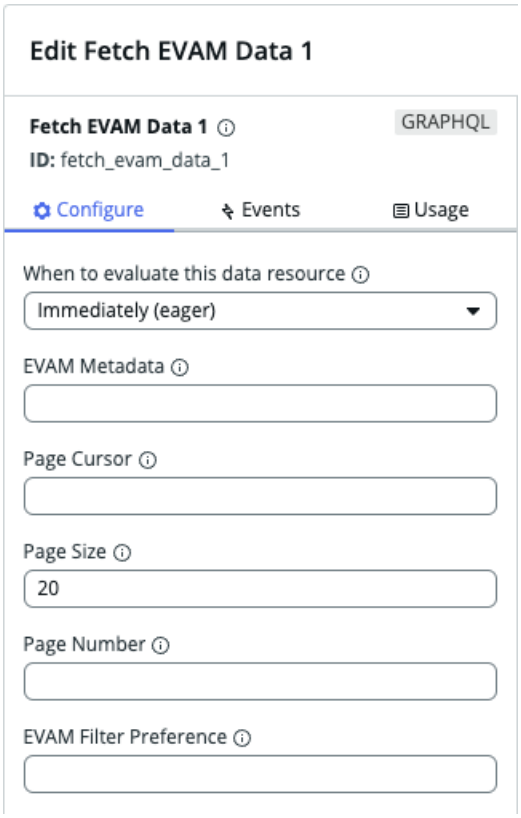
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

EVAM data sources (continued)

EVAM data source	Description
	• EVAM definition: In the **EVAM Definition** field, add the EVAM definition record that is associated with the data resource. • Page cursor: In the **Page Cursor** field, add a page cursor for the EVAM definition. • Page size: In the **Page Size** field, add a size for the EVAM pagination. • Page number: In the **Page Number** field, add the page number for the EVAM pagination. • EVAM Filter Preference: In the **EVAM Filter Preference** field, enter the sys_ids for the EVAM data filter. To add an event handler for when that data fetch is initiated, succeeded, or failed, select **Events**. In the **Now Code Editor** section, which is next to the Configuration pane, preview the EVAM definition script. **EVAM Data Resource configuration** 
Fetch EVAM Data	To add an EVAM definition and other information about the data source, select **Configuration**. • Type: GRAPHQL. • When to evaluate: To have the EVAM data resource instance evaluated on a page load, select **Immediately**. To use an event handler to evaluate the EVAM data resource, select **Only when invoked**. • EVAM definition: In the **EVAM Definition** field, enter the EVAM definition record that is associated with the data resource.

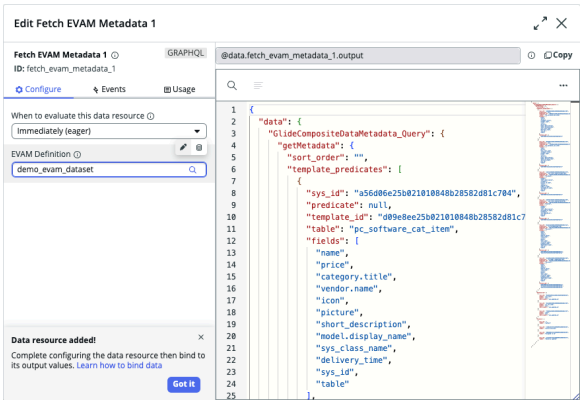
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

EVAM data sources (continued)

EVAM data source	Description
	• Page cursor: In the **Page Cursor** field, enter a page cursor for the EVAM definition. • Page size: In the **Page Size** field, enter a size for the EVAM pagination. • Page number: In the **Page Number** field, enter the page number for the EVAM pagination. • EVAM Filter Preference: In the **EVAM Filter Preference** field, enter the sys_ids for the EVAM data filter. To add an event handler for when that data fetch is initiated, succeeded, or failed, select **Events**. Preview the EVAM definition script in the **Now Code Editor** to the right of the Configuration pane. 
Fetch EVAM Metadata	To select when to evaluate the EVAM data resource and add an EVAM definition, select **Configuration**. • Type: GRAPHQL. • When to evaluate: To have the EVAM data resource instance evaluated on a page load,

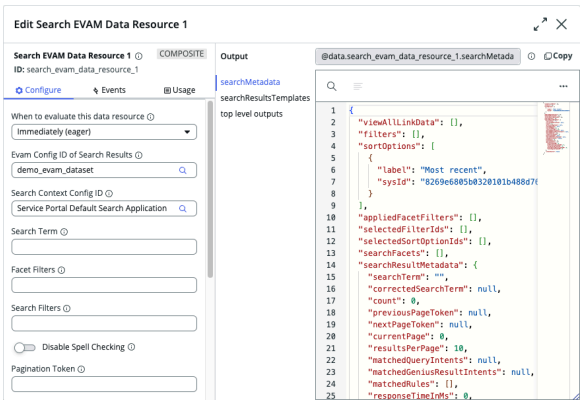
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

EVAM data sources (continued)

EVAM data source	Description
	select **Immediately**. To use an event handler to evaluate the EVAM data resource, select **Only when invoked**. • EVAM definition: In the **EVAM Definition** field, enter the EVAM definition record that is associated with the data resource. To add an event handler for when that data fetch is initiated, succeeded, or failed, select **Events**. Preview the EVAM definition script in the **Now Code Editor** to the right of the Configuration pane. 
Search EVAM Data Resource	To add an EVAM definition, and other information about the data source, select **Configuration**. • Type: Composite. • When to evaluate: To have the EVAM data resource instance evaluated on a page load. select **Immediately**. To use an event handler to evaluate the EVAM data resource, select **Only when invoked**. • EVAM Config ID: In the **EVAM Config ID** field, add the **EVAM Config ID** that is associated with the data record. • Search Context Config ID: In the **Search Context Config ID** field, add the **Search Context Config ID** that you are searching for. • Search Term: In the **Search Term** field, add the search terms that you are searching for. • Facet Filters: In the **Facet filters** field, add any facet filters to help define your search.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

EVAM data sources (continued)

EVAM data source	Description
	• Search Filters: In the **Search Filters** field, add any search filters to help define your search. • Disable Spell Checking: In the **Disable Spell Checking** field, select if you desire spell checking to be on or off during your search. • Pagination Token: In the **Pagination Token** field, enter a pagination token if needed. To add an event handler for when that data fetch is initiated, succeeded, or failed, select **Events**. Preview the EVAM definition script in the **Now Code Editor** to the right of the Configuration pane. 

Add an Entity View Action Mapper data resource to a page

Add an Entity View Action Mapper (EVAM) data resource to your page in UI Builder so that you can standardize how the data sources in your components are displayed in your cards and lists.

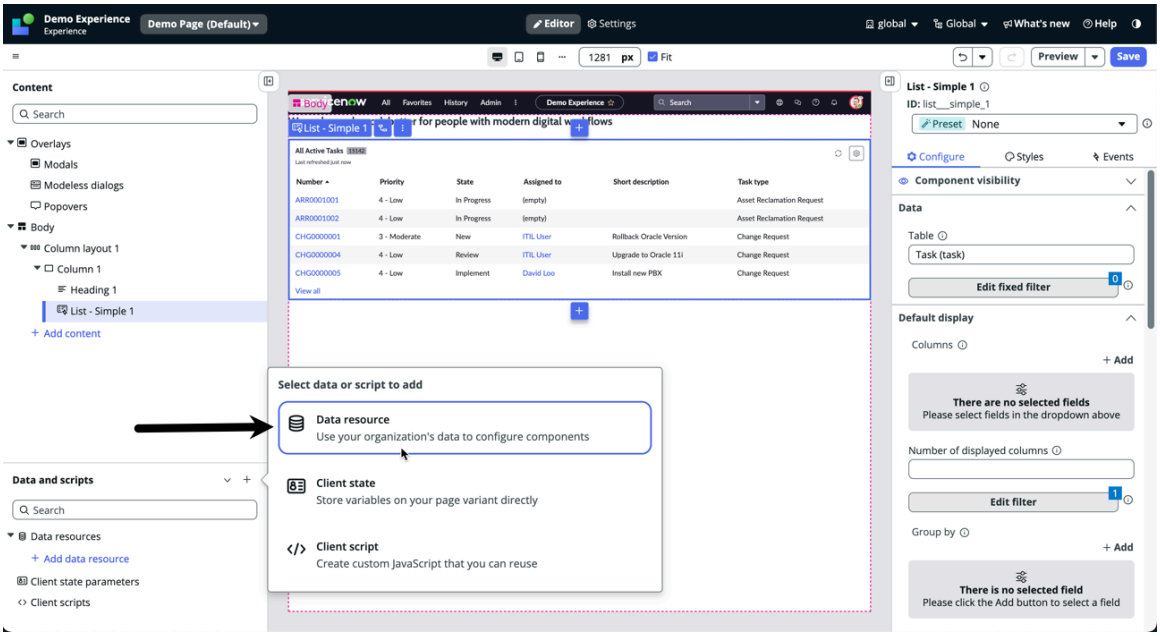
Before you begin

Role required: ui_builder_admin

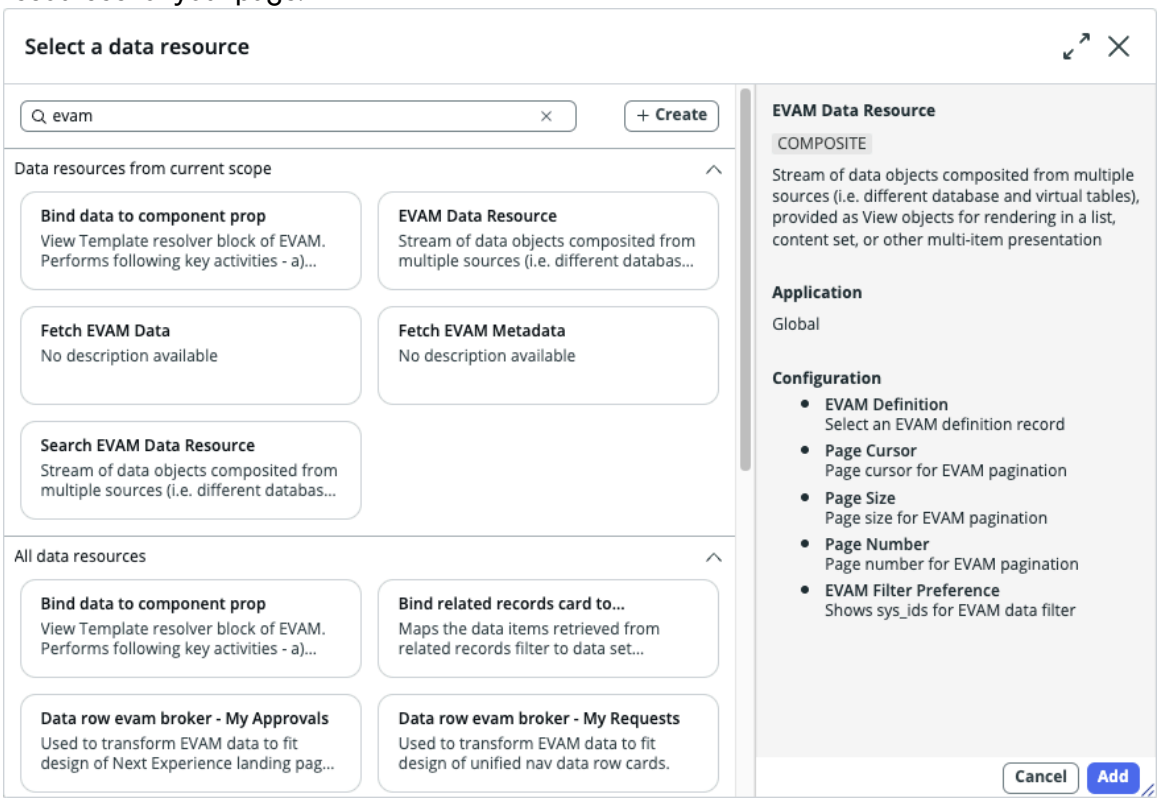
Procedure

1. Navigate to **All** > **Now experience framework** > **UI Builder**.
2. Open an experience to work in or create an experience by selecting **Create** > **Experience**.
   See Configure how users interact with your applications in UI Builder for more information on creating experiences.
3. Open or create a page.
   For more information, see Manage UI Builder pages and page variants.
4. **Optional:** If you do not have any components on your page, add a **Data set** component to your page.
   For more information, see Add and configure components.
5. In the **Data and scripts** panel, select the **+** icon and then select **Data resource**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



6. Enter `evam` in the search field.
7. Select the EVAM data resource that you want to add and repeat this step for all the EVAM data resources for your page.

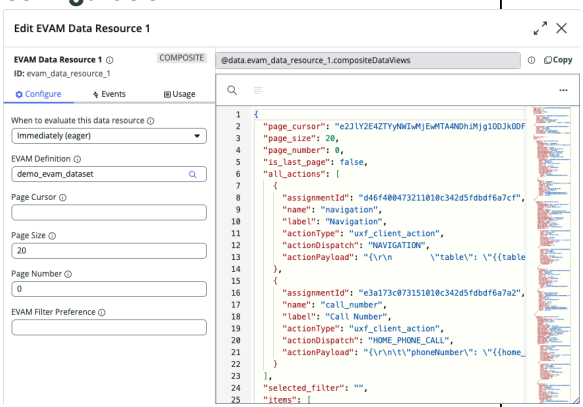


8. Configure each EVAM data resource.

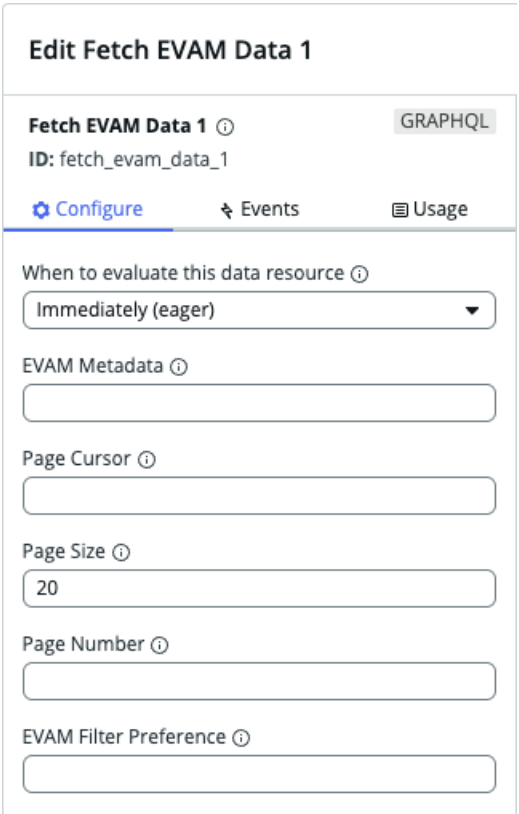
EVAM data sources

EVAM data source	Description
EVAM Data Resource	Add an EVAM definition and other information about the data source.

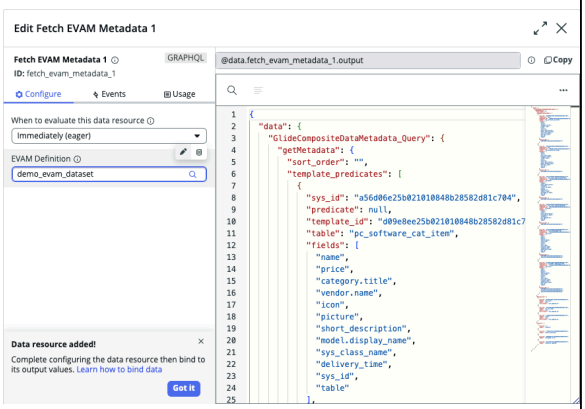
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

EVAM data source	Description
	○ Type: Composite. ○ When to evaluate: In the When to evaluate this data resource list, select **Immediately** to have the EVAM data resource instance evaluated on page load or select **Only when invoked** to use an event handler to evaluate the EVAM data resource. ○ EVAM definition: In the **EVAM Definition** field, add the EVAM definition record that is associated with the data resource. ○ Page cursor: In the **Page Cursor** field, add a page cursor for the EVAM definition. ○ Page size: In the **Page Size** field, add a size for the EVAM pagination. ○ Page number: In the **Page Number** field, add the page number for the EVAM pagination. ○ EVAM Filter Preference: In the **EVAM Filter Preference** field, add the sys_ids for the EVAM data filter. To add an event handler for when that data fetch is initiated, succeeded, or failed, select **Events**. In the Now Code Editor section, which is next to the Configuration pane, preview the EVAM definition script. **EVAM Data Resource configuration** 
Fetch EVAM Data	To add an EVAM definition and other information about the data source, select **Configuration**. ○ Type: GRAPHQL. ○ When to evaluate: To have the EVAM data resource instance evaluated on a page load, select **Immediately**. To use an

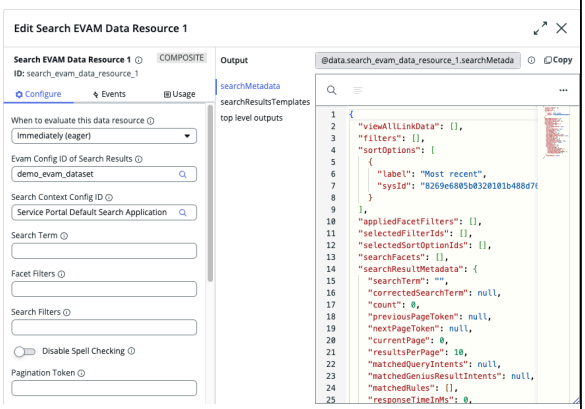
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

EVAM data source	Description
	event handler to evaluate the EVAM data resource, select **Only when invoked**. ○ EVAM definition: In the **EVAM Definition** field, add the EVAM definition record that is associated with the data resource. ○ Page cursor: In the **Page Cursor** field, add a page cursor for the EVAM definition. ○ Page size: In the **Page Size** field, add a size for the EVAM pagination. ○ Page number: In the **Page Number** field, add the page number for the EVAM pagination. ○ EVAM Filter Preference: In the **EVAM Filter Preference** field, add the sys_ids for the EVAM data filter. To add an event handler for when that data fetch is initiated, succeeded, or failed, select **Events**. Preview the EVAM definition script in the **Now Code Editor** to the right of the Configuration pane. 
Fetch EVAM Metadata	To select when to evaluate the EVAM data resource and add an EVAM definition, select **Configuration**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

EVAM data source	Description
	○ Type: GRAPHQL. ○ When to evaluate: To have the EVAM data resource instance evaluated on a page load, select **Immediately**. To use an event handler to evaluate the EVAM data resource, select **Only when invoked**. ○ EVAM definition: In the **EVAM Definition** field, add the EVAM definition record that is associated with the data resource. To add an event handler for when that data fetch is initiated, succeeded, or failed, select **Events**. Preview the EVAM definition script in the **Now Code Editor** to the right of the Configuration pane. 
Search EVAM Data Resource	To add an EVAM definition and other information about the data source, select **Configuration**. ○ Type: Composite. ○ When to evaluate: To have the EVAM data resource instance evaluated on a page load, select **Immediately**. To use an event handler to evaluate the EVAM data resource, select **Only when invoked**. ○ EVAM Config ID: In the **EVAM Config ID** field, add the **EVAM Config ID** that is associated with the data record. ○ Search Context Config ID: In the **Search Context Config ID** field, add the Search Context Config ID that you are searching for. ○ Search Term: In the **Search Term** field, add the search terms that you are searching for ○ Facet Filters: In the **Facet filters** field, add any facet filters to help define your search.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

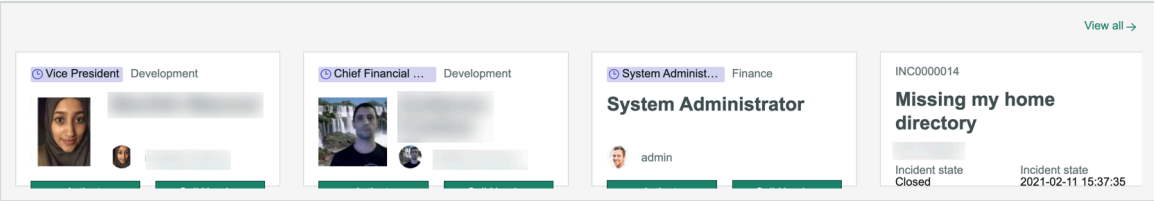
EVAM data source	Description
	○ Search Filters: In the **Search Filters** field, add any search filters to help define your search. ○ Disable Spell Checking: In the **Disable Spell Checking** field, select if you desire spell checking to be on or off during your search. ○ Pagination Token: In the **Pagination Token** field, add a pagination token if needed. To add an event handler for when that data fetch is initiated, succeeded, or failed, select **Events**. Preview the EVAM definition script in the **Now Code Editor** to the right of the Configuration pane. 

9. Configure your data set component:
   - Bind an EVAM data resource to your component.
   - Set the **Show grid/list** toggle to on to let users choose between the grid or list view of the EVAM information on the page.
   - Set the other configuration settings for the cards on the page.
10. Select **Save**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

11. View and test your page by selecting the **Preview** button.

Finished state of the EVAM



Edit code with the Now Code Editor (advanced feature)

Now Code Editor is a rich-text editor like interface that supports Cascading Style Sheets (CSS), Hypertext Markup Language (HTML), JavaScript, Extensible Markup Language (XML), and JavaScript Object Notation (JSON). Use Now Code Editor to modify UI configuration, data resource configuration, styles, events, client-side and server-side scripts in Next Experience UI Builder components.

Now Code Editor supports the following features:

- Basic editing
- Debugging
- Command palette
- Code formatting
- Syntax checking and highlighting
- Auto-suggestions
- Script macros for common code

Basic editing

Action	Description
Format code	Applies proper indentation to the script. Keyboard shortcut: • Windows: Shift+Alt+F • Mac: Shift+Option+F
Highlight syntax	Highlights the syntax of the code.
Check syntax	Checks for formatting errors and highlights syntax errors. • Windows: Shift+Alt+C • Mac: Shit+Option+C
Show suggestions	Displays a list of valid elements at the insertion point such as:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Action	Description
	• Class names • Function names • Object names • Variable names Select and click an entry to add it to the script. Keyboard shortcut: • Windows: Control+Space • Mac: Control+Space You can also enable or turn off **Syntax highlighting** from the **Settings** menu.
Toggle comments	Comments one or more lines of code with two consecutive forward-slashes //. Keyboard shortcut: • Windows: Control+/ • Mac: Command+/
Show minimap	Displays the minimap of the code snippet. You can display or hide the minimap option, from the **Settings** menu.
Enable word wrap	Enables word wrap function in the editor area. You can toggle the **Enable word wrap** option from the **Settings** menu.
Show command palette	Displays a list of available commands for the common operations. You can execute editor commands, find and replace text, fold and unfold code blocks, toggle comments and many more tasks using the same interactive window. Keyboard shortcut • Windows: F1 • Mac: F1
Expand editor or collapse editor	Expands or collapses the editor. Keyboard shortcut • Windows: Control+M • Mac: Control+M

Debugging

To launch Script Debugger, click the Script Debugger icon in the toolbar.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: You can add a breakpoint, conditional breakpoint, or logpoint, only when debugging is enabled and selected language is JavaScript.

Task	Do this
Add breakpoint	Right-click beside a line number in the ruler area and select **Add breakpoint**.
Add conditional breakpoint	1. Right-click beside a line number in the ruler area and select **Add conditional breakpoint**. 2. Enter a break condition in the editor.
Add logpoint	Right-click beside a line number in the ruler area and select **Add logpoint**.
Compare text in Diff mode	Use the side-by-side view icon and inline view icon to toggle between views.

Code editor macros

for

- Description: Inserts a standard for loop with an example array.
- Output:

```
for (var i=0; i< myArray.length; i++) {
 //myArray[i];

}
```

method

- Description: Inserts an empty JavaScript function template.
- Output:

```
/*-------------------------------------------------------
-------------
   * Description:
   * Parameters:
   * Returns:

 -------------------------------------------------------
-----------*/
   : function() {

   },
```

info

- Description: Inserts a *GlideSystem* information message.
- Output:

```
gs.addInfoMessage(gs.getMessage(""));
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

doc

- Description: Inserts a comment block for describing a function or parameters.
- Output:

```
/**

* Description:

* Parameters:

* Returns:
*/
```

vargror

- Description: Inserts a *GlideRecord* query for two values with an OR condition.
- Output:

```
var gr = new GlideRecord('');

var qc = gr.addQuery('field', 'value1');

qc.addOrCondition('field', 'value2');
gr.query();

while (gr.next()) {


}
```

vargr

- Description: Inserts a standard *GlideRecord* query for a single value.
- Output:

```
var gr = new GlideRecord("");
gr.addQuery("name", "value");
gr.query();
if (gr.next()) {

}
```

Component Builder

Learn how to use Component Builder to assemble reusable components for your UI Builder pages.

Create custom components to reuse across pages with component builder

Reuse custom components across experiences and pages in UI Builder.

Create custom components in UI Builder's component building UI. The component building UI shares many similarities to page building and enables you to configure components the same way you would when configuring a page in UI Builder.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The power of custom components

Building custom components improves efficiency, consistency, and maintainability across your experience. By creating reusable UI elements, you avoid duplicating configurations, ensure a uniform look and feel, and make it easier to manage updates or design changes. Custom components also simplify complex layouts, support flexible configurations, and enable for easier testing and debugging. They're also helpful in team environments, where shared components help streamline collaboration and keep the user experience consistent at scale.

Custom components are especially valuable when designing pages for different user types within your experience. For example, you can create a reusable component that includes both a list and a data visualization, then tailor its content or behavior based on the user group enabling you to maintain a consistent layout while delivering role-specific information.

Components edited in the Component Builder will automatically update on all pages where they are used.

Important: Custom components created in an instance remain available only within that instance until they are migrated to other instances using update sets or application installations.

Custom components or page collections

To build efficient, scalable digital experiences, it's important to reuse elements wherever possible. Two ways to do this are through custom components and page collections. Each serves a distinct purpose depending on the scope of reuse that you need.

Custom Components

Use custom components when you want to replicate a specific part of a page like a heading, list, form, or buttons across multiple pages.

You want to apply a consistent theme or configuration to a component or group of components.

You're designing pages to be modular or flexible where parts of a page may change but shared components remain consistent.

You want to manage changes in one place and have them updated everywhere the component is used.

Multiple teams are simultaneously working on different sections of the same page.

Page Collections

You want to reuse an entire page layout and configuration across multiple pages or experiences.

You must build a variation of full pages for different users or workflows.

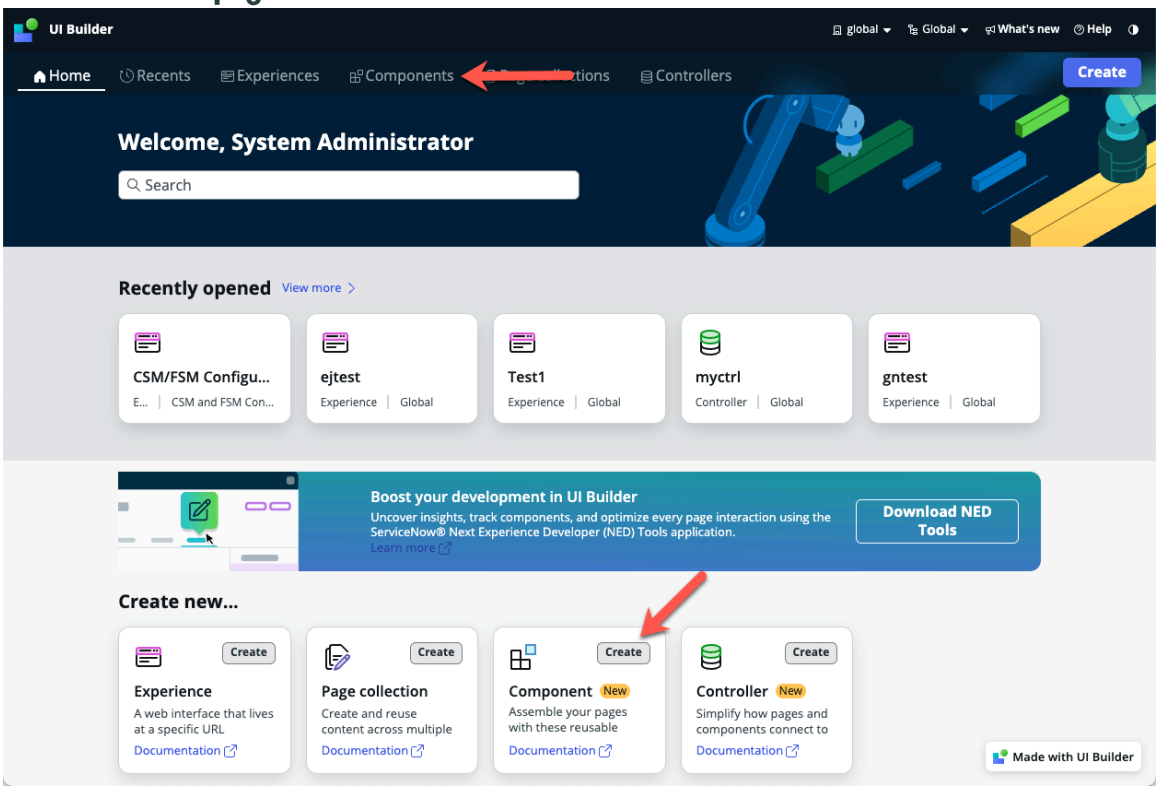
You want to use a set of tabs across multiple pages or experiences.

Custom component UI

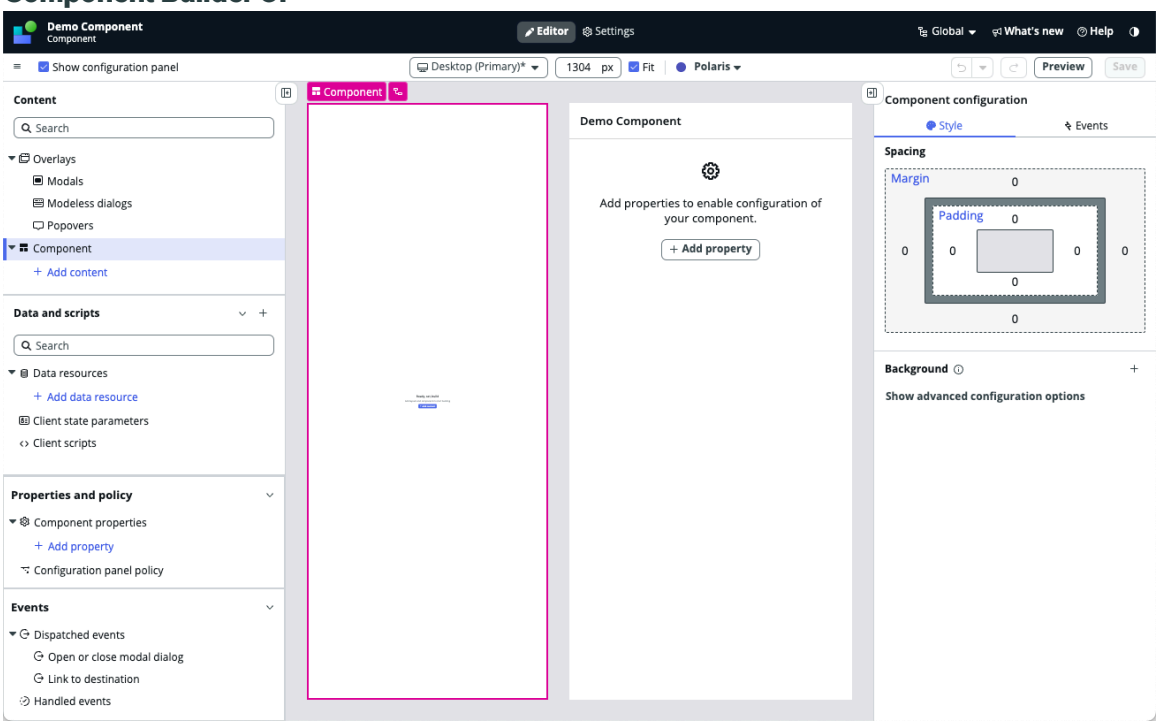
You can access the component builder in UI Builder by selecting **Create** in the header or the **Component** tile on the UI Builder home page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

UI Builder home page



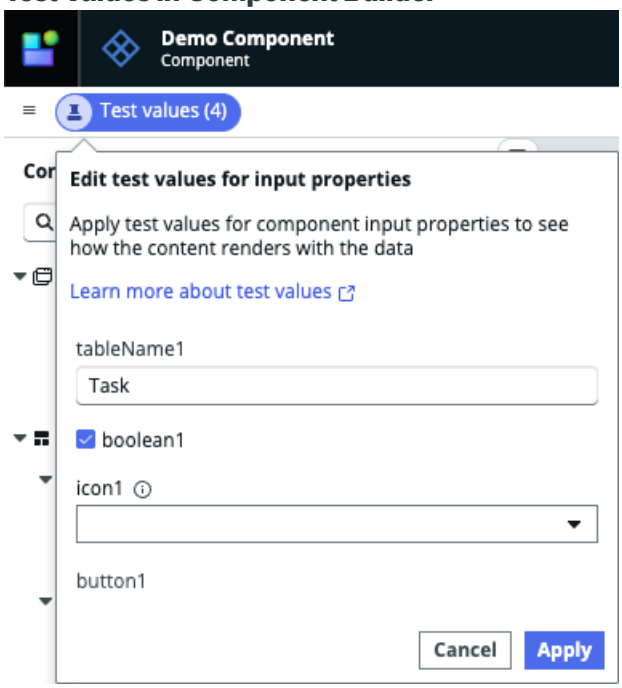
Component Builder UI



Components built with UI Builder can be found in the toolbox when adding a component to a page and in the component list on the home page of UI Builder. You can update or modify custom components by locating it in the components list on the home page of UI Builder.

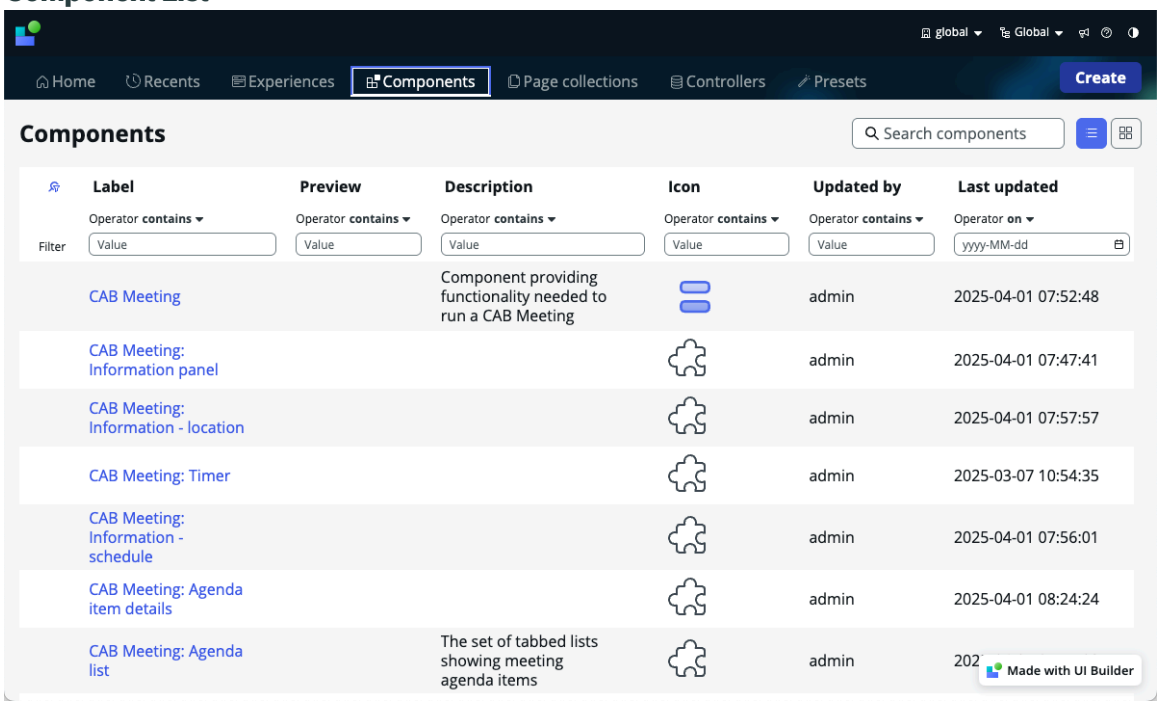
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Test Values in Component Builder



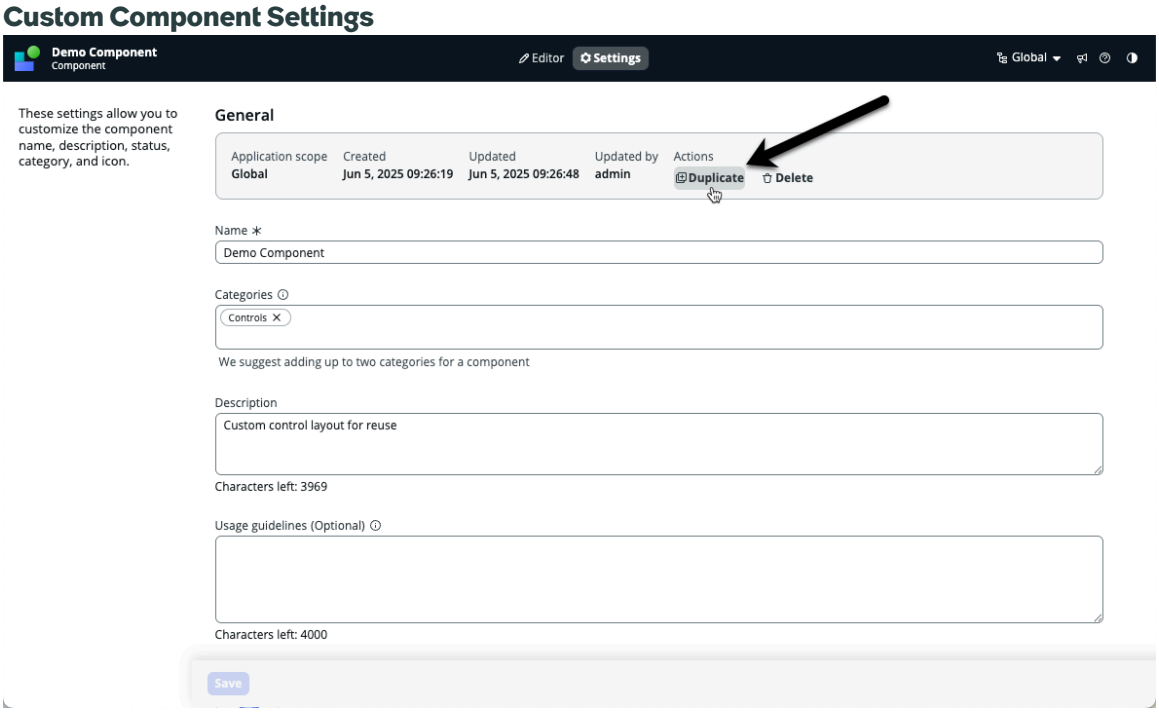
Use test values in component builder to supply simulated values for required and optional URL parameters when building a custom component. Test values help validate how a component will act when added to a page by ensuring bindings and data resources are functioning correctly. For more information about test values see, Test values in a page.

Component List



You can quickly duplicate a custom component from the component settings screen by selecting **Duplicate**, which creates an exact copy of the component for reuse or modification.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Component Builder vs NOW CLI Component Development

There are two ways to build components for UI Builder. The first is using the low-code component within UI Builder, which offers a drag-and-drop interface for creating custom components. The second is using NOW CLI developer tools to build components through code. Both methods produce components that can be added to the UIB toolbox and reused across experiences. Keep in mind that changes to included components may impact both types.

Although both tools create components for UI Builder, there are important differences to consider.

Component Builder in UI Builder:

- Components built within UI Builder can reference controllers and data resources.
- Creates "Macroponent Components," which are stored in the sys_ux_macroponent table.
- Component Builder is ideal for users who prefer a visual, drag-and-drop interface for building components.
- Great for quickly creating simple to moderately complex components.

NOW CLI components:

- Intended for developers who need to write custom HTML, CSS, and JavaScript.
- Suitable for building complex and customizable components.
- Components created with NOW CLI are stored in the sys_uib_toolbox_component table.

Best practices

The UI Builder custom component builder lacks governance capabilities and can lead to duplication and inconsistency in your experience. Teams may create similar components with slight variations, resulting in a fragmented UI, and confusing user experience. It is recommended that your team completes regular audits and cross-team communication to maintain alignment and avoid fragmentation as your experiences grows.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

All components are designed to be upgrade safe, as long as their security policy is set to *read_only*. This provides greater upgrade protection for larger components or page partials compared to other deployable units like bundles and page templates. However, this also means that out-of-the-box (OOTB) components may not be editable.

Create components to reuse across pages

Build reusable custom components to use across experiences and pages in UI Builder.

Before you begin

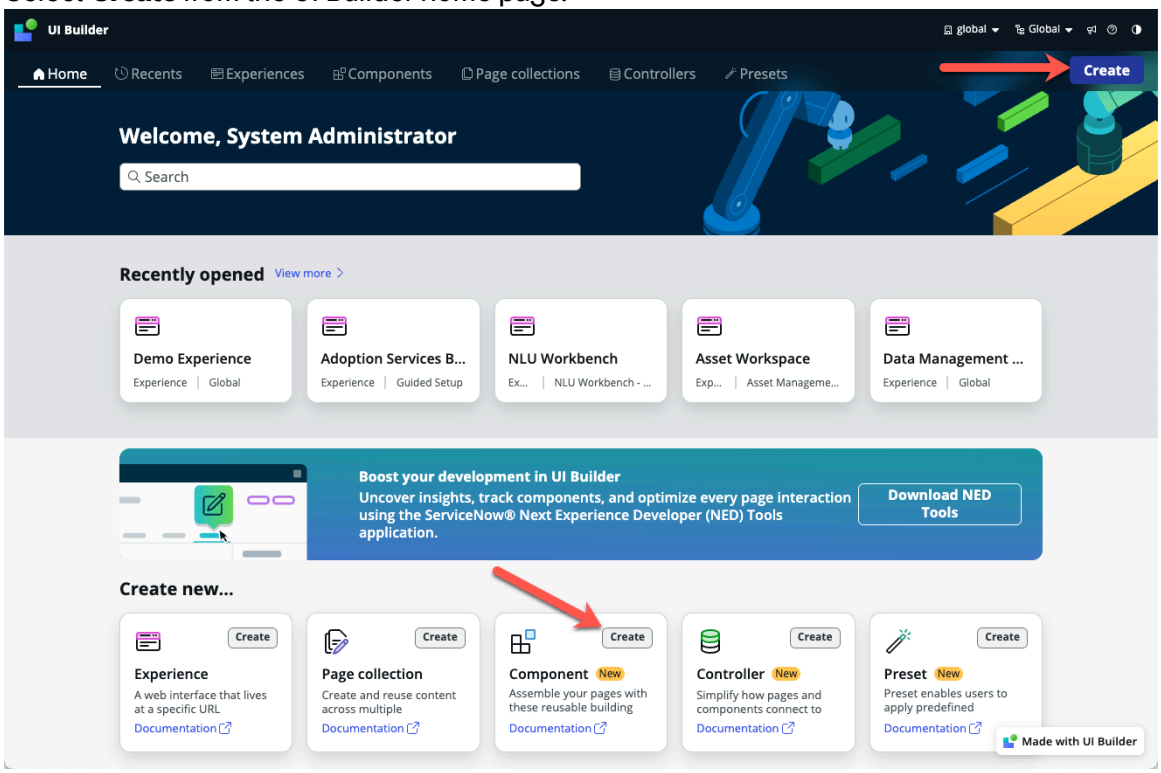
Role required: ui_builder_admin

About this task

In this Component Builder example, we will create a stopwatch component to track elapsed time, which can be added to any page. The component will include customizable properties that can be modified once it's placed onto a page.

Procedure

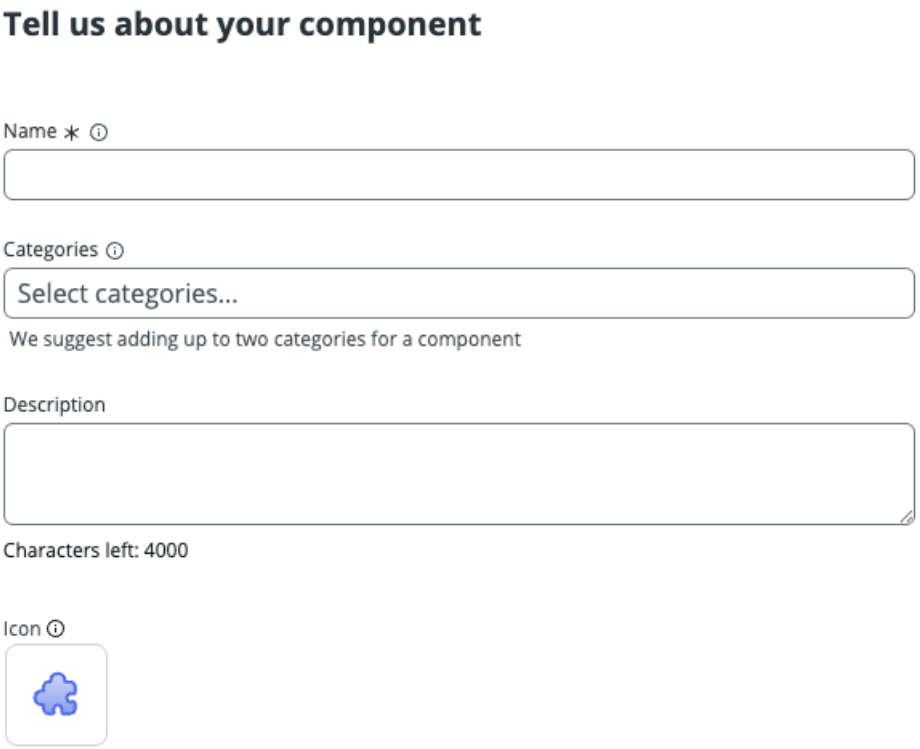
1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Select **Create** from the UI Builder home page.



3. Select **Component**.
4. In the form, fill in the fields.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Create a component dialog box

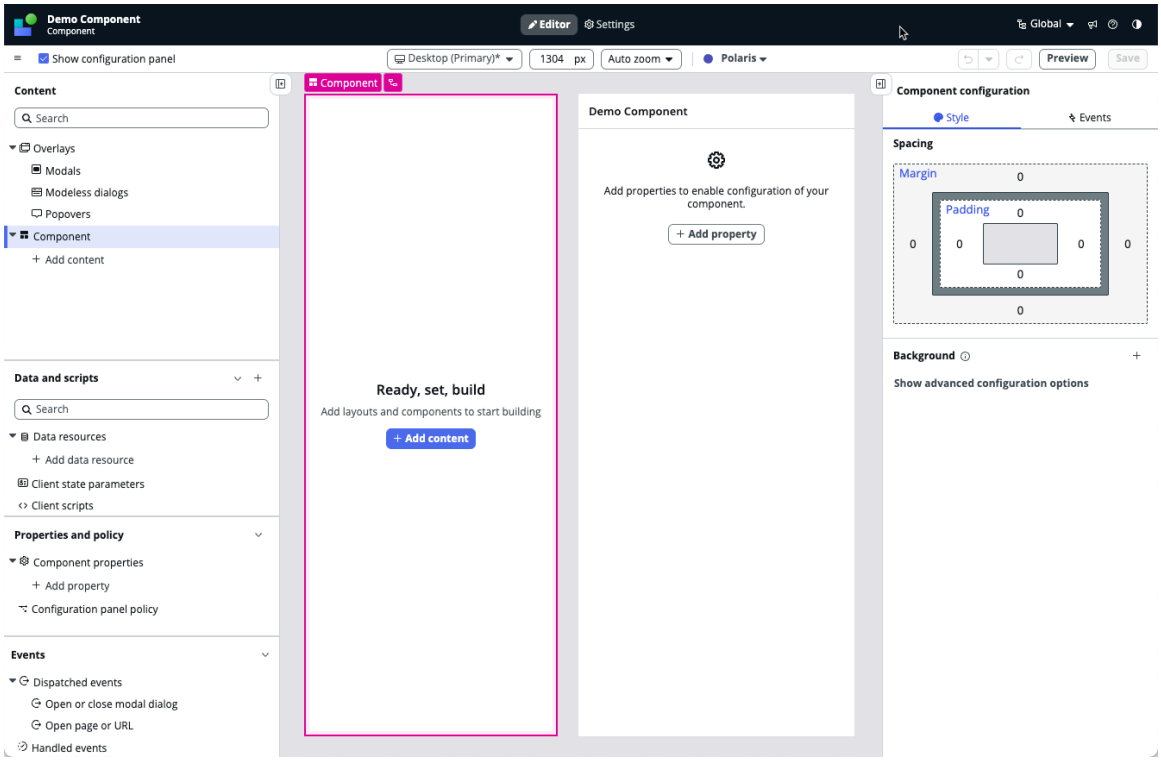


Create a component form

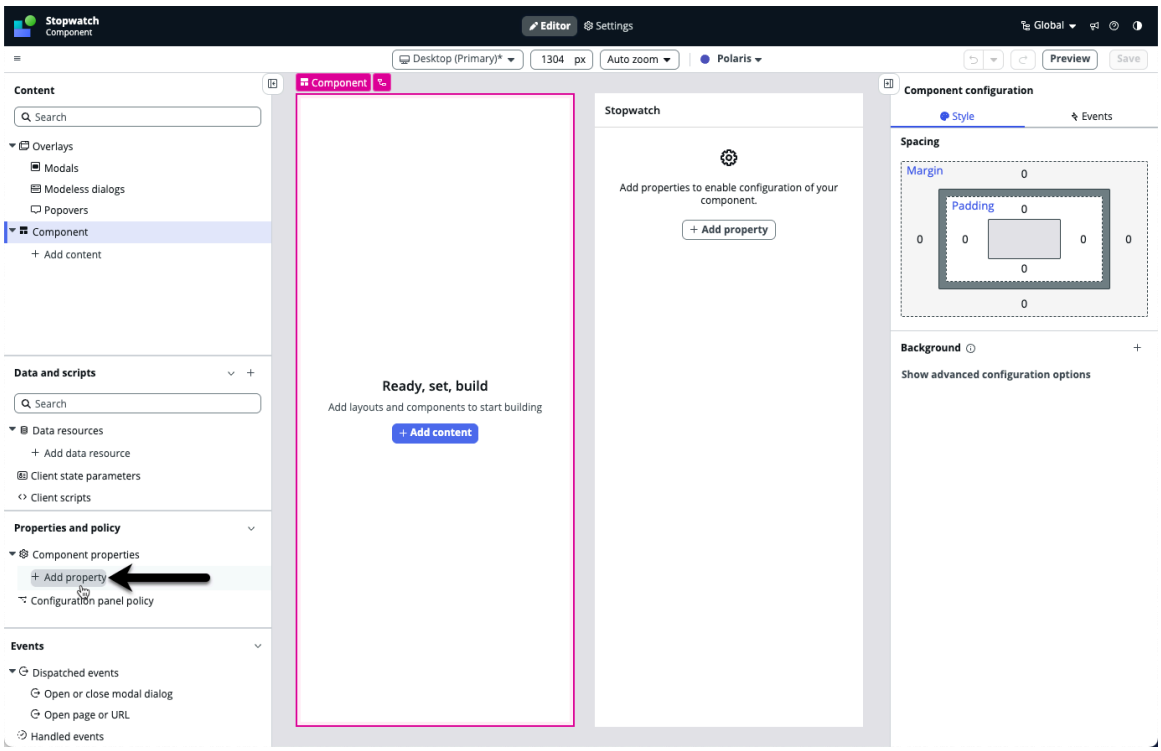
Field	Description
Name	Add a name to track your component internally. The component name appears in the toolbox. For this example, it is `Stopwatch`.
Categories	Add a category to your component to help easily find it in the toolbox. You can update the category later in the Settings. For this example select **Content**.
Description	Add description to describe when and how to use your component.
Icon	Select an icon to represent the component in the toolbox.

5. Select **Create**.
   The page opens in Component Builder view.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



6. Add a custom component property by selecting **+ Add property** in the **Properties and policy** section.



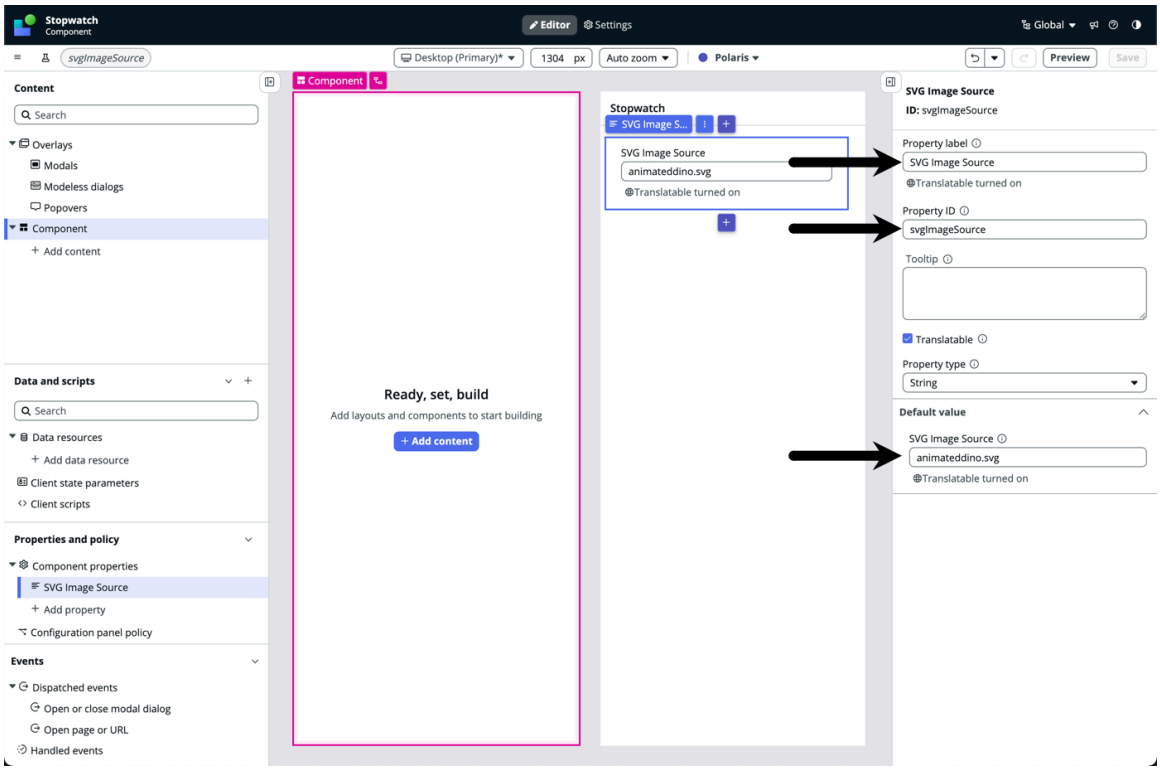
a. Select **String** from the list.

b. Enter the following values in the configuration panel.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Component properties

Field	Value
Property label	`SVG Image Source`
Property ID	`svgImageSource`
Default value	`animateddino.svg`



c. Select **Save**.

7. Configure the layout for the custom component by selecting the **+ Add Content** button.

a. Select **Single column**, then select **Add**.

b. In the content tree, under **Column 1**, select **+Add content**.

c. Select **Card Base Container**, then select **Add**.

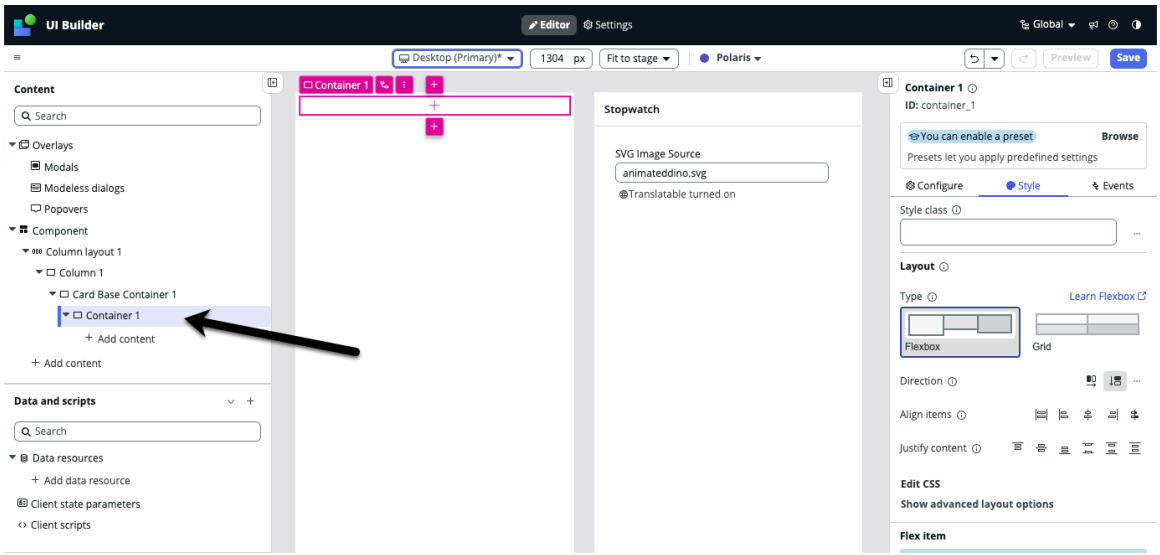
d. In the content tree, under **Card Base Container 1**, select **+ Add content**.

e. Select **Layouts**.

f. Select **Flexbox**, then select **Add**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

g. Select **Save**.

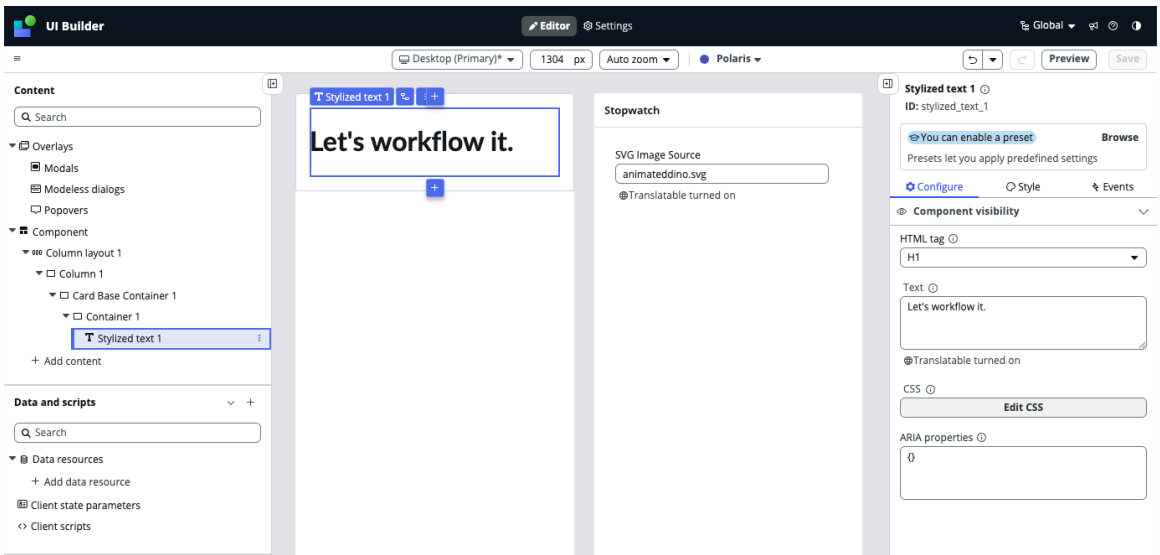


8. Add a stylized text component in our Flexbox container by selecting **+ Add content** under **Container 1**.

a. Select the **Stylized text** component, then select **Add**.

b. Select **Cancel** to close the preset window.

c. Select **Save**.



9. To configure the stylized text component we will add some client state parameters.

a. Select **Client state parameters** in the **Data and scripts** section.

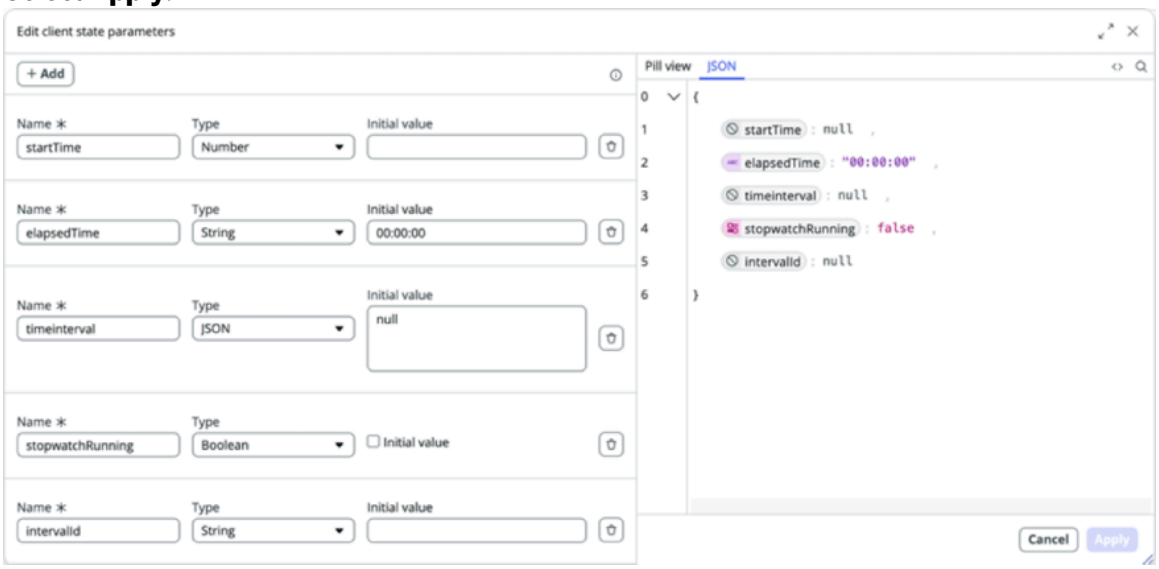
b. In the **Edit client state parameters** dialog, enter the following values:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Client state parameters

Name	Type	Initial value
startTime	Number	
elapsedTime	String	00:00:00
timeinterval	JSON	
stopwatchRunning	Boolean	
intervalId	String	

c. Select **Apply**.



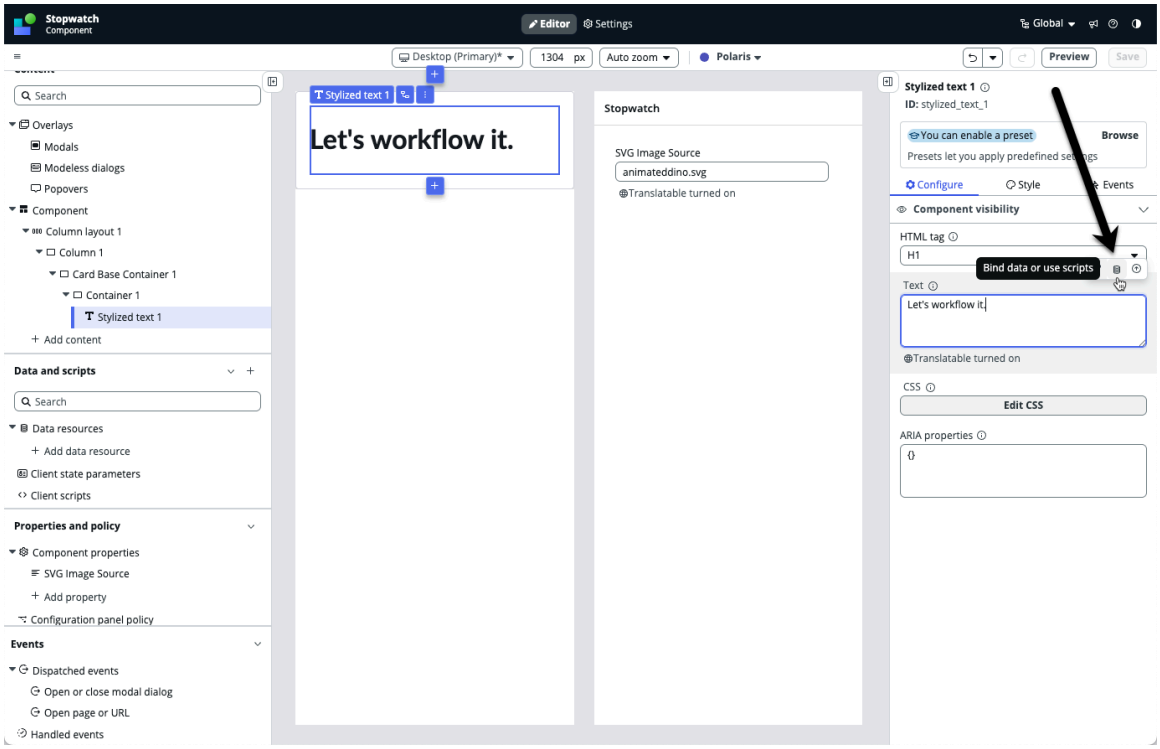
d. Select **Save**.

10. To configure the stylized text component we are going to bind the **elapsedTime** client state parameter.

a. Select the **Stylized text 1** component in the content tree.

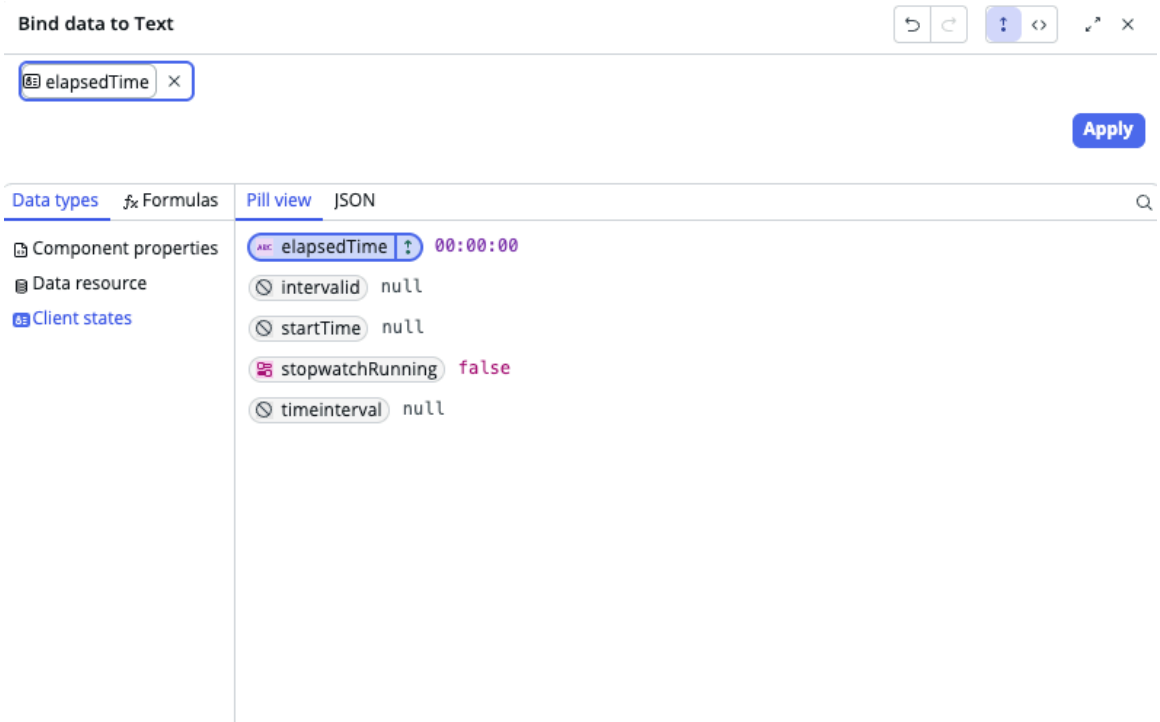
b. Select the bind icon when pointing to the **Text** field of the stylized text component.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



c. Select **Client states**.

d. Double-click the **elapsedTime** pill.



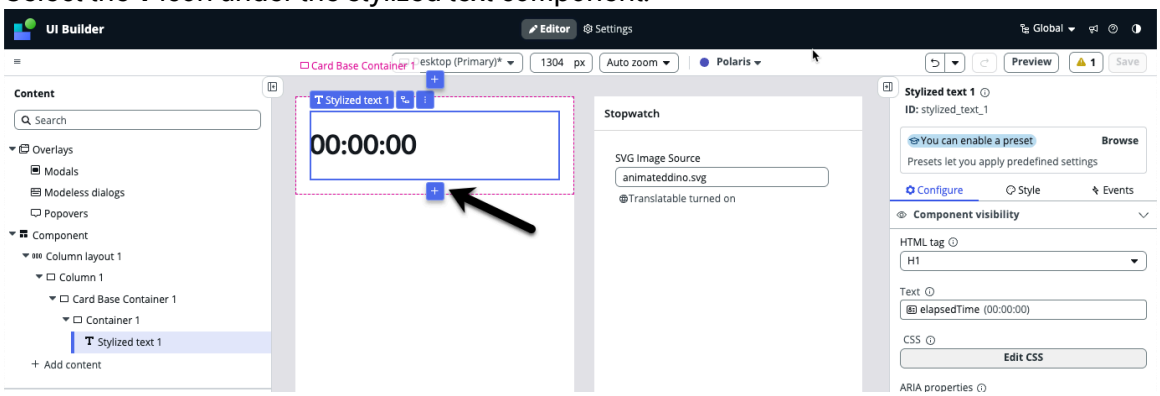
e. Select **Apply**.

f. Select **Save**.

11. Add a Flexbox container to hold the buttons.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. Select the **+** icon under the stylized text component.



b. Select **Layouts**.

c. Select **Flexbox**.

d. Select **Add**.

12. Add and configure the stopwatch running renderer and the start button.

a. Select **+ Add content** in the container you added in the previous step and add a **Conditional renderer**.

b. Select **+ Add condition** in the content tree.

c. Select **Single component**, then select **Next**.

d. Select **Button iconic**, then select **Next**.

e. In the **Edit settings** dialog, enter the following values:

- Component label: `Start`
- Component ID: `start_button`
- Render content: **Always**

f. Select **Apply**.

g. Select the **Start** button in the content tree.

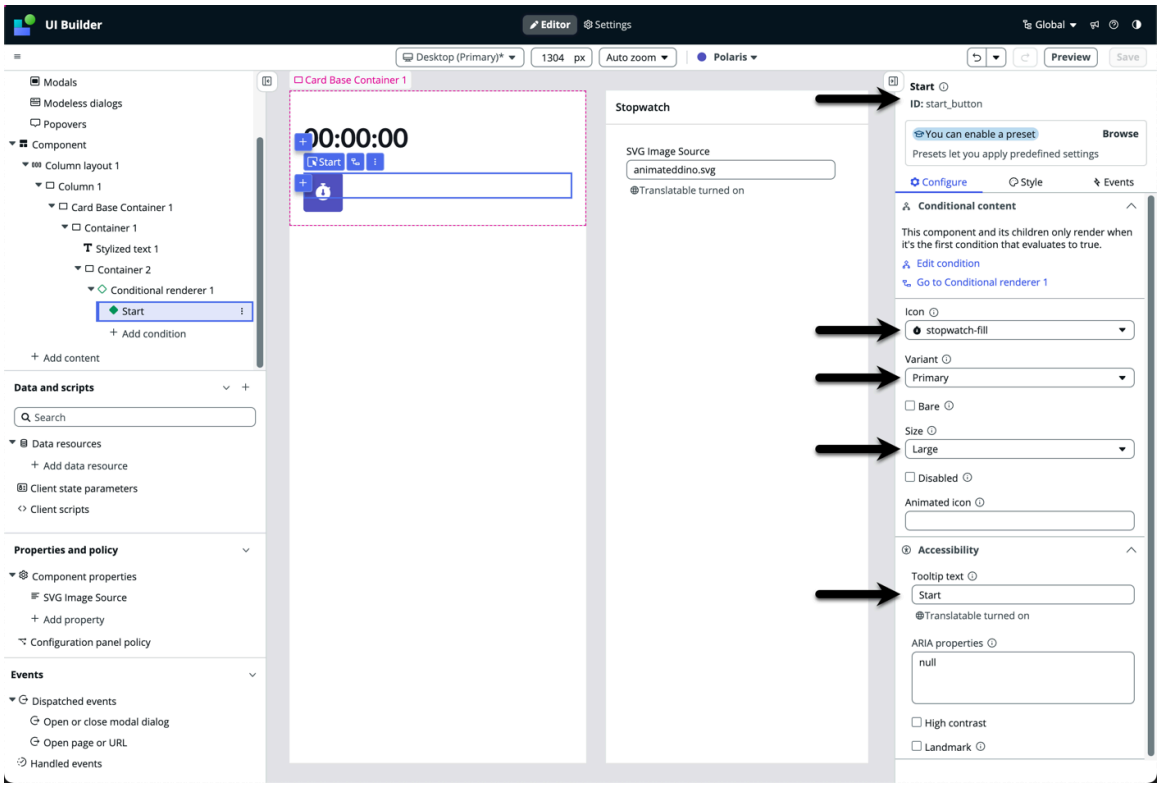
h. Set the following properties in the configuration panel.

Component properties

Field	Value
Icon	**stopwatch-fill**
Variant	**Primary**
Size	**Large**
Tooltip text	`Start`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

i. Select **Save**.



13. Configure the stop button to display while the stopwatch is running.

a. Select **+ Add condition** in the content tree.

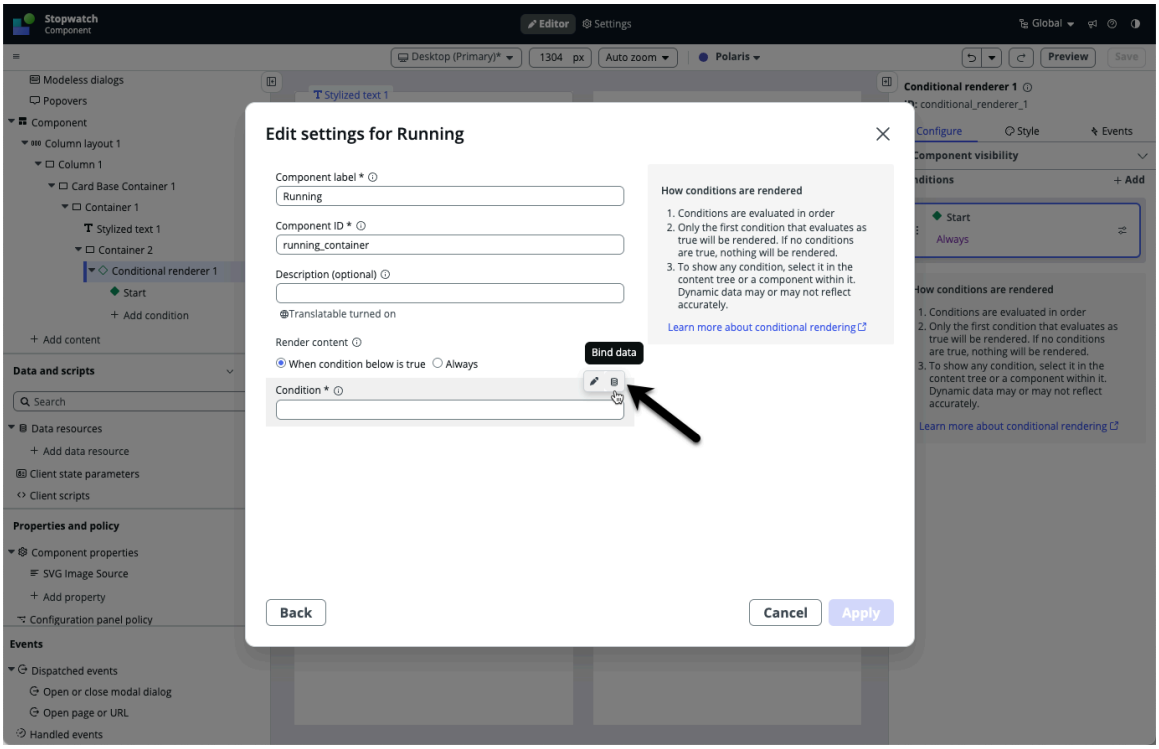
b. Select **Empty container**, then select **Next**.

c. In the **Edit settings** dialog, enter the following values:

- Component label: `Running`
- Component ID: `running_container`
- Render content: **When condition below is true**

d. Select the bind icon while pointing at the **Condition** field.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



e. Select **Client states**, then double-click the **stopwatchRunning** pill.

f. Select **Apply**.

g. Select **Apply**.

h. Select **+ Add content** in the **Running** container and add a **Button iconic**.

i. Select **Button iconic 1** in the configuration panel and enter the following values:

- Component label: `Stop`
- Component ID: `stop_button`

j. Select **Apply**.

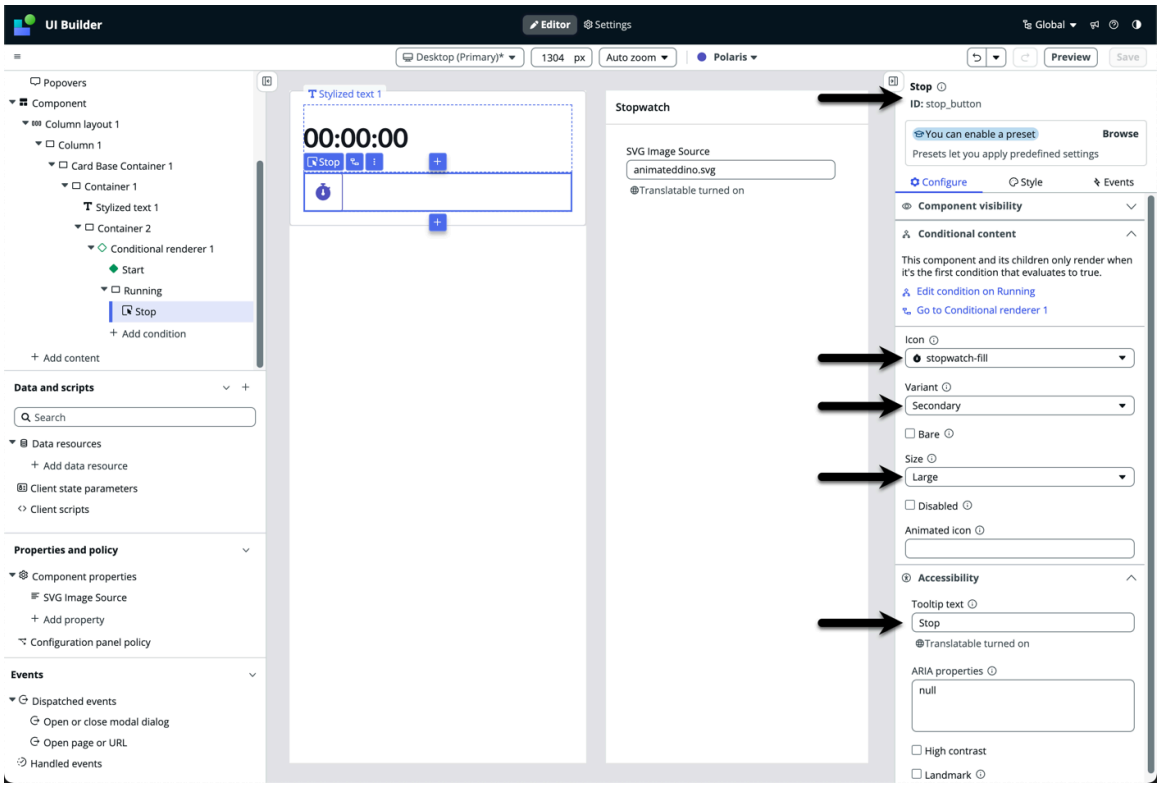
k. Set the following properties in the configuration panel.

Component properties

Field	Value
Icon	**stopwatch-fill**
Variant	**Secondary**
Size	**Large**
Tooltip text	`Stop`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

l. Select **Save**.

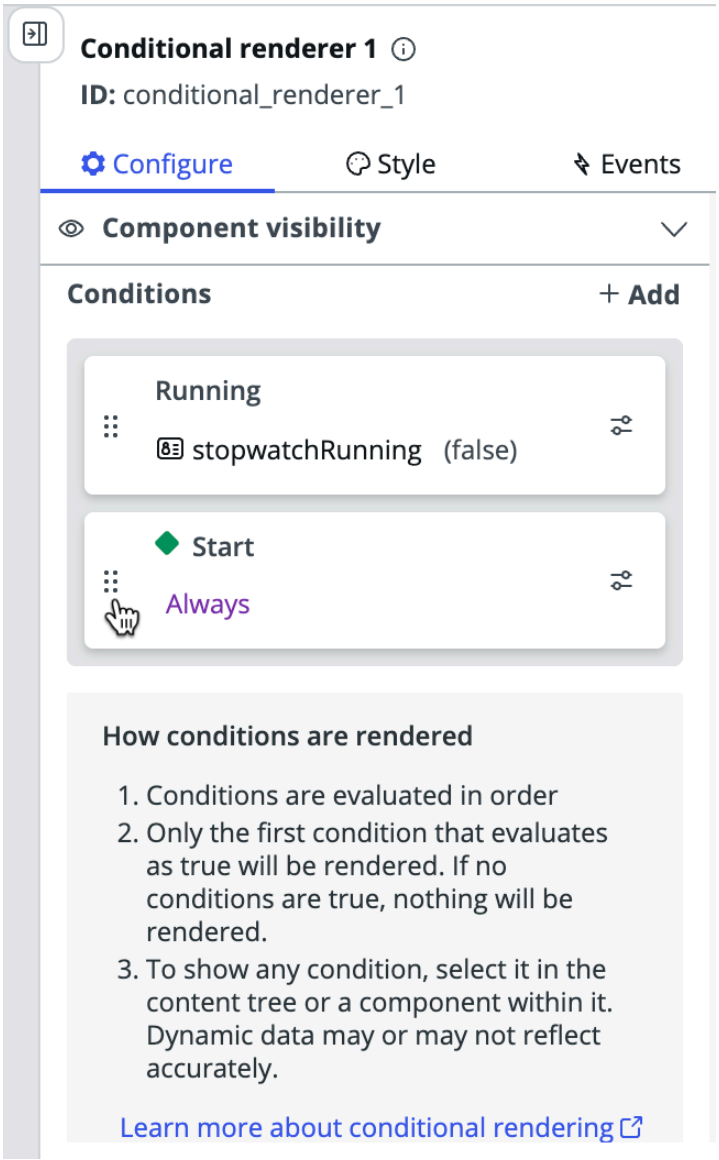


14. Reorder the conditions so that **Running** appears above **Start**.

a. Select **Conditional renderer 1** in the content tree.

b. In the configuration panel, select and drag the drag handle icon ⠿ to move the conditions into position.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Conditions are evaluated in order from top to bottom, so **Running** must appear above **Start** for the buttons to display correctly.

15. Add the logic for the component by selecting **+** next to **Client scripts** in the **Data and scripts** section.

    a. Enter `Start` in the **Script name** field.

    b. Paste the following script into the script field.

```
function handler({ api, helpers }) {
  console.log('Start script running');

  if (api.state.stopwatchRunning === undefined) {
    api.setState('stopwatchRunning', false);
    api.setState('elapsedTime', '00:00:00');
    api.setState('startTime', null);
    api.setState('intervalId', null);
  }

  let running = api.state.stopwatchRunning;
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
 function pad(n) {
   return n < 10 ? '0' + n : n.toString();
 }

 function updateElapsedTime(startTime) {
   if (!running) {
     if (api.state.intervalId) {
       helpers.timing.clearInterval(api.state.intervalId);
       api.setState('intervalId', null);
       console.log('Interval cleared');
     }
     return;
   }

   if (!startTime) {
     console.log('No startTime passed to updateElapsedTime,
cannot update');
     return;
   }

   const now = Date.now();
   const elapsedMs = now - startTime;

   const totalSeconds = Math.floor(elapsedMs / 1000);
   const hours = pad(Math.floor(totalSeconds / 3600));
   const minutes = pad(Math.floor((totalSeconds % 3600) /
60));
   const seconds = pad(totalSeconds % 60);

   const formattedTime = `${hours}:${minutes}:${seconds}`;

   console.log('Setting elapsedTime:', formattedTime);
   api.setState('elapsedTime', formattedTime);
 }

 if (!running) {
   const now = Date.now();

   api.setState('startTime', now);
   api.setState('elapsedTime', '00:00:00');
   api.setState('stopwatchRunning', true);
   running = true;

   if (api.state.intervalId) {
     helpers.timing.clearInterval(api.state.intervalId);
     api.setState('intervalId', null);
     console.log('Existing interval cleared before starting
new one');
   }

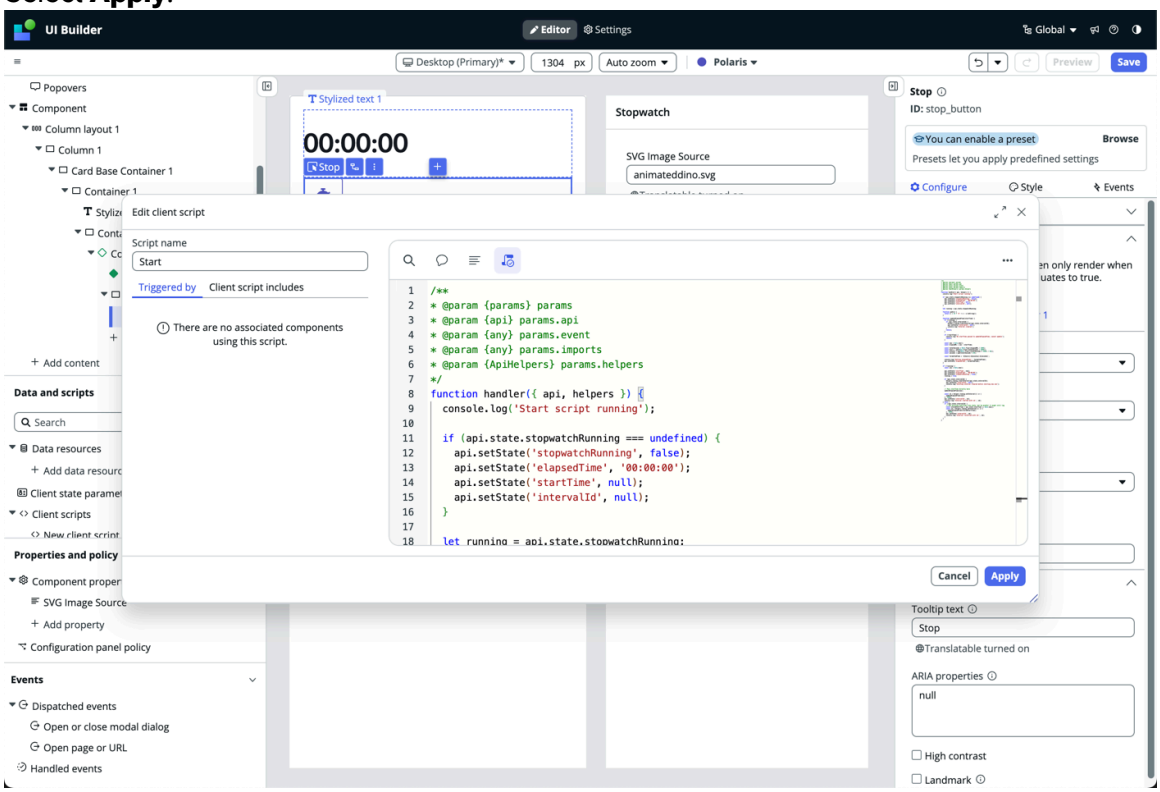
   // Pass startTime directly here
   updateElapsedTime(now);

   const id = helpers.timing.setInterval(() => {
     updateElapsedTime(now);
   }, 1000);
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    api.setState('intervalId', id);
    console.log('Interval started with id:', id);
  } else {
    if (!api.state.intervalId) {
      // Use existing startTime from state, but be mindful it
might still lag
      const storedStartTime = api.state.startTime ||
Date.now();
      const id = helpers.timing.setInterval(() => {
        updateElapsedTime(storedStartTime);
      }, 1000);
      api.setState('intervalId', id);
      console.log('Interval restarted with id:', id);
    }
  }
}
```

c. Select **Apply**.



d. Select **+** next to **Client scripts** to add a second script.

e. Enter `Stop` in the **Script name** field.

f. Paste the following script into the script field.

```
function handler({ api, helpers }) {
  console.log('Stop script running');

  if (api.state.stopwatchRunning === undefined) {
    // If state is not initialized yet, initialize it to
avoid errors
    api.setState('stopwatchRunning', false);
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
    api.setState('elapsedTime', '00:00:00');
    api.setState('startTime', 0);
    api.setState('intervalId', null);
  }

  if (api.state.stopwatchRunning) {
    api.setState('stopwatchRunning', false);
    if (api.state.intervalId) {
      helpers.timing.clearInterval(api.state.intervalId);
      api.setState('intervalId', null);
    }
  }
}
```

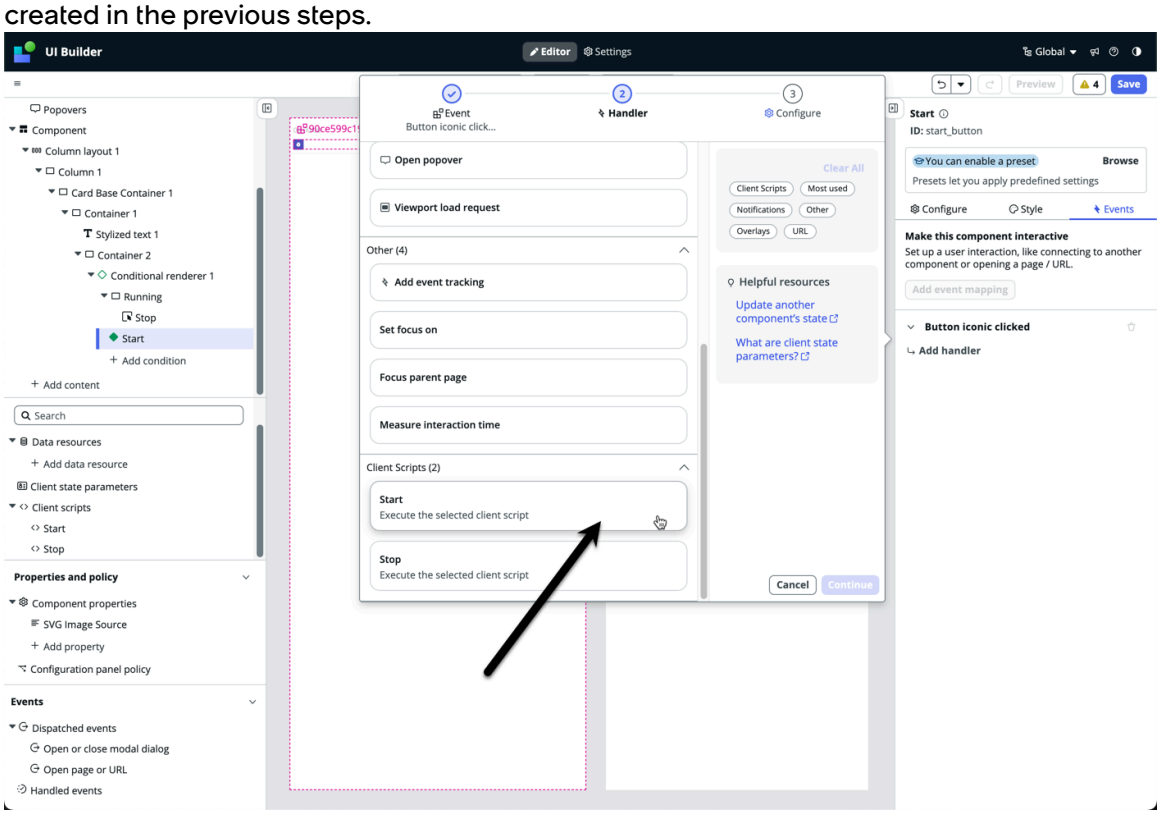
g. Select **Apply**.

h. Select **Save**.

16. Add event handlers to the buttons we created in the previous steps.

a. Select the **Start** button in the content tree and open the **Events** tab in the configuration panel.

b. Select **Add handler** under **Button iconic clicked**, then select the **Start** client script we created in the previous steps.



c. Select **Continue**, then select **Add**.

d. Select the **Stop** button in the content tree and open the **Events** tab in the configuration panel.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

e. Select **Add handler**, then select the **Stop** client script we created in the previous steps.

f. Select **Continue**, then select **Add**.

g. Select **Save**.

17. Select **Preview** to test the configured components.



Result
Your custom component is now available in the UI Builder toolbox.

Enable configuration of components with inherited controllers in component builder

Learn how components can inherit page resources.

Using component builder, developers add data resources in two ways: directly within the component or through inheritance from the page.

Direct configuration

When you add data resources directly to a component, they remain contained within that component. These internal data resources are not visible or accessible when the component is placed on a page.

Inheritance

You can configure a component to inherit data resources from the page where it's placed. When inheritance is enabled, the component scans for data resources of the same type and automatically connects to them. For example, a component configured to inherit form controllers will look for form controllers on the page. This allows the component to be used across different page contexts and leverage existing data resources without manual configuration each time.

When you place a component configured for inheritance on a page, the inheritance behavior works as follows:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Matching resources found	Behavior
No match found	Creates a new resource of the same type using the component's initial configuration
One	Connects to that resource
Multiple	Connects to the first instance of that resource

Enable configuration of components with inherited controllers

Configure components to automatically inherit controllers and data resources when placed on pages.

Before you begin

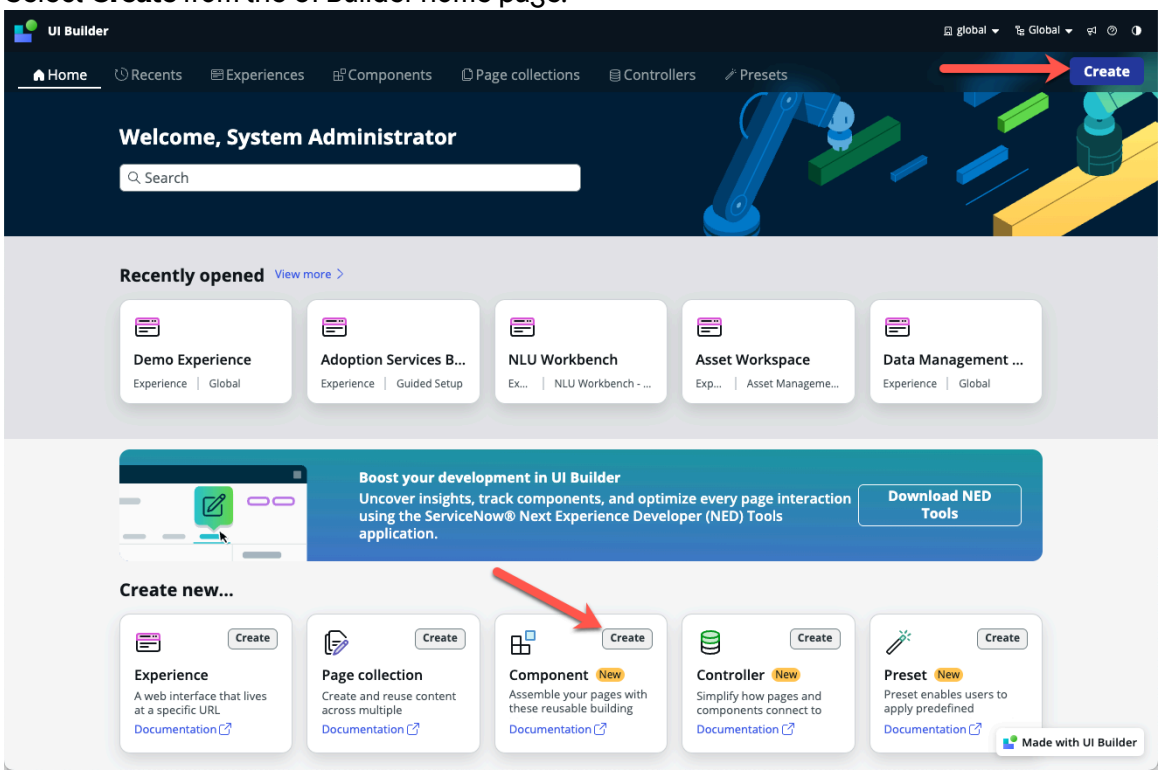
Role required: ui_builder_admin

About this task

In this Component Builder example, we will create a text component and configure it to inherit the List controller. We will then place it on an list page where the component connects to the List controller and displays the appropriate table name.

Procedure

1. Navigate to **All** > **Now Experience Framework** > **UI Builder**.
2. Select **Create** from the UI Builder home page.



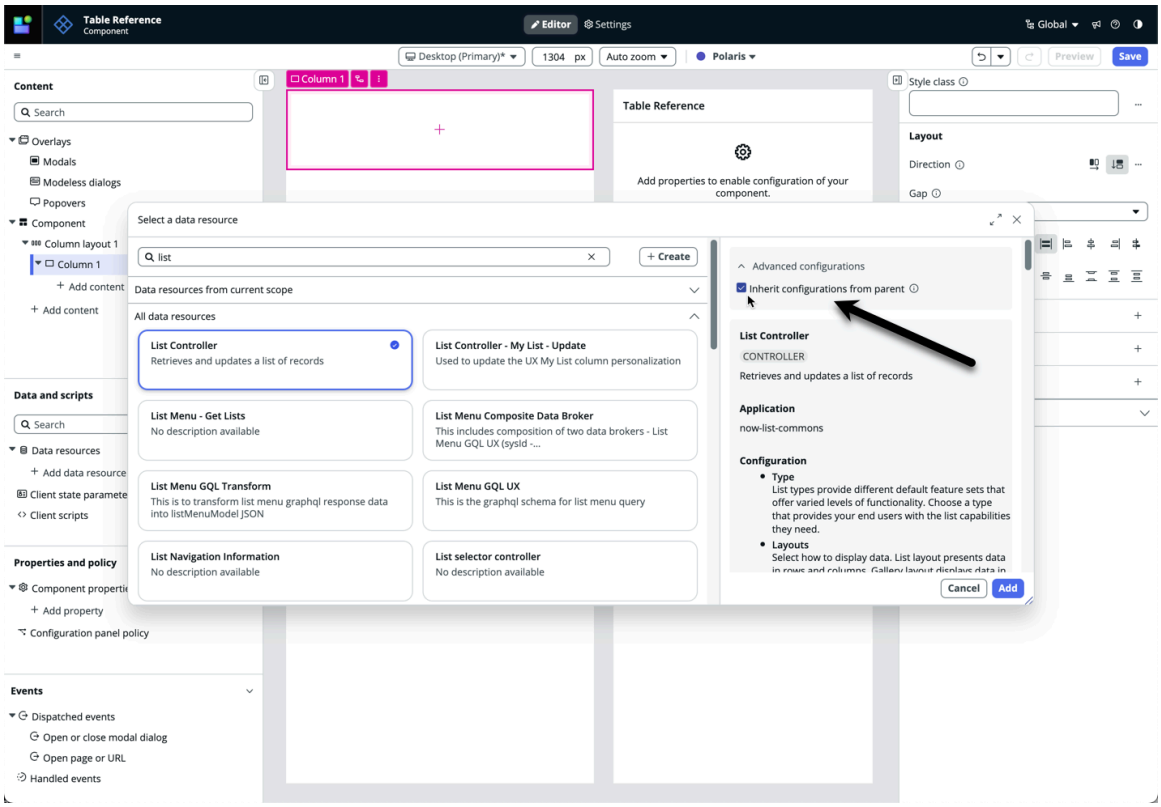
3. Select **Component**.
4. In the form, enter the following values:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Component properties

Field	Value
Name	`Table Reference`
Categories	**Content**
Description	`Text component that displays the name of the current page's table`
Icon	(Default)

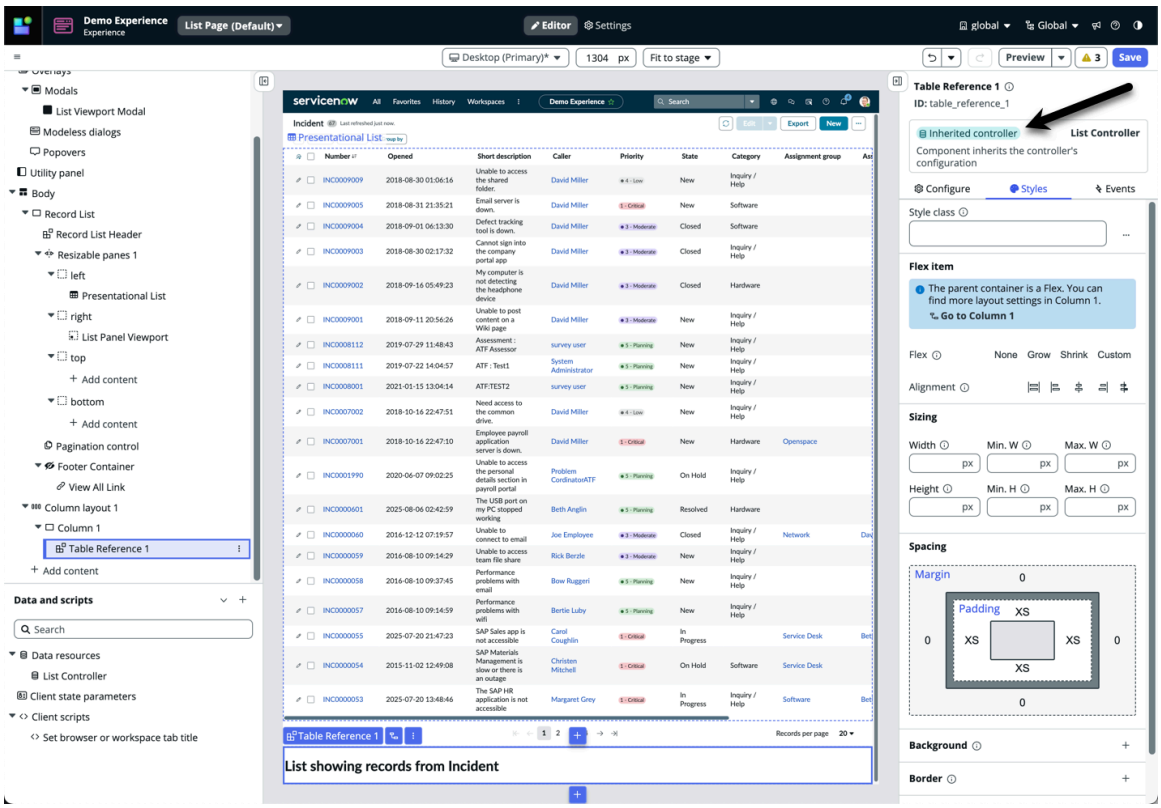
5. Select **Create**.
6. Add a data resource to the component.
   a. In the data resource drawer, select **+ Add data resource**.
   b. On the left, select **List Controller**.
   c. On the right, select **Advanced configurations** to expand the section.
   d. Select the **Inherit configurations from parent** option to enable the property.
   e. Select **Add**.



7. Create the component.
   a. In the content tree, select **+ Add content**.
   b. Select **Single column**, then select **Add**.
   c. In the content tree, select **+ Add content** under **Column 1**.
   d. Select a **Stylized text**, then select **Add**.
   e. Select **Cancel** to close the preset window.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

8. Configure the component to use the List controller.
   a. In the configuration panel, hover over **Text** and select the bind data icon.
   b. On the left, select the **Formulas** tab, then double-click **CONCAT**.
   c. In the upper section, double-click **value1** to edit the field and enter `"List showing records from "`.
   d. Double-click **values** to edit the field and enter `@data.list_controller_1.tableLabel`.
   e. Select **Apply**.
9. Select **Save** to save your work.
   Your custom component is now available in the toolbox and ready for use in a page.
10. Place the component on a list page to see it automatically inherit the List controller and display the table name.



Application Runtime Policy (ARP)

Application Runtime Policy (ARP) is an opt-in, Zero Trust security framework for app creation and runtime that enforces fine-grained controls over accessing network, scripting, record, and platform resources. ARP provides automatic policy creation for applications in development, which helps simplify policy creation and the ServiceNow Store certification process.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Get started

Explore



Application Runtime Policy (ARP) is an opt-in zero-trust security framework for custom applications in development and runtime that enforces fine-grained controls over what resources an application can access at runtime.

Configure



Configure Application Runtime Policy to automatically generate policies, which can either be automatically approved or require review and manual approval.

Use



Application Runtime Policy (ARP) generates policies to control what external resources and cross-scope privileges an application can access at runtime. After opting in, manage policy records, configure resource quotas, and automatically track cross origin and cross scope access during development.

Reference



Reference topics provide additional information about forms and options available through Application Runtime Policy (ARP).

Helpful resources

Some ServiceNow resources that can provide you with helpful information are:

ServiceNow Community

ServiceNow Community

Developer

developer.servicenow.com

Impact

http://impact.servicenow.com

ServiceNow University

ServiceNow University

Best Practices

Best Practices

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Partner

https://www.servicenow.com/partners.html

ServiceNow

http://servicenow.com

ServiceNow Store

http://servicenow.com

Support

- https://support.servicenow.com/now
- Known Error Portal

Exploring Application Runtime Policy

Application Runtime Policy (ARP) is an opt-in zero-trust security framework for custom applications that tracks and enforces fine-grained controls over what resources an application can access at runtime.

Application Runtime Policy (ARP) overview

Application Runtime Policy (ARP) is an opt-in, zero-trust security framework for custom app creation and runtime. It enforces fine-grained controls over scripting, record, platform, and cross-scope or cross-origin network resources.

When ARP is activated for an application in development, network and cross-scope access policies are automatically tracked and recorded in the application's policy metadata as you develop application functions. The generated policies can be automatically approved or can require review and manual approval, depending on how ARP is configured for the application.

Activating ARP also generates one default quota configurations record for the application in development.

ARP users

Users

User	Description
Custom application developers	Developers who build custom applications on the ServiceNow AI Platform. They opt in to ARP during development to define and review the resource access policies their application requires before making the application available to other instances in their organization.
ServiceNow Store partner developers	Partner developers who develop and publish applications to the ServiceNow Store. ARP gives ServiceNow Store partners a structured way to declare and certify their application's runtime resource access, which reduces the time required for certification review.

ARP workflow

The following steps describe the end-to-end ARP workflow from application development to installation on a different ServiceNow AI Platform instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

1. Opt in to ARP at development time from the custom application [sys_app] form by setting the **Application Runtime Policy** field to **Tracking** or **Enforcing**.
2. As you develop application functionality, ARP tracks client and server-side cross origin network requests and automatically creates policy records in the application's metadata.
   - In Tracking mode, policies automatically have the **Allowed** status.
   - In Enforcing mode, policies are created with a status of **Requested** and the access is blocked until you approve the policy.
   - The first instance of each violation is recorded in the Runtime Policy Violation Log, regardless of whether ARP is activated in Tracking or Enforcing mode.
3. If using the Enforcing mode, review each policy record and sets its status to **Allowed** to allow access, or **Denied** to block it.
4. When the application is ready for publication, the approved policies are packaged and published with the application to your organization's Application Repository or to the ServiceNow Store.

   For more information about publishing to the Application Repository, see Publish an application to the application repository. For more information about publishing an application to the ServiceNow Store, see Publish an application to the ServiceNow Store.

   If you publish to the ServiceNow Store, the Runtime Policy summary displays the information that will be included on the application's listing details.

5. When the application is installed to a new instance, ARP enforces policies that are configured to permit access. Undeclared cross-origin and cross-scope access is rejected and logged.

> **Note:** After an application built with ARP is installed, its cross-scope privileges can be modified and the update can be published to the Application Repository as an application customization. For more information about modifications to cross-scope privileges, see Cross-scope privilege record. For more information about customizing applications, see Manage customizations to applications.

ARP benefits

ARP benefits

Benefit	Feature	Users
Limits each application to its own scope by default, so unintended access to external hosts or other application APIs must be explicitly declared and approved.	Network policies, Cross-Scope Privilege	Custom application developers, ServiceNow Store partner developers
Automatically creates and approves policies in Tracking mode, so developers can build out application functionality without stopping to manually configure each access requirement.	Tracking mode	Custom application developers, ServiceNow Store partner developers
Prevents a single application from consuming excessive instance resources by defining quota limits for API transactions, UI transactions, and scheduled jobs.	Quota configurations	Custom application developers, ServiceNow Store partner developers
Reduces the overhead of granular policy creation by letting developers define broad	Segment policies	Custom application developers,

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ARP benefits (continued)

Benefit	Feature	Users
access patterns for an entire pillar using a single segment policy record.		ServiceNow Store partner developers
Enables automatic ServiceNow Store certification by conforming to variable runtime guarantees about application behavior on ServiceNow AI Platform instances	ARP policy enforcement	ServiceNow Store partner developers
Provides application-level cross-scope and cross-origin access information to potential customers before they obtain or install the application, improving transparency.	ServiceNow Store application listing details, Application Manager application details	ServiceNow AI Platform administrators

What to explore next

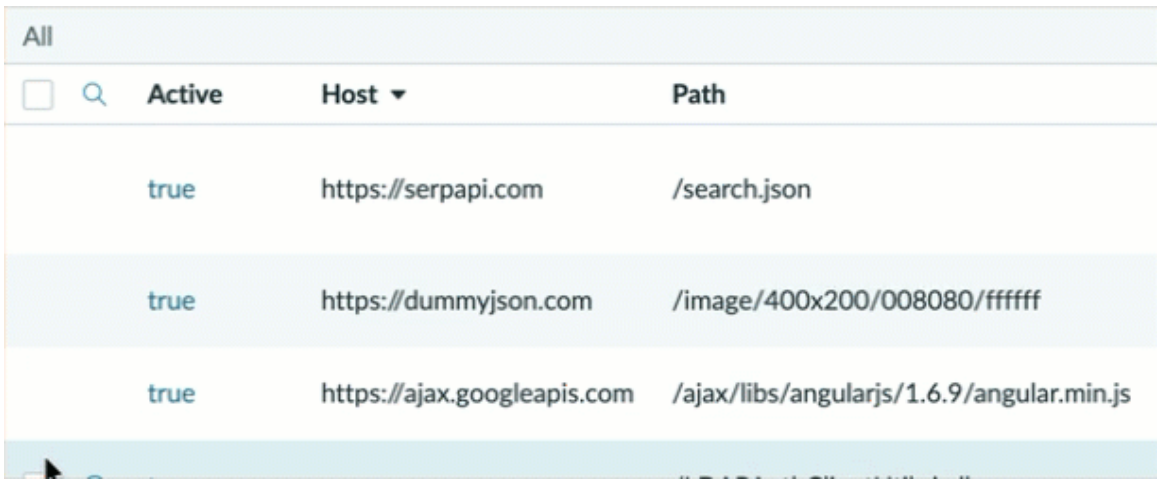
To learn more about configuring and using Application Runtime Policy, see:

- Configuring Application Runtime Policy
- Using Application Runtime Policy
- Application Runtime Policy reference

Network policies

Network policies allow list the external hosts and endpoints that an application is permitted to access. Application Runtime Policy (ARP) generates network policies automatically when your application attempts to connect to external networks during development.

Some application functionality might require access to an external network to use a third-party API or content delivery URL. When an application's client-side or server-side code attempts to connect to an external network, Application Runtime Policy (ARP) tracks the access and generates a policy in the Network Policies table [sys_arp_network_policy]. To navigate to a specific policy record from the network policies table, hover over a policy and select the preview icon that appears.



The Network Policy table displays the following information:

Active

A true|false value that shows whether each network policy is enabled.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Host

Displays the external host that the application attempted to access when the policy was generated.

Path

Resource paths or endpoints within the external network that the application attempted to access when the policy was generated. This might include API endpoints, image paths, or script locations. For Inbound Cross Scope Requests policy types, either a path or resource can be included (not both).

Policy Type

Specifies the type of external network access the policy governs. Policy type describes both the direction of the request and whether it originates from client-side or server-side code.

- **Client Cross-origin Request** is a client-side request to an external host, such as a fetch or XHR call made from a UI page script.
- **Client Cross-origin Scripts** is a client-side request to load an external script resource.
- **Inbound Cross Scope Request** is a fetch or XHR request originating from the application as it runs in the browser, and attempting to access cross scope REST APIs, processors, or ServiceNow AI Platform APIs.
- **Inbound Global Request** is a global API request with a fetch call that doesn't follow the ARP governance model (for example, websocket connections).
- **Server Outbound Request** is a server-side outbound call to an external host, such as a call made from a script include.

Resource

Identifies the specific platform processor and sub-processor involved in an external request, in `processor.subprocessor[.context]` format. For Inbound Cross Scope Request policy types, either a path or a resource can be included (not both).

Scheme

Specifies the network protocol used when the application accesses the external host, if applicable.

Short Description

A description of the policy. When using ARP, "Auto-created from violation" is the default, but can be changed.

Status

Determines whether the application can access external networks as defined in the network policy. While an application is still in development, you can change the state as needed.

- **Allowed** is the default policy state when ARP is in tracking mode. It enables external network access as defined in the network policy.
- **Requested** is the default policy state when ARP is in enforcing mode. It prevents external network access as defined in the network policy and indicates that you haven't approved or denied the policy.
- **Denied** state prevents external network access as defined in the network policy.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Segment policies

Segment policies enable broad access patterns across one or more Application Runtime Policy segment without requiring granular policy records for each access. If your application requires segment policies, you must create them manually.

Segment policies provide a declarative way to enable broad categories of access. For example, rather than creating individual policy records for every external network call, you can create a single segment policy. This segment policy covers a certain type of access that will be allowed without a granular policy record.

When you activate a segment policy with one or more wildcard policies, ARP doesn't create individual policy records for access attempts covered by those wildcard policies. Access is granted without requiring per-resource approval.

Because they provide such broad resource access, only use segment policies when out-of-scope resources called at runtime can't be known during application development. For more information about activating segment policies, see Create a segment policy.

Cross scope privilege

Cross-scope privilege records define rules for access to application resources outside your application's scope. Application Runtime Policy generates cross scope privilege records automatically when your application attempts to access resources beyond its own scope.

When an application attempts to access resources that belong to a different application scope, Application Runtime Policy (ARP) generates a new policy within the Cross Scope Privilege table. ARP's cross scope access enhances any existing cross scope privileges, so all related functionalities (for example, Restricted Caller Access) still operate without interference. To navigate to a specific policy record from the Cross Scope Privilege table, hover over a policy and select the preview icon that appears.

The Cross Scope Privilege table displays the following information:

Source Scope

The application requesting runtime access to another application's resources.

Target Scope

The application whose resources are being requested.

Target Name

The name of the table, script include, or script object being requested.

Operation

The operation the script performs on the target. The target type determines the available operations. Tables support the read, write, create, and delete operations. Script includes and script objects only support the execute API operation.

Status

Determines whether the application can access resources from different application scopes as defined in the cross scope privilege record. While an application is still in development, you can change the state as needed.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- **Allowed** is the default policy state when ARP is in tracking mode. It enables access to resources in different application scopes as defined in the network policy.
- **Requested** is the default policy state when ARP is in enforcing mode. It prevents access to resources in different application scopes as defined in the cross scope privilege record and indicates that you haven't approved or denied the privilege.
- **Denied** state prevents access to resources in different application scopes as defined in the cross scope privilege record.

Application

The name of the application that attempted to access resources outside its own scope.

Violation Logs

The Runtime Policy Violation Log records network, record, and scripting violations encountered while exercising application functions, giving you a consolidated view of cross-origin or out-of-scope access attempts. The ARL Violation Log records violations of event handlers, scheduled jobs, and API and UI transactions.

Runtime Policy Violation Log

When Application Runtime Policy (ARP) is enabled in Tracking or Enforcing mode, the first instance of every network, record, and scripting policy violation is recorded in the Runtime Policy Violation Log. You can review the log by navigating to **All** > **System Policy** > **Application Runtime Policy** > **Runtime Policy Violation Log**. If access attempts that you want to allow are blocked while developing an application, you can use the log to find the policy record that needs to be reviewed and approved.

ARL Violation Log

After the application is installed, platform resource usage violations are recorded at runtime. The ARL Violation Log includes violation records related to API and UI transactions, event handlers, and scheduled jobs. Administrators can review the log by navigating to **All** > **System Policy** > **Application Runtime Policy** > **ARL Violation Log** to determine whether to manually create policies on their instance for any violations.

Quota configurations

Quota configurations define limits on the resources that an application can use at runtime. Application Runtime Policy uses these limits to prevent a single application from consuming excessive platform resources.

Quota configurations enforce limits on how many resources your application can use, including API transactions, UI transactions, scheduled jobs, and event handlers. Activating Application Runtime Policy (ARP) automatically configures a set of default quota limits for an application that's in development. For information about the default values for automatically generated quota configurations, see Default Quota Configuration form.

You can change quota limits from the default in the automatically generated record if needed by updating the Default Quota Configuration form.

If an application exceeds the default quota configurations, requests beyond that limit are rejected, deferred, or logged based on the selected default quota configuration mode.

The Default Quota Configurations table displays the following information:

Configuration name

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

A unique name used to identify the configuration.

Application Condition

Additional conditions that determine the scope for the default quota configurations.

Enable Auto-Generated Quotas

When true, enables automatic generation of application resource quotas in instances that install the application.

Order

Dictates the order in which application quotas are evaluated. If multiple default quota configurations apply to an application, the default quota configuration with the lowest number is applied first.

API Transaction Quota %

Total percentage of threads that can execute API requests for a given application and scope. When the default quota configuration is in enforcing mode and the quota threshold is exceeded, API requests are rejected.

Default value: 30%.

Event Handler Quota %

Percentage of total processing time that can be allocated to event handlers. When the default quota configuration is in enforcing mode and the quota threshold is exceeded, event handler processing is deferred until the next time window.

Default value: 20%.

Interactive Transaction Quota %

Total percentage of threads that can execute UI requests. When the default quota configuration is in enforcing mode and the quota threshold is exceeded, UI requests are rejected.

Default value: 30%.

Scheduled Job Quota %

Percentage of total processing time that can be allocated to jobs. When the default quota configuration is in enforcing mode and the quota threshold is exceeded, scheduled jobs are deferred until the next time window.

Default value: 20%.

Mode

Determines how default quota configurations are enforced.

- **Disabled** prevents enforcements of a default quota configuration.
- **Enforced** causes requested resources to be deferred or rejected when the quota threshold is exceeded.
- **Log Only** adds log entries to the Runtime Policy Violation Log to indicate when the quota threshold is exceeded. No resources are rejected or deferred.

Configuring Application Runtime Policy

Configure Application Runtime Policy to automatically approve generated policies, or to require review and manual approval.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Configuration overview

Application Runtime Policy (ARP) is free and available by default on all production and non-production instances. You don't have to activate a plugin.

When you begin developing an application, ARP is inactive by default. To enable automatic policy generation, you must set ARP to either tracking or enforcing mode from the custom application form.

Opt in to Application Runtime Policy

Enable Application Runtime Policy for an application that's in development by setting the policy mode on the custom application form.

Before you begin

Role required: arp_admin or admin.

About this task

Application Runtime Policy (ARP) is an opt-in feature. It must be activated for each custom application by setting the **Application Runtime Policy** field to the active state while the app is in development. The mode selected when activating ARP determines whether the automatically-generated policy records require manual review and approval.

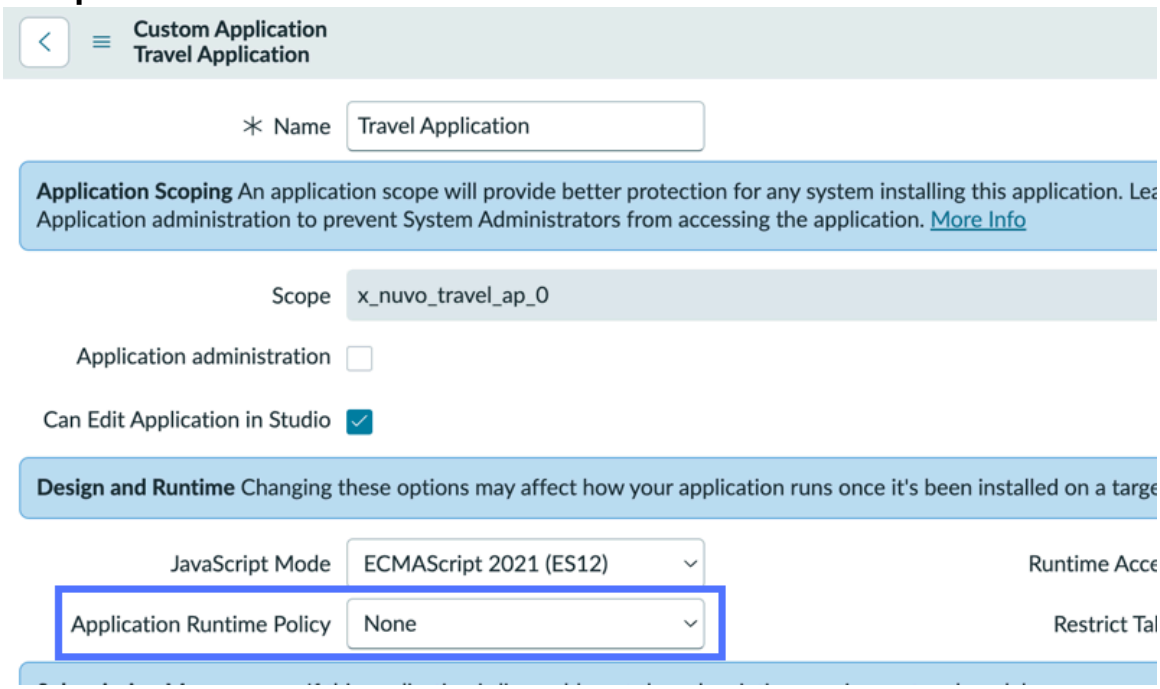
Procedure

1. Navigate to **All** > **System Applications** > **My Company Applications**.
2. From the **In Development** tab, select the application for which you want to enable ARP.
3. In the **Design and Runtime** section of the custom application form, select one of the following options in the **Application Runtime Policy** field.

Value	Description
None	The default. ARP is not enabled for this application and policies aren't automatically created.
Tracking	ARP is enabled. Policy records are automatically created and approved as you exercise application functionality. **Note:** Tracking mode should only be used during application development. When you're ready to publish your application, change the ARP mode to **Enforcing**.
Enforcing	ARP is enabled. Policy records are automatically created as you exercise application functionality, but out-of-scope access is blocked until you review and approve each policy record.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example



For more information about ARP modes, see Application Runtime Policy modes.

4. Select **Update** to save the custom application form.

Result

ARP is enabled for the application in the selected mode. When you exercise application functionality that accesses out-of-scope resources, ARP creates policy records in the applicable policy tables automatically.

Application Runtime Policy modes

When enabled, Application Runtime Policy operates in two possible modes: Tracking and Enforcing. Tracking mode automatically allows out-of-scope access attempts, while Enforcing mode automatically blocks out-of-scope access attempts until you review them.

Tracking mode

When Application Runtime Policy (ARP) is set to tracking mode for an application you're developing, network and cross-scope policy records are generated during development and set to the **Allowed**. No access is blocked, and you don't have to manually approve out-of-scope access attempts. If anything in the policy records does require modification, you can make manual changes as needed.

When a violation occurs in tracking mode, an informational message appears on the application page the violation originated from, indicating that the application attempted out-of-scope access. The access is allowed and any features that rely on out-of-scope access function as programmed.

Note: If you use Tracking mode while developing an application, you must change the mode to Enforcing before the application can be published to the ServiceNow Store or the Application Repository.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Enforcing mode

When an application is set to enforcing mode during development, ARP automatically creates network and cross-scope policy records but sets the state to **Requested** by default. Until the application author approves a policy, the corresponding access attempt is blocked.

When a violation occurs in enforcing mode, an error message appears on the application page indicating that the access attempt was blocked. Features that rely on the out-of-scope access might not be loaded, or might not function properly. You must navigate to the applicable policy table, find the policy record, and set its status to **Allowed** to enable access.

When an application with ARP in Enforcing mode is installed in another instance, either from the Application Repository or the ServiceNow Store, ARP functions in the following ways:

- ARP does not automatically create policy records in a requested state, even with the mode set to Enforcing.
- ARP blocks access to out-of-scope resources like tables, scriptables, script includes, and network APIs for which no policies exist.
- All blocked access attempts are logged in the ARP Violation Log table for administrator review.

Note: All policy records generated by ARP must be in the **Allowed** or **Denied** state before publishing to the Application Repository or the ServiceNow Store. Policies left in the **Requested** state block publication.

Using Application Runtime Policy

Application Runtime Policy (ARP) generates policies to control what external resources and cross-scope privileges an application can access at runtime. After opting in, manage policy records, configure resource quotas, and monitor access attempts during development.

Overview of using Application Runtime Policy

After you opt an application in to Application Runtime Policy (ARP), you can manage generated policy records, define resource limits, and monitor access attempts as you continue development.

Review network policies

Review automatically created network policy records and change their status as needed based on intended application behavior.

Before you begin

- For network policies to be automatically generated, Application Runtime Policy (ARP) must be enabled in either Tracking or Enforcing mode for the application you're developing.
- If ARP is in Enforcing mode, you must review and approve the generated network policies before the application can access the specified resource. For more information about Tracking and Enforcing modes, see Application Runtime Policy modes.

Role required: arp_admin or admin.

About this task

Note: All policy records generated by ARP must be in the **Allowed** or **Denied** state before publishing to the Application Repository or the ServiceNow Store. Policies left in the **Requested** state block publication.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **System Policy** > **Application Runtime Policy** > **Network Policies**.
2. From the list, locate a policy record with the **Requested** status and review the policy details. Policies with the **Allowed** or **Denied** status can also be reviewed and changed as needed.
3. **Optional:** Change details of the policy record, such as the **Short Description**, if needed.
   a. Double-click the value in the column you want to change (or use the keyboard shortcut).
   b. Enter the updated value.
   c. Select the Save (Enter) icon or use the Enter key to save your selection.

Example



> ⚠ **Warning:** Changing automatically generated policy details other than the short description could block necessary resource access. Test application functionality again after making any changes to policy record details.

4. Double-click the value in the **Status** field (or use the keyboard shortcut) to open the drop-down list of status options.
5. Select an option to allow or block resource access as defined in the policy.
   - Select **Allowed** to enable access.
   - Select **Denied** to block access.
6. Select the Save (Enter) icon or use the Enter key to save your selection.
7. Repeat for each **Requested** policy record.

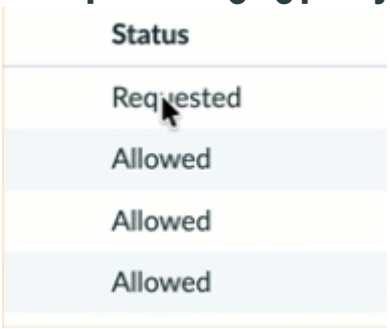
Result

Network policies with the **Allowed** state permit the application to connect to the specified network resource. When you reload the application page, any features that rely on the specified network resources function as programmed.

Network policies with the **Denied** state prevent the application from connecting to the specified network resource. To prevent errors, make sure the application doesn't continue attempting to access the blocked resources.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example: Changing policy status from the table



Create a segment policy

Create a segment policy to enable broad access to one or more policy categories without requiring individual policy records for each access attempt.

Before you begin

- You must be in the scope of the application you want to create a segment policy for. For more information about changing application scope, see Application picker.
- The application must be in development.
- Application Runtime Policy (ARP) must be activated in either Tracking or Enforcing mode for the selected application scope.

Role required: arp_admin or admin.

About this task

Segment policies provide a way to enable broad categories of allowed access to an application in development. Activating a segment policy for an area suppresses the creation of individual policy records via ARP for that area. If granular visibility into access patterns is needed for certification or review purposes, use individual policy records instead.

Because they provide such broad resource access, only use segment policies when out-of-scope resources called at runtime can't be known during application development.

Procedure

1. Navigate to **All** > **System Policies** > **Application Runtime Policy** > **Segment Policies**.
2. Verify that you're in the correct scope for the application you want to create a segment policy for.
   For more information, see Application picker.
3. Select **New**.
4. Select the **Active** option to enable segment policy at creation.
5. In the **Short Description** field, enter a description of the policy.
6. In the **Wildcard Policies** section, select the categories for which you want to enable broad access.

Option	Description
Network	Covers external network access.
Scripting	Covers cross-scope script access.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

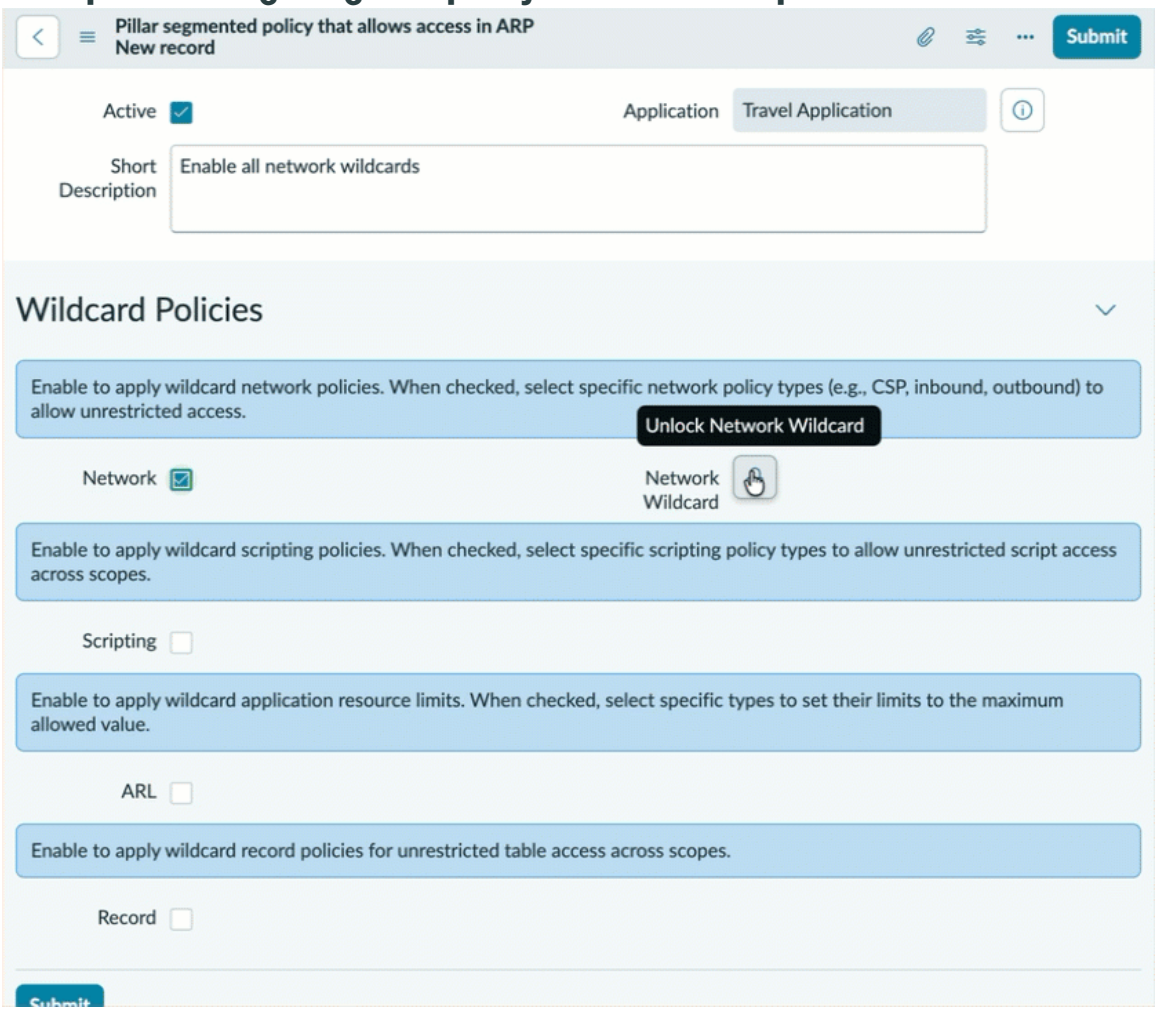
Option	Description
ARL	Covers application resource limits, including transaction event and scheduled job quotas.
Record	Covers cross-scope table access. There are no additional wildcard options for this category.

7. Within each selected category, select the Unlock Wildcard icon .
8. Select one or more wildcard options from the drop-down menu to specify sub-categories of unrestricted resource access.
   Select individual wildcards or select all wildcards to cover all subcategories. For more information about wildcard options for each category, see Wildcard Policies options.
9. Select **Submit**.

Result

The segment policy is created and activated. ARP no longer creates individual policy records for access attempts in the selected categories. Existing individual policy records for those categories can be deleted. Application functionality that relies on the covered categories continues to work without per-resource policy approval.

Example: Creating a segment policy with wildcard options



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Review cross-scope privilege policies

Review automatically created cross-scope privilege policy records and change their status as needed based on intended application behavior.

Before you begin

- For cross-scope privilege policies to be automatically generated, Application Runtime Policy (ARP) must be enabled in either Tracking or Enforcing mode for the application you're developing.
- If ARP is in Enforcing mode, you must review and approve the generated cross-scope privilege policies before the application can access the specified resource. For more information about Tracking and Enforcing modes, see Application Runtime Policy modes.

Role required: arp_admin or admin.

About this task

Note: All policy records generated by ARP must be in the **Allowed** or **Denied** state before publishing to the Application Repository or the ServiceNow Store. Policies left in the **Requested** state block publication.

Procedure

1. Navigate to **All** > **System Policy** > **Application Runtime Policy** > **Cross Scope Privilege**.
2. From the list, locate a policy record with the **Requested** status and review the policy details. Policies with the **Allowed** or **Denied** status can also be reviewed and changed as needed.
3. Double-click the value in the **Status** field (or use the keyboard shortcut) to open the drop-down list of status options.
4. Select an option to allow or block resource access as defined in the policy.
   - Select **Allowed** to enable access.
   - Select **Denied** to block access.
5. Select the Save (Enter) icon or use the Enter key to save your selection.
6. Repeat for each **Requested** policy record.

Result

Cross-scope privilege policies with the **Allowed** status permit the application to call the specified script or access the specified table in other application scopes. The cross-scope call completes successfully when the application functionality is next exercised.

Cross-scope privilege policies with the **Denied** state prevent the application from calling the specified script or accessing the specified table in another application scope. To prevent errors, make sure the application doesn't continue attempting to access the blocked resource.

Example: Changing policy status from the table



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Application Runtime Policy reference

Reference topics provide additional information about forms and options available through Application Runtime Policy.

Wildcard Policies options

When using a segment policy to enable broad access permissions by category, wildcard options for each category further specify which kinds of access are permitted.

Segment policies can be configured with the following wildcard options.

Network wildcards

Client Cross-origin Request

Enables client-side requests to an external host, such as a fetch or XHR call made from a UI page script.

Client Cross-origin Scripts

Enables client-side requests to load an external script resource.

Inbound Cross Scope Request

Enables access to API resources in different application scopes.

Inbound Global Request

Enables runtime access to ServiceNow global APIs.

Server Outbound Request

Enables outbound call to an external host, such as a call made from a script include.

Scripting wildcards

Script Includes

Enables unrestricted calls to script includes defined in other application scopes. For more information about script includes, see Script includes.

Scriptable

Enables unrestricted access to ServiceNow provided server-side JavaScript APIs, such as GlideRecord.

ARL (application resource limits) wildcards

Scheduled Job Limit

Sets the number of scheduled jobs the application can create to the maximum allowed value (100%).

Event Handler Limit

Sets the number of event handlers the application can trigger to the maximum allowed value.

API Transaction Limit

Sets the number of API transactions the application can make to the maximum allowed value.

Interactive Transaction Limit

Sets the number of transactions triggered by UI elements to the maximum allowed value.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Default Quota Configuration form

A description of the fields on the Default Quota Configuration form.

Default Application Quota Configuration rule fields

Field	Description
Configuration Name	A unique name to identify the default configuration.
Application	The scoped application's name. To change this field, use the application picker to change the application. This field can't be Global.
Order	A number that dictates the order in which application quotas are evaluated. If multiple default quota configurations apply to an application, the default quota configuration with the lowest number is applied first.
Enable Auto-Generated Quotas	When selected, application resource quotas based on this default configuration are generated in instances that install the application.
Application Condition	The condition that further specifies the scope that the default quota configuration applies to. The condition builder displays conditions that are only applicable to the application scope. For example, Version is 3.0.
Scheduled Job Quota %	Percentage of total processing time that can be allocated to jobs. When the default quota configuration is in enforcing mode and the quota threshold is exceeded, scheduled jobs are deferred until the next available time window. Default value: 20%
API Transaction Quota %	Total percentage of threads that can execute API requests for a given application and scope. When the default quota configuration is in enforcing mode and the quota threshold is exceeded, API requests are rejected. Default value: 30%
Interactive Transaction Quota %	Total percentage of threads that can execute UI requests. When the default quota configuration is in enforcing mode and the quota threshold is exceeded, UI requests are rejected. Default value: 30%
Event Handler Quota %	Percentage of total processing time that can be allocated to event handlers. When the default quota configuration is in enforcing

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Default Application Quota Configuration rule fields (continued)

Field	Description
	mode and the quota threshold is exceeded, event handler processing is deferred until the next available time window. Default value: 20%
Mode	Determines how default quota configurations are enforced. Options include the following: • **Disabled** prevents enforcements of a default quota configuration. • **Enforced** causes requested resources to be deferred or rejected when the quota threshold is exceeded. This is the default value. • **Log Only** adds log entries to indicate when the quota threshold is exceeded. No resources are rejected or deferred.

Testing and debugging applications

Verify the application meets your business requirements. Your testing should cover record operations (such as create, read, update, and delete), user interface elements (such as views and UI policies), runtime operations (such as business rules), and event script actions.

Testing and debugging on Core UI

Automated Test Framework (ATF)

Create and run automated tests to confirm that your instance works after making a change. Review failed test results to identify the changes that caused the failure and the changes that you should review.

Debugging scripts

Debug scripts using session logs and ServiceNow AI Platform debugging tools, such as a walk-through script debugger and error messages that display in the UI.

Script Debugger and Session Log

The Script Debugger enables users with the script_debugger role to debug server-side JavaScript, while the Session Log enables you to view and download required logs.

Script Tracer and debugging scripts

Use the Script Tracer to filter your debugging search and quickly narrow down script problems. Finding specific lines of scripts, rather than doing a wide search, helps save time and improves productivity.

Related ServiceNow applications and features

Test Management 2.0

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The ServiceNow® Test Management 2.0 application streamlines the management of testing processes to help you deliver software products more efficiently and with fewer errors. You can create multiple versions of a test and integrate with Agile Development 2.0.

Impersonate a user

Administrators can impersonate other authenticated users for testing purposes and view impersonation logs. When impersonating another user, the administrator has access to exactly what that user can access in the system, including the same menus and modules.

Automated Test Framework (ATF)

The Automated Test Framework (ATF) enables you to create and run automated tests to confirm that your instance works after making a change. For example, after an upgrade, during application development, or when deploying instance configurations with update sets. Review failed test results to identify the changes that caused the failure and the changes that you should review.

Note: By default, the system property to run automated tests is disabled to prevent you from accidentally running them on a production system. Run tests only on development, test, and other non-production instances to avoid data corruption and outage.

For developer training, see Using the Automated Test Framework on the ServiceNow Developer Site.

Get started

Explore  Learn about Automated Test Framework (ATF)	**Headless Browser for ATF**  Headless Browser for ATF	**Use**  Build and execute successful ATF tests
ATF tests and testing  Explore the ATF tests types and testing ways	**Administer**  Administering the ATF	**Test step categories**  Delve into the ATF test step categories

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Reference		
 Learn miscellaneous information about ATF		

Troubleshoot and get help

- Ask or answer questions about Automated Test Framework in the ServiceNow Community
- Search the Known Error Portal for known error articles

Exploring Automated Test Framework

The Automated Test Framework helps you ensure the integrity of your instance by enabling the creation and execution of automated tests following any changes.

Benefits

Automated Test Framework provides these benefits for change managers and developers.

- Reduce upgrade and development time by replacing manual testing with automated testing.
- Design tests once and reuse them in different contexts and with different test data sets.
- Keep test instances clean by rolling back test data and changes made after each test run.
- Create test suites to organize and run tests in batches.
- Schedule test suite runs.
- Enable non-technical test designers to create tests of standard ServiceNow AI Platform functionality.
- Reduce test design time by copying quick start tests and test suites.
- Create custom test steps to expand test coverage.

Automated Test Framework records and components

The Automated Test Framework consists of these records and components.

Test

A test is a logical grouping of related automated test steps that verify some functionality or feature. Each test is a record in the Test [sys_atf_test] table. Test designers typically create a test to verify one feature or a group of related features. For example, the **CSM: Create Product Case** test validates the creation of Product Case records. Each test has a related list of test steps and test results.

Test suite

A test suite is a collection of tests that run in a specific order. Test designers typically create a test suite to test an application or a group of related features. For example, the **CSM: Case Management** test suite validates the functionality of the Customer Service Management

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

application. Test designers can schedule running test suites and starting any required client test runners.

Quick start test

A quick start test is a test or test suite installed with the demo data of an application. Use quick start tests as templates to build your own tests and test suites. See this quick start test overview video for more information.

Test step

A test step combines a step configuration with the runtime test data necessary to run a step. The test step always specifies the order in which it runs in the test. Test steps have their own related list of step results. Each test step is a record in the Test Step [sys_atf_step] table that specifies a test action, the step configuration, and an execution order. Test designers add test steps to tests to verify functionality. For example, the first test step of the **CSM: Create Product Case** test is to impersonate the demo user John Jason who is authorized for Case Management.

Step configuration

A step configuration is a specific test action the Automated Test Framework can run. Step configurations do not contain any runtime test data and can only be run when test designers add them as part of a test step. Each step configuration is a record in the Test Step Config [sys_atf_step_config] table that specifies the input variables used to run the step configuration and the output variables available to other step configurations. For example, the **Impersonate** step configuration allows a test to impersonate another user.

The Automated Test Framework provides a default list of step configurations for most use cases and allows test designers to create their own custom step configurations.

Step variable

A step variable stores step-specific input and output values. For example, the **Open a New Form** step configuration has variables to specify the table and form view names. Use step variables to specify a particular test step target or to pass information to other test steps.

Test result

A test result stores the output of a test or test suite run. Each test result is a record in the Test Result [sys_atf_test_result] table that specifies the duration of the test run, the status of the test, and screenshots where available. Use test results to identify failing or non-running tests, and use the test logs to see more information about test results. By default, the system deletes test and test suite results 30 days after creation unless you enable the option to retain the test result indefinitely.

Step result

A step result stores the output of a test step run. Each step result is a record in the Step Result [sys_atf_test_result_step] table that specifies the status of the test step, a summary of the output, and a complete log of the output generated by the test step. Use step results to identify failures and functionality needing review.

Assert type

The **Assert type** field specifies the conditions that must be met for a test to pass. Test designers can use assertions to specify whether the results of an operation are expected or unexpected. For example, suppose you want to test that a record cannot be updated. In this case, you would

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

add a Record Insert test step and set the **Assert type** field to **Record was not inserted**. The test passes when the record insert fails.

Some test step categories that include steps with an **Assert type** field include:

- Server category: Assert which CRUD operations cause a test to pass or fail.
- Custom UI category: Assert which component states cause a test to pass or fail, and whether visible text causes a test to pass or fail.
- Forms in Service Portal category: Assert whether a form canceled in the browser due to validation errors, or a form successfully submitted to the server causes a test to pass.

Client test runners

A client test runner is a browser tab that runs client-side test steps within a ServiceNow AI Platform user interface. Client test runners require a browser tab to function. If no client test runner is available when you run a test, the system prompts you to open one. Testers can manually start a client test runner or select an existing client test runner. Test designers can schedule starting client test runners when they schedule running a test suite.

User roles

Assign roles to define Automated Test Framework permissions.

Role Title [name]	Role Description
atf_test_admin	Create or edit Automated Test Framework properties. Has permission to: • View the tests page • Create/edit/delete tests • Create/edit/delete test steps • View the step config page • View the test runner page • View the test suite results, test results and result items pages • Execute user tests • View, create, edit, delete and execute test suites • Create/edit step config records • Create/edit Automated Test Framework properties
atf_test_designer	View Automated Test Framework properties only (cannot create or edit properties). Has permission to: • View the tests page • Create/edit/delete tests • Create/edit/delete test steps • View the step config page • View the test runner page • View the test suite results, test results and result items pages

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Role Title [name]	Role Description
	• Execute user tests • View, create, edit, delete and execute test suites • View Automated Test Framework properties
atf_ws_designer	View or set basic authentication profiles needed for REST endpoints that require authentication. See REST category for more information.

ATF doesn't support these elements of Next Experience, but support for these features is planned for future releases:

- Pages built with UI Builder, including pages with lists and form components.
- Landing pages

Note: ATF still supports the Core UI, including Classic Environment (such as classic lists and forms).

Getting started with the Automated Test Framework

If you are new to the Automated Test Framework, read this overview to learn what the framework can do. Next, follow the tutorial to create and run a test that uses the most basic of ATF features. After you feel comfortable with the basics, explore more advanced features provided by the ATF.

ATF features provide flexibility in how you test your instance.

Test step configuration categories

Category	Description
Service Catalog in Service Portal	Perform end-to-end testing for a catalog item in the Service Portal. • Open a record producer, catalog item, or order guide. • Set variable values and catalog item quantity. • Validate variable values, states, price, and items included in an order guide. • Navigate in an order guide. • Open and toggle catalog items in an order guide. • Add an item or an order guide to a shopping cart. • Order a catalog item or an order guide. • Submit a record producer.
Application Navigator	Create tests to check navigation features.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Category	Description
	• Verify that application menus are listed in the left navigation bar. • Verify that application modules are listed in the left navigation bar. • Navigate to a module as if a user clicked the module in the left navigation bar.
Custom UI	Create simple tests that mimic user actions with no scripting. • Set component values. • Assert that specified text is or is not on a page. • Validate component values. • Click components. • Validate the states of components (read-only or not read-only).
Form	Create tests of forms. • Open a new form or an existing record. • Set field values. • Validate field values or field states (such as mandatory, not mandatory, read only, not read only, visible, and not visible). • Validate whether a UI action is visible. • Click a button on a modal page. • Click a UI action. • Submit a form.
Service Catalog	Perform end-to-end testing for a catalog item. • Open a catalog item or a record producer. • Search for a catalog item. • Set variable values and catalog item quantity. • Validate variable values, states, and price. • Add an item to a shopping cart. • Order a catalog item. • Submit a record producer.
Forms in Service Portal	Create tests of forms in the Service Portal. • Open a form. • Set field values.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Category	Description
	• Validate field values or field states (such as mandatory, not mandatory, read only, not read only, visible, and not visible). • Validate whether a UI action is visible. • Click a UI action. • Submit a form.
REST	Create and send an Inbound REST request and verify the response. • Test any REST endpoint on the instance. • Use a REST request to create records, as well as retrieve, update, or delete records created in a previous test step or that already existed on the instance. • Verify the response status code, response headers, response time, and response payload.
Server	Perform more complex operations, including the following: • Perform unit tests using JavaScript, including tests using the Jasmine test framework. • Test business rules, script includes, and other scripts. • Create tests that operate on data that you define.

Output variables

Many test steps return output variables whose values you can use as inputs to a later step. For example, you can use output variables to accomplish the following tasks:

- Perform a server-side assert on a record that you previously inserted.
- Create a record as one user, and then reopen its form as a different user.

Custom test step configurations

In addition to the steps built into the Automated Test Framework, you can create custom test step configurations. These custom steps can take input variables and return output variables that you define.

Note: You can only define custom test steps that run on the server. The Automated Test Framework does not support creating custom step configurations that run on the browser.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Data preservation

The Automated Test Framework automatically tracks and deletes any data created by running tests, and automatically rolls back changes after testing.

Test suites

Test suites enable you to execute a batch of tests in a specified order. In addition, test suites can be hierarchical, with suites nested within other suites. You can associate test suites with schedules that determine when the system runs the test suites.

Build and run your first automated test

Follow these step-by-step instructions to create and run your first automated test. This test creates a new user record.

- The Automated Test Framework (com.glide.automated_testing_framework) plugin must be activated. It is activated by default on zBoot or upgrade.
- If necessary, enable test execution. For instructions, see Enable or disable executing Automated Test Framework tests.

  Note: By default, the system property that is used to run automated tests is disabled to prevent you from accidentally running these tests on a production system. To avoid data corruption or an outage, run tests only on development, test, and other non-production instances.

- Role required: admin

Create new test

Create a new automated test record.

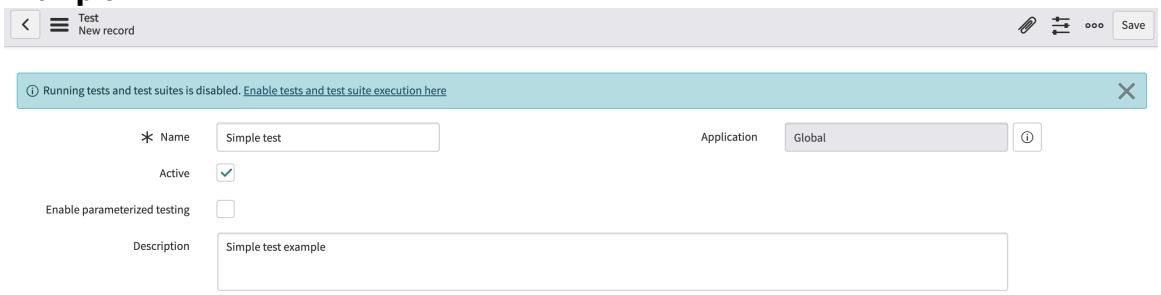
Before you begin

Role required: atf_test_admin or admin

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Tests**.
2. Click **New**.
3. On the Test new record form, enter a name for your test in the **Name** field.
   The system identifies this test by this name whenever it displays a list of tests (for example, under the Tests module).

   Example

4. In the **Description** field, enter a description for your test.
5. Click **Save**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Result

The system creates a new test record and returns to the list of tests. For more information about creating new automated tests, see Create a new automated test.

Add the first step to the new test

Add the first of three steps to the automated test.

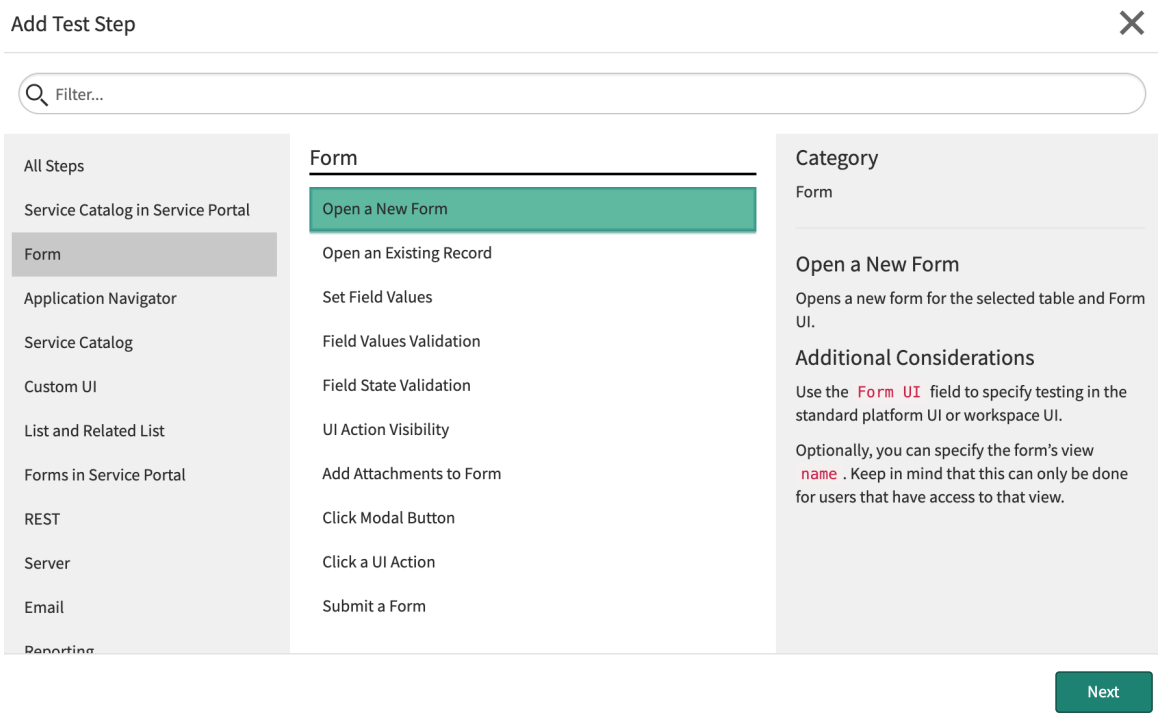
Before you begin

Role required: atf_test_admin or admin

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Tests**.
2. Click the test that should contain the new test steps.
3. In the Test Steps related list at the bottom of the Test form, click **Add Test Step**.
4. In the middle column, click the row for the step type **Open a new form**, then select **Next**.

Example

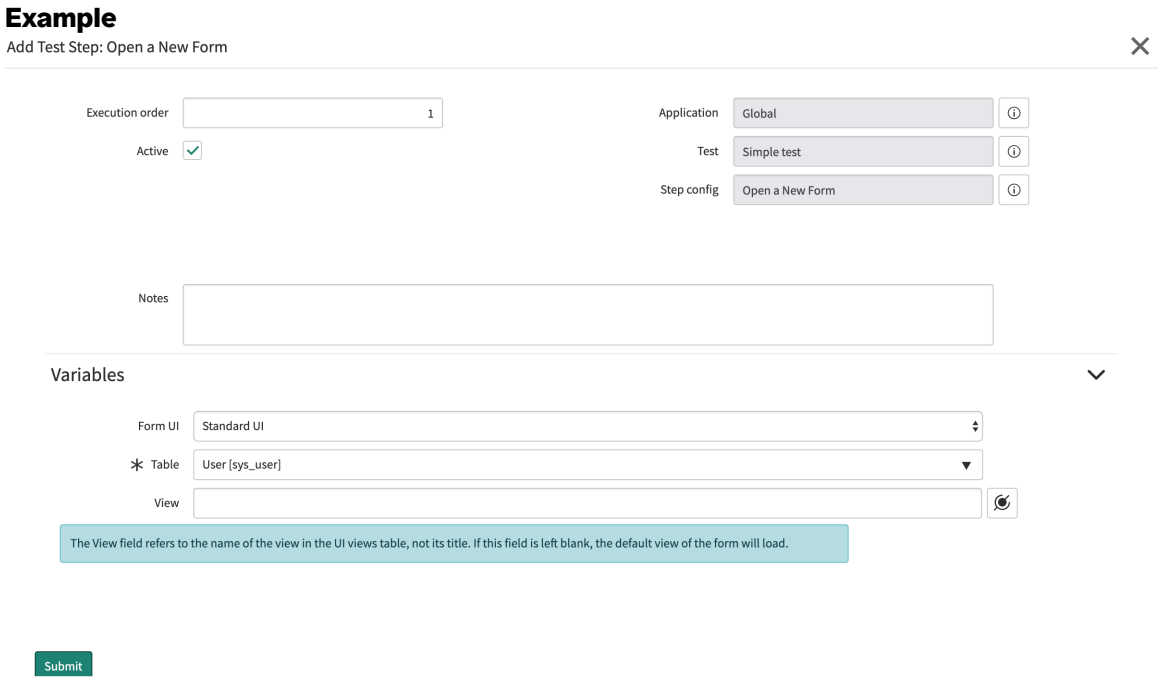


The system displays the **Add Test Step** form for the Open a new form step.

5. From the **Table** field, select the **User [sys_user]** table and click **Submit**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example



The system creates the new step and returns to the test record.

6. Click **Update**.

Result

The system creates a new test record and returns to the list of tests. For more information about adding steps to automated tests, see Add steps to an automated test.

What to do next

For some ideas on how to continue learning about the Automated Test Framework, see Next steps with the Automated Test Framework.

Add the second step to your automated test

Add the second of three steps to the automated test.

Before you begin

Role required: atf_test_admin or admin

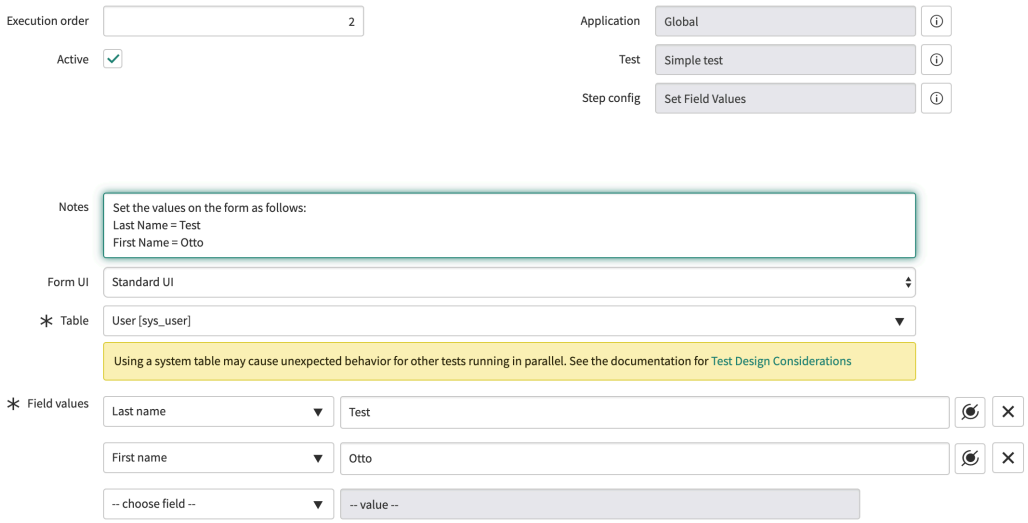
Procedure

1. Click the test that should contain the new test steps.
2. In the Test Steps related list at the bottom of the Test form, click **Add Test Step**.
3. In the middle column, click **Set Field Values**, then click **Next**.
   The system displays the **Add Test Step** form for the Set values step.
4. In the field values section, set **Last name** to `Test` and **First name** to `Otto` (or other names of your choice).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example

Add Test Step: Set Field Values



5. Click **Submit**.

Result

The system creates the new step and returns to the test record. For more information about adding steps to automated tests, see Add steps to an automated test.

Add the third step to your test

Add the last of three steps to the automated test.

Before you begin

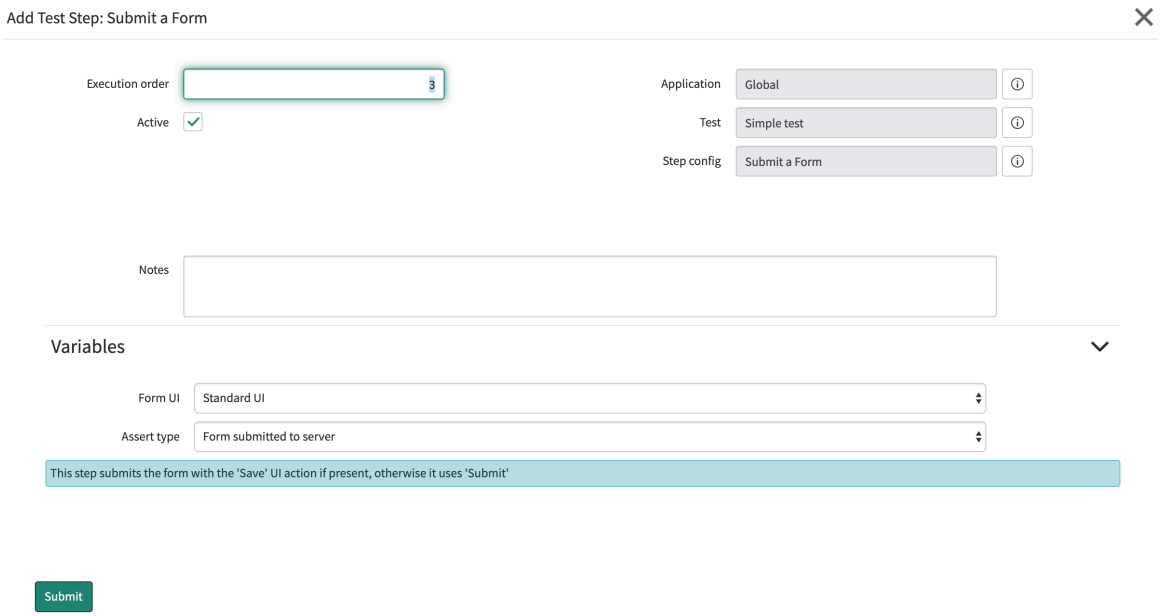
Role required: atf_test_admin or admin

Procedure

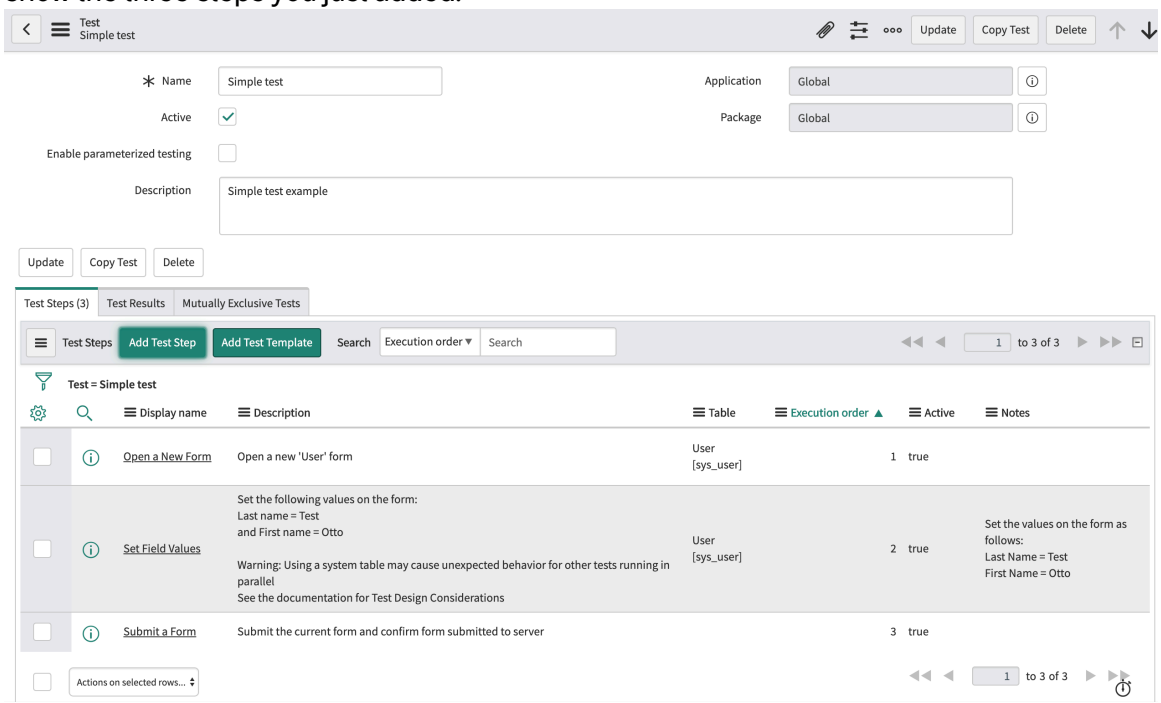
1. Click the test that should contain the new test steps.
2. In the Test Steps related list at the bottom of the Test form, click **Add Test Step**.
3. In the middle column, click **Submit a Form**, then click **Next**.
   The system displays the **Add Test Step** form for the Submit a form step.
4. Leave all values as set by default and click **Submit**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example



The system creates the new step and returns to the test record. The test record should now show the three steps you just added.



5. Click **Update**.

Result

The system returns to the list of tests. For more information about adding steps to automated tests, see Add steps to an automated test.

Run your test

After adding test steps, run and monitor the progress of the automated test.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin
Role required: atf_test_admin or admin

Procedure

1. If necessary to view the Tests list, click **Tests**.
2. Click the row containing the test you just created.
   The system displays the **Test** form.
3. Click **Run Test**.
   Because this test includes a form step (any step involving a UI), the system displays a dialog box asking you to choose among any currently-running test clients or start a new test runner. To continue, select **start a new test runner** and click **Run Test**.

Result
The system displays the Run Test progress dialog. For more information about running automated tests, see Run an automated test.

Monitor test progress and view test results

Monitor the progress of the automated test and view its test results.

Before you begin
Role required: atf_test_admin or admin

About this task
If needed, you can cancel a test even while it's running. For more information, see Cancelling automated tests and test suites.

Procedure

1. Monitor the progress of the test in the Run Test progress dialog.

   Note: If your test creates data, the system rolls back that data after all steps in the test complete.

2. When complete, click **Go to Results** on the Run Test progress dialog to display the **Test Results** list, where you can view and analyze the results.

Next steps with the Automated Test Framework

After you feel comfortable creating and running simple tests, explore the more advanced features of the Automated Test Framework.

Batch tests together with test suites
If you commonly run a set of tests together, you can group them using test suites. To learn about test suites, see Building and running automated test suites.

Pass data from one test step to another with input and output variables

You can pass data from one test step to another using input variables and output variables. For more information, see Passing data from one automated test step to another.

Reuse common sequences of steps with templates
If you frequently add the same sequence of steps to your tests, save time by creating a template. To learn more about templates, see Add a predefined list of steps (template) to an automated test

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Domain separation and Automated Test Framework

Domain separation is supported in the Automated Test Framework. Domain separation enables you to separate data, processes, and administrative tasks into logical groupings called domains. You can control several aspects of this separation, including which users can see and access data.

Support level: Standard*

The support level is Standard but has some exceptions or special conditions.

- Includes all aspects of **Basic** level support.
- Business logic: The service provider (SP) creates or modifies processes per customer. The use cases reflect proper use of the application by multiple SP customers in a single instance.
- The instance owner must be able to configure minimum viable product (MVP) business logic and data parameters. This configuration is done per tenant, as expected for the specific application.

Sample use case: An admin must be able to make comments required when a record closes for one tenant, but not for another.

For more information on support levels, see Application support for domain separation.

ATF use case

Automated test framework design and runtime access are solely for the owner of the instance to validate domain-specific processes. By designing tests for my customers we confirm results per tenant.

Testing

When testing domain separation during ATF test steps, you must set the domain first. This should be part of the first impersonation step of each of the ATF test steps when they are dependent on a domain being set. To learn more about domain separation recommended practices, see Domain separation recommended practices for service providers.

Related topics

Domain separation for service providers

Headless Browser for Automated Test Framework

Improve your UI testing by automating the creation of browsers to process Automated Test Framework (ATF) User Interface (UI) tests. This feature is known as a headless browser, and helps you test UI functionality without having to manually open a browser to the Client Test Runner, which is what processes the UI tests in your local browser.

Important: Headless Browser is a legacy feature of ATF; it is recommended to instead use the ATF Test Generator and Cloud Runner application. Cloud Runner offers you an easy setup and a seamless user experience. You can install Cloud Runner from the store app.

If you are an on-premise instance user, you must continue using the Headless Browser option. Cloud Runner is not currently available for on-premise instances.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Background

ServiceNow customers use Automated Test Framework (ATF) to test applications and instances. When developing in the ServiceNow platform, ATF ensures that your changes have both the desired behavior and don't break existing features.

Since the ServiceNow Orlando release, customers can automate the testing and deployment of their applications via the Continuous Integration/Continuous Delivery (CI/CD) API.

Overview of Headless Browser

The automation provided by the ServiceNow Headless Browser for ATF skips the step of manually opening a browser during testing. There are several sequential procedures to follow in the one-time setup. Below are the instructions for both Linux and Windows setup.

See Headless Browser for ATF for more information on the Support Model.

Headless Browser setup for Linux

The ServiceNow® Headless Browser for Automated Test Framework provides automation so you can skip having to manually open a browser during testing. The Headless Browser setup is available in both Linux and Microsoft Windows. This topic covers the setup for Linux.

Important: Headless Browser is a legacy feature of ATF; it is recommended to instead use the ATF Test Generator and Cloud Runner application. Cloud Runner offers you an easy setup and a seamless user experience. You can install Cloud Runner from the store app.

If you are an on-premise instance user, you must continue using the Headless Browser option. Cloud Runner is not currently available for on-premise instances.

There are several sequential procedures to follow in the one-time setup.

Prerequisites

Note: MFA must be disabled in your instance to use the Headless Browser option.

Role required: admin on your ServiceNow instance and local administrator on the host machine.

- Install the Docker application
- Install Java Runtime Environment (JRE) 1.8 - required for keytool utility

  Note: The version must be Java 1.8, or errors result in your ServiceNow instance when trying to validate the certificate created with the keytool utility. To learn more, see Create encryption keys using the Java KeyStore keytool.

- Install OpenSSL
- Two-way communication
  - There must be two-way communication between the instance URL and your server.
  - Find the IP address of your server and get your hostname. You can use one or both of them, but you need at least one. Make sure the address or hostname are visible from your ServiceNow instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- See My IP Address to find the IP address ranges of your instance. You can then configure your server to allow inbound access to all of the indicated IP addresses/ranges.
- Use Port 2376 or your own default port for this procedure. Make sure your server's firewall rules allow inbound requests on this port from the IP addresses you get from the instance.

Note: If you don't have a hostname and are connecting via the IP address, you can enter the IP address and put "localhost" in the Hostname environment variable.

Steps to set up the Headless Browser for Linux

Follow these six steps (plus verification) to set up the Headless Browser for Linux.

Generate certificates for Headless Browser setup for Linux

Generate TLS/SSL certificates to secure the Docker REST API and authenticate HTTP requests.

Before you begin

Complete the prerequisites listed in the Headless Browser setup for Linux topic.

Role required: admin on your ServiceNow instance and local administrator on the host machine.

About this task

Warning: Get certificate authority keys from a trusted certificate authority.

By default when exposing the Docker API, requests are not authenticated, which can leave your host machine vulnerable to attack. Docker API, however, does support TLS authentication where requests are verified against public private keys provided in the HTTPS encryption. In this step you create those keys for the server and the client.

Tip: To make remembering these easier, enter the following commands in your Linux terminal. **Note**: Do not add these environment variables to your terminal profile. For security reasons, they should only exist for the duration of the current session.

- `export PASSWORD="<password to generate the certificates with>"`
- `export SERVERIP="<this server's IP address>"`
- `export HOSTNAME="<hostname of this server>"`

To learn more, see Use TLS (HTTPS) to protect the Docker daemon socket.

Procedure

1. Open a command line.
2. Generate a self-signed certificate authority key or retrieve a keypair from a trusted certificate authority.
   The following commands are an example. Note that your configuration might vary.
   - `openssl genrsa -aes256 -passout pass:$PASSWORD -out ca-key.pem 4096`
   - `openssl req -passin pass:$PASSWORD -new -x509 -days 365 -key ca-key.pem -sha256 -out ca.pem`
   - `chmod 0400 ca-key.pem`
   - `chmod 0444 ca.pem`
3. Generate the server keypair using the certificate authority key.
   The following commands are an example. Note that your configuration might vary.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- `openssl genrsa -out server-key.pem 4096`
- `openssl req -subj "/CN=$HOSTNAME" -new -key server-key.pem -out server.csr`
- `echo "subjectAltName = DNS:$HOSTNAME,IP:$SERVERIP,IP:127.0.0.1" > extfile.cnf`
- `openssl x509 -passin pass:$PASSWORD -req -days 365 -in server.csr -CA ca.pem -CAkey ca-key.pem -CAcreateserial -out server-cert.pem -extfile extfile.cnf`

4. Create the client keypair using the certificate authority key.
   The following commands are an example. Note that your configuration might vary.
   - `openssl genrsa -out client-key.pem 4096`
   - `openssl req -subj "/CN=$HOSTNAME" -new -key client-key.pem -out client.csr`
   - `echo "extendedKeyUsage = clientAuth" > extfile.cnf`
   - `openssl x509 -passin pass:$PASSWORD -req -days 365 -in client.csr -CA ca.pem -CAkey ca-key.pem -CAcreateserial -out client-cert.pem -extfile extfile.cnf`

   Now you've created all your encryption keys.

5. Import the CA Public Key and Client Keypair to a java keystore.
   The following commands are an example. Note that your configuration might vary.

   Create the keystore file and create a password for it (and save for later use): `keytool -genkey -keyalg RSA -alias dse -keystore my.keystore`

   Delete a default entry from the keystore file: `keytool -delete -alias dse -keystore my.keystore`

   Import the CA public key to the keystore: `keytool -import -keystore my.keystore -trustcacerts -alias ca -file ca.pem`

   Import the client keypair.

   Note: You are creating a new pkcs12 keystore file and importing the keypair to it. Then copy the contents to your original keystore file.

   - `openssl pkcs12 -export -name clientkeypair -in client-cert.pem -inkey client-key.pem -out clientkeypair.p12`
   - `keytool -importkeystore -destkeystore my.keystore -srckeystore clientkeypair.p12 -srcstoretype pkcs12 -alias clientkeypair`

   Now that you have added all of the certificates to the keystore file, save the file **my.keystore** for later, as it will be uploaded to the ServiceNow instance. In addition, be sure to remember the password that you entered when prompted to create the keystore file; you will need to enter that into a form in the ServiceNow instance.

Configure Docker for Headless Browser setup in Linux

Complete Step 2 in the Linux setup for the ServiceNow® Headless Browser for ATF: Configure Docker Server to authenticate all requests.

Before you begin

Complete Step 1: Generate certificates for Headless Browser setup for Linux

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

This task: After creating your client and server keys as directed in Step 1, now you configure the Docker Server to authenticate all requests using those keys, and expose the Docker API on Port 2376.

Role required: admin on your ServiceNow instance and local administrator on the host machine.

Procedure

1. Configure Docker to use the certificates that you generated in Step 1.
   a. Find or create the /etc/docker/daemon.json file. You can run: `touch /etc/docker/daemon.json`
   b. Edit the daemon.json file and add the following JSON values. Be sure to replace with the correct paths to your certificates:

```
{
"debug": true,
"tlscacert": "<path to your certificates>/ca.pem",
"tlscert": "<path to your certificates>/server-cert.pem",
"tlskey": "<path to your certificates>/server-key.pem",
"tlsverify": true
}
```

   To learn more, see Configure the Docker daemon.
2. Configure Docker to expose the remote API on a port (Port 2376 is suggested).
   You can configure Docker to accept remote connections with the docker.service systemd unit file for Linux distributions using **systemd**, such as recent versions of RedHat, CentOS, Ubuntu and SLES, or with the daemon.json file, which is suggested for Linux distributions that do not use systemd.

   If using **systemd (systemctl)**:

   a. Use the command `sudo systemctl edit docker.service` to open an override file for docker.service in a text editor.
   b. Add or modify the following lines, substituting your own port if not using 2376.

```
[Service]
ExecStart=
ExecStart=/usr/bin/dockerd -H tcp://0.0.0.0:2376
```

   c. Save the file.
   d. Reload the systemctl configuration.

```
sudo systemctl daemon-reload
```

   e. Restart Docker

```
sudo systemctl restart docker.service
```

3. To enable Docker access for the current user via a command line, copy certificates to the user's Docker home directory:
   - `mkdir -pv ~/.docker`
   - `cp ca.pem ~/.docker`
   - `cp client-key.pem ~/.docker/key.pem`
   - `cp client-cert.pem ~/.docker/cert.pem`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

4. Set DOCKER_HOST and DOCKER_TLS_VERIFY environment variables for your user:
   - `echo "export DOCKER_HOST=tcp://${SERVERIP}:2376 DOCKER_TLS_VERIFY=1" >> ~/.bash_profile`
   - `source ~/.bash_profile`

   To learn more, see Manage Docker as a non-root user.

Create the Docker image and containers for Headless Browser setup in Linux

Pull the Docker image from the Public Registry.

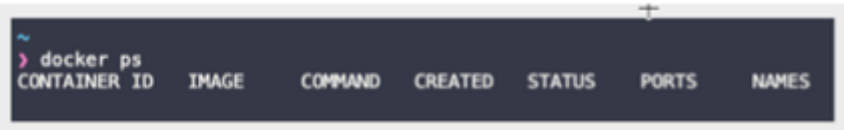
Before you begin

Complete Step 2: Configure Docker for Headless Browser setup in Linux

Role required: admin on your ServiceNow instance and local administrator on the host machine.

Procedure

1. In the command line, run `docker ps` to verify that Docker is working.
   Your results should look like this:



2. Pull the latest Docker image from the GitHub repo ATF Headless Runner.
   This command is an example, as these images are subject to change.
   `docker pull ghcr.io/servicenow/atf-headless-runner:lin-1.0.0`

   Note: The instance-to-image compatibility makes sure that the automation script inside the Docker image is compatible with the instance code. Elements such as the user interface might change over time to support new features or upgrades.

Add secrets to Docker for Headless Browser setup in Linux

Create a Docker secret, which stores the password of the ServiceNow user who will log into the instance to execute the tests. Docker Secrets is a feature of ServiceNow® for securely storing the passwords that will be used in containers.

Before you begin

Complete Step 3: Create the Docker image and containers for Headless Browser setup in Linux

Role required: none.

About this task

Inside of the Docker container is an automation script that opens a web browser, logs into the instance, and opens the client test runner page. In order to log into the ServiceNow instance, you will need a user password. In this procedure you use a Docker feature called **Docker Secrets** so you can securely store passwords. When you run containers, the password is automatically available to log in to your instance.

Role required: admin on your ServiceNow instance and local administrator on the host machine.

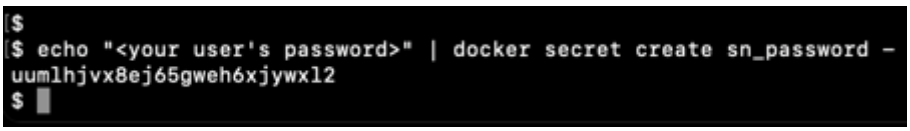
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. At the command line, enter `docker swarm init`
2. Enter `echo "<your user's password>" | docker secret create sn_password -`

   Note: Replace `<your user's password>` with the user's ServiceNow password.

   Your results should look like this:



Result

The result is your secret ID, which you must save for later use. The secret ID will be added to the ServiceNow instance in the **sys_property** `sn_atf.headless.secret_id`.

Set up instance for Headless Browser in Linux

Step 5 in the Linux setup for the ServiceNow® Headless Browser for ATF: Set up your instance so it can support the Headless Browser.

Before you begin

Complete Step 5: Add secrets to Docker for Headless Browser setup in Linux

Role required: admin on your ServiceNow instance and local administrator on the host machine.

This task: Set up the instance so that it successfully communicates and authenticates with the host machine.

Procedure

1. Create user:
   a. Navigate to **User Administration** > **Users** and select **New**.
   b. Create a user - User ID, which can be whatever you want, as well as a user name and password. The password should be the same as the one you created for your Docker Secret container.
   c. Add the role **atf_test_designer** for this user. (To learn more, see the User Roles section in the Exploring Automated Test Framework topic.)
2. Create certificate:
   a. Navigate to **System Definition** > **Certificates** to open the **sys_ certificate** table. Create a new certificate (the name can be whatever you prefer):
      - **Type**: Java Key Store
      - **Password**: Password for the keystore that you created in Step 1 (Generate certificates for Headless Browser setup for Linux).
   b. Select the Attach (paper clip) icon to attach the keystore file you saved earlier to this record.
   c. Select **Submit**.
   d. Click **Validate certificate** and confirm that the success message displays.
3. Create protocol profile:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. Navigate to **System Security** > **Protocol Profiles** to open the **sys_protocol_profile** table. Create a new protocol profile record:

Option	Description
Protocol	One word, lowercase: “docker”
Default port	2376 (or the one you chose to use)
Keystore	Docker host keystore

b. Select **Submit**.

4. Create a Docker spoke connection:
   a. Navigate to **Connections & Credentials** > **Connection & Credential Aliases** to open the **sys_alias** table.
   b. Select the alias with the name **Docker**.
   c. Under the Connections related list, select **New**.
   d. Fill in these fields:
      - **Name**: Any text you prefer
      - **Credential field**: Leave blank
   e. Select the **URL Builder** check box.
   f. Select the **Mutual authentication** check box.
   g. In the **Protocol profile** field, select the protocol profile you created earlier.
   h. In the **Host** field, add the IP address or Host name of your server.
   i. Select **Submit**.
   The Connection URL is automatically created by the system.
5. Modify properties:

> ⚠ **Warning:** By default, the *com.glide.communications.trustmanager_trust_all* property is set to **false**. The ServiceNow AI Platform trusts only certificates that it can verify against the JVM certificate store. Self-signed and enterprise-signed certificates are not trusted.

You need to do this only when using self-signed certificates. To learn more, see Generate certificates for Headless Browser setup for Linux.

- `com.glide.communications.httpclient.verify_hostname`: false
- `com.glide.communications.trustmanager_trust_all`: true

Configure ATF for Headless Browser in Linux

Step 6 in the Linux setup for the ServiceNow® Headless Browser for ATF: Configure ATF with properties.

Before you begin

Complete Step 5: Set up instance for Headless Browser in Linux

Role required: admin on your ServiceNow instance and local administrator on the host machine.

This task: Now that you've set up your connection and authentication, configure ATF with several properties so that it can start containers successfully on the host machine.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. In your instance, navigate to **ATF** > **Administration** > **Properties**.
2. Enable the top two properties: **Enable test/test suite execution** and **Enable scheduled test suite execution**.
3. Scroll down to the **Headless Runner Properties** section.
4. Enable the top check box.
5. Enter the following values in the Headless Runner Properties form:

Form Label	Property Name	Value to Imput
The user account used to login from the headless client test runner and begin the tests	sn_atf.headless.username	Username of the integration user that you created in Add secrets to Docker for Headless Browser setup in Linux
The Docker secret ID that has the password stored of the user account	sn_atf.headless.secret_id	Docker "Secret ID" that you obtained earlier, as well as the Docker secret name that you created in Step 2 of the Add secrets to Docker for Headless Browser setup in Linux procedure.
The Docker image that is used for headless Client Test Runner	sn_atf.headless.docker_image_name	Image name with tag that you downloaded: `ghcr.io/servicenow/atf-headless-runner:<tagname>`

Note: If you forgot the secret ID, go to the Docker host and run the command `docker secret list`

6. Leave the rest of the fields as they are.
7. Select **Save**.

Verify Headless Browser procedures in Linux

Step 7, the final step in the Linux setup for the ServiceNow® Headless Browser for ATF: Verify that your Headless Browser setup procedures are successful.

Before you begin

Role required: admin

Complete Step 6: Configure ATF for Headless Browser in Linux

Role required: admin on your ServiceNow instance and local administrator on the host machine.

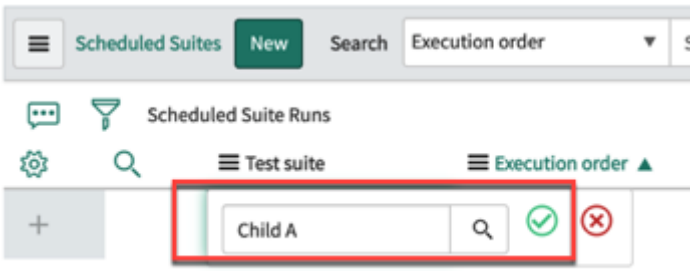
This task: Verify that Steps 1-6 in your Headless Browser setup procedures are successful.

Procedure

1. Go to **ATF Schedules** and next to the Suite Schedules link, select **New**.
2. In the Run box, select **On Demand**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

3. Select **Save**.
4. In the Scheduled Suite related list, add a new record for your test suite by double-clicking the **+** sign and choose your browser name of **Chrome** or **Firefox**.
   Use **Child A** as a good example to run.



5. Set the OS name to Linux.
6. Select **Execute Now**.
7. Verify that your test passed.
   When your verification is successful, any suites with UI tests run by scheduled suites or via CICD now automatically create headless test runners without the need to manually open the "scheduled client test runner" page.

Headless Browser setup for Microsoft Windows

The ServiceNow® Headless Browser for Automated Test Framework (ATF) provides automation so you can skip having to manually open a browser during testing. The Headless Browser setup is available in both Linux and Microsoft Windows. This topic covers the setup for Windows.

Important: Headless Browser is a legacy feature of ATF; it is recommended to instead use the ATF Test Generator and Cloud Runner application. Cloud Runner offers you an easy setup and a seamless user experience. You can install Cloud Runner from the store app.

If you are an on-premise instance user, you must continue using the Headless Browser option. Cloud Runner is not currently available for on-premise instances.

There are several sequential procedures to follow in the one-time setup. Below are the instructions for the Microsoft Windows setup.

Prerequisites

Note: MFA must be disabled in your instance to use the Headless Browser option.

Role required: admin on your ServiceNow instance and local administrator on the host machine.

Warning: The only supported version of Microsoft Windows as a host is Windows Server 2019 v10.0.17763.737. No other versions are supported. If you are unable to meet these requirements, a Linux host is recommended.

- Make sure that the following programs are installed on your Windows server:
  - Docker: Install Docker for Headless Browser setup for Microsoft Windows
  - Java keytool: Chocolatey tool for javaruntime
  - OpenSSL: Chocolatey tool for openssl
- Two-way communication

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- There must be two-way communication between the instance URL and your server.
- Find the IP address of your server and get your hostname. You can use one or both of them, but you need at least one. Make sure the address or hostname are visible from your ServiceNow instance.
- See My IP Address to find the IP address ranges of your instance. You can then configure your server to allow inbound access to all of the indicated IP addresses/ranges.
- Use Port 2376 or your own default port for this procedure. Make sure your server's firewall rules allow inbound requests on this port from the IP addresses you get from the instance.

Note: If you don't have a hostname and are connecting via the IP address, you can enter the IP address and put "localhost" in the Hostname environment variable.

- To learn more, see Use TLS (HTTPS) to protect the Docker daemon socket.

Steps to set up the Headless Browser for Windows

Follow these seven steps (plus verification) to set up the Headless Browser for Windows.

Install Docker for Headless Browser setup for Microsoft Windows

Step 1 in the Windows setup for the ServiceNow® Headless Browser for Automated Test Framework: Install Docker.

Before you begin
Role required: admin

- See Prerequisites in the Headless Browser setup for Microsoft Windows topic.
- **Tip:** To make remembering these easier, set the following environment variables:
  - set PASSWORD=<password>
  - set SERVERIP=<server ip>
  - set HOSTNAME=<hostname>

Warning: The only supported version of Microsoft Windows as a host is Windows Server 2019 v10.0.17763.737. No other versions are supported. If you are unable to meet these requirements, a Linux host is recommended.

Procedure

1. Open an elevated PowerShell session and install the Docker-Microsoft PackageManagement Provider from the PowerShell Gallery.
   PowerShell

   ```
   Install-Module -Name DockerMsftProvider -Repository PSGallery -Force
   ```

   If you're prompted to install the NuGet provider, type `Y` to install it as well.

2. Use the PackageManagement PowerShell module to install the latest version of Docker.
   PowerShell

   ```
   Install-Package -Name docker -ProviderName DockerMsftProvider
   ```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

When PowerShell asks you whether to trust the package source 'DockerDefault', type A to continue the installation.

3. After the installation completes, restart the computer.
   PowerShell

   ```
   Restart-Computer -Force
   ```

Generate certificates for Headless Browser setup for Microsoft Windows

Generate TLS/SSL certificates to secure the Docker REST API and authenticate HTTP requests.

Before you begin

Role required: admin on your ServiceNow instance and local administrator on the host machine.

Warning: The only version of Microsoft Windows that ServiceNow supports as a host is Windows Server 2019 v10.0.17763.737. No other versions are supported. If you are unable to meet these requirements, a Linux host is recommended.

- Complete the prerequisites listed in the Headless Browser setup for Microsoft Windows topic. Make sure that the following programs are installed on your Windows server:
  - Docker: Docker application for Windows
  - Java keytool: Chocolatey tool for javaruntime
  - OpenSSL: Chocolatey tool for openssl

Note the following requirements:

- Two-way communication

  Warning: Be sure to get certificate authority keys from a trusted certificate authority.

  - There must be two-way communication between the instance URL and your server.
  - Find the IP address of your server and get your hostname. You can use one or both of them, but you need at least one.

    Note: If you don't have a hostname and are connecting via the IP address, you can enter the IP address and put "localhost" in the Hostname environment variable.

  Tip: To make remembering these easier, set the following environment variables:
  - `export PASSWORD="<password to generate the certificates with>"`
  - `export SERVERIP="<this server's IP address>"`
  - `export HOSTNAME="<hostname of this server>"`

- Port: Use Port 2376 or your own default port for this procedure.

  Note: Make sure your firewall rules allow inbound requests on this port

- To learn more, see Use TLS (HTTPS) to protect the Docker daemon socket.

About this task

By default when exposing the Docker API, requests are not authenticated, which can leave your host machine vulnerable to attack. Docker API, however, does support TLS authentication where

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

requests are verified against public private keys provided in the HTTPS encryption. In this step you create those keys for the server and the client.

Procedure

1. Open a command line.
2. Generate a self-signed certificate authority key or retrieve a keypair from a trusted certificate authority.
   The following commands are an example. Note that your configuration might vary.

```
openssl genrsa -aes256 -passout pass:%PASSWORD% -out ca-key.pem
 4096
openssl req -passin pass:%PASSWORD% -new -x509 -days 365 -key
 ca-key.pem -sha256 -out ca.pem
```

3. Generate the server keypair using the certificate authority key.
   The following commands are an example. Note that your configuration might vary.

```
openssl genrsa -out server-key.pem 4096
openssl req -subj /CN=%HOSTNAME% -new -key server-key.pem -out
 server.csr
echo extendedKeyUsage = clientAuth > extfile.cnf
openssl x509 -passin pass:%PASSWORD% -req -days 365 -in
 server.csr -CA ca.pem -CAkey ca-key.pem -CAcreateserial -out
 server-cert.pem -extfile extfile.cnf
```

4. Create the client keypair using the certificate authority key.
   The following commands are an example. Note that your configuration might vary.

```
openssl genrsa -out client-key.pem 4096
openssl req -subj /CN=%HOSTNAME% -new -key client-key.pem -out
 client.csr
echo subjectAltName = DNS:%HOSTNAME%,IP:%SERVERIP%,IP:127.0.0.1
 > extfile.cnf
openssl x509 -passin pass:%PASSWORD% -req -days 365 -in
 client.csr -CA ca.pem -CAkey ca-key.pem -CAcreateserial -out
 client-cert.pem -extfile extfile.cnf
```

5. Import the CA Public Key and Client Keypair to a Java keystore.
   This instructions are an example Your configuration might vary.

   Create the keystore file and create a password for it (and save for later use): `keytool -genkey -keyalg RSA -alias dse -keystore my.keystore`

   Delete a default entry from the keystore file: `keytool -delete -alias dse -keystore my.keystore`

   Note: This entry is auto-generated, s.o is not needed.

   Import the CA public key to the keystore: `keytool -import -keystore my.keystore -trustcacerts -alias ca -file ca.pem`

   Import the client keypair.

   Note: You are creating a new pkcs12 keystore file and importing the keypair to it. Then copy the contents to your original keystore file.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- `openssl pkcs12 -export -name clientkeypair -in client-cert.pem -inkey client-key.pem -out clientkeypair.p12`
- `keytool -importkeystore -destkeystore my.keystore -srckeystore clientkeypair.p12 -srcstoretype pkcs12 -alias clientkeypair`

Now that you have added all of the certificates to the keystore file, save the file **my.keystore** for later, as it will be uploaded to the ServiceNow instance. In addition, be sure to remember the password that you entered when prompted to create the keystore file; you will need to enter that into a form in the ServiceNow instance.

Configure Docker for Headless Browser setup in Microsoft Windows

Configure Docker Server to authenticate all requests.

Before you begin

Complete Step 2: Generate certificates for Headless Browser setup for Microsoft Windows

Role required: admin on your ServiceNow instance and local administrator on the host machine.

About this task

After creating your client and server keys, now you configure the Docker Server to authenticate all requests using those keys, and expose the Docker API on Port 2376.

Procedure

1. Configure Docker to use the certificates you generated in Step 2.
2. Find or create the `C:\ProgramData\docker\config\daemon.json` file.
3. Add the following properties to the daemon.json file.
   Be sure to replace items in these commands with the correct paths to your certificates:

   Note: The double slashes are important to copy exactly.

   ```
   {
   "tlscacert": "C:\\Users\\Administrator\\certs\\ca.pem",
   "tlscert": "C:\\Users\\Administrator\\certs\\server-cert.pem",
   "tlscert": "C:\\Users\\Administrator\\certs\\server-cert.pem",
   "tlskey": "C:\\Users\\Administrator\\certs\\server-key.pem",
   "tlsverify": true,
   "hosts": ["tcp://0.0.0.0:2376", "npipe://"]
   }
   ```

   To learn more, see Configure the Docker daemon.
4. In administrator PowerShell, run `restart-service *docker*`

Create the Docker image and containers for Headless Browser setup in Microsoft Windows

Pull the Docker image from the Public Registry.

Before you begin

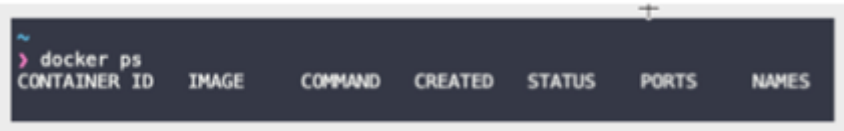
Complete Step 3: Configure Docker for Headless Browser setup in Microsoft Windows

Role required: admin on your ServiceNow instance and local administrator on the host machine.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. In an administrator command line, run `docker ps` to verify that Docker is working.
   Your results should look like this:



2. Pull the latest Docker image from the GitHub repo ATF Headless Runner.
   This command is an example, as these images are subject to change:
   `docker pull ghcr.io/servicenow/atf-headless-runner:win-1.0.0`

> **Note:** The instance-to-image compatibility makes sure that the automation script inside the Docker image is compatible with the instance code. Elements such as the user interface might change over time to support new features or upgrades.

Add secrets to Docker for Headless Browser setup in Microsoft Windows

Create a Docker secret that stores the password of the ServiceNow user who will log into the instance to execute the tests. Docker Secrets is a feature of ServiceNow® for securely storing the passwords that will be used in containers.

Before you begin

Complete Step 4: Create the Docker image and containers for Headless Browser setup in Microsoft Windows

Role required: Go to your ServiceNow instance and create a new user to be used by the Docker container to log in. Give this user the roles of **admin** or **atf_test_admin**. Save the password to be used in Step 2 of the following procedure.

About this task

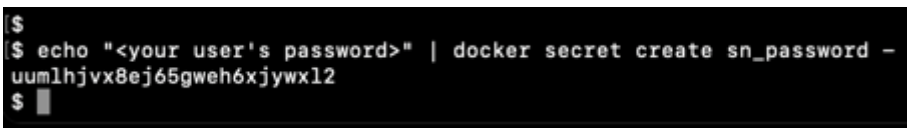
Inside the Docker container is an automation script that opens a web browser, logs into the instance, and opens the Client Test Runner page. In order to log into the ServiceNow instance, you will need a user password. In this step you use a Docker feature called Docker Secrets so you can securely store passwords. Then when you run containers, the password is automatically available to log in to your instance.

Procedure

1. In the admin command line, enter `docker swarm init`
2. In the admin command line, enter `echo <your user's password> | docker secret create sn_password -`

> **Note:** Replace `<your user's password>` with the user's ServiceNow password.

Your results should look like this:



Result

The result is your secret ID, which you must save for later use. The secret ID will be added to the ServiceNow instance in the sys_property `sn_atf.headless.secret_id`.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Set up instance for Headless Browser in Microsoft Windows

Step 6 in the Microsoft Windows setup for the ServiceNow® Headless Browser for ATF: Set up your instance so it can support the Headless Browser.

Before you begin

Complete Step 5: Add secrets to Docker for Headless Browser setup in Microsoft Windows.

Role required: admin on your ServiceNow instance and local administrator on the host machine.

This task: Now that you have configured the host machine, you will set up the instance so that it successfully communicates and authenticates with the host machine.

Procedure

1. Create user:
   a. Navigate to **All** > **User Administration** > **Users** and select **New**.
   b. Create a user - User ID, which can be whatever you want, as well as a user name and password. The password should be the same as the one you created for your Docker Secret container.
   c. Add the role **atf_test_designer** for this user. To learn more, see the User Roles section in the Exploring Automated Test Framework topic.
2. Create certificate:
   a. Navigate to **System Definition** > **Certificates** to open the **sys_ certificate** table. Create a new certificate (the name can be whatever you prefer):
      - **Type**: Java Key Store
      - **Password**: Password for the keystore that you created in Step 2 (Generate certificates for Headless Browser setup for Microsoft Windows).
   b. Select the Attach (paper clip) icon to attach the keystore file you saved earlier to this record.
   c. Select **Submit**.
   d. Click **Validate certificate** and confirm that the success message displays.
3. Create protocol profile:
   a. Navigate to **System Security** > **Protocol Profiles** to open the **sys_protocol_profile** table. Create a new protocol profile record:

Option	Description
Protocol	One word, lowercase: “docker”
Default port	2376 (or the one you chose to use)
Keystore	Docker host keystore

   b. Select **Submit**.
4. Create a Docker spoke connection:
   a. Navigate to **Connections & Credentials** > **Connection & Credential Aliases** to open the **sys_alias** table.
   b. Select the alias with the name **Docker**.
   c. Under the Connections related list, select **New**.
   d. Fill in these fields:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- **Name**: Any text you prefer
- **Credential field**: Leave blank

e. Select the **URL Builder** check box.

f. Select the **Mutual authentication** check box.

g. In the **Protocol profile** field, select the protocol profile you created earlier.

h. In the Host field, add the IP address or Host name of your server.

i. Select **Submit**.
The Connection URL is automatically created by the system.

5. Modify properties:
   - `com.glide.communications.httpclient.verify_hostname`: false
   - `com.glide.communications.trustmanager_trust_all`: true

> **Warning:** By default, the *com.glide.communications.trustmanager_trust_all* property is set to **false**. The ServiceNow AI Platform trusts only certificates that it can verify against the JVM certificate store. Self-signed and enterprise-signed certificates are not trusted.

You need to do this only when using self-signed certificates. To learn more, see Generate certificates for Headless Browser setup for Microsoft Windows.

Configure Automated Test Framework (ATF) for Headless Browser in Microsoft Windows

Step 7 in the Microsoft Windows setup for the ServiceNow® Headless Browser for ATF: Configure ATF with properties.

Before you begin

Complete Step 6: Set up instance for Headless Browser in Microsoft Windows

Role required: admin on your ServiceNow instance and local administrator on the host machine.

Now that your connection and authentication are set up, you will configure ATF with several properties so that it can start containers successfully on the host machine.

Procedure

1. In your instance, navigate to **ATF** > **Administration** > **Properties**.
2. Enable the top two properties: **Enable test/test suite execution** and **Enable scheduled test suite execution**.
3. Scroll down to the **Headless Runner Properties** section.
4. Enable the top check box.
5. Enter the following values in the Headless Runner Properties form:

Form Label	Property Name	Value to Imput
The user account used to login from the headless Client Test Runner and begin the tests	sn_atf.headless.username	Username of the integration user that you created in Add secrets to Docker for Headless Browser setup in Microsoft Windows

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Form Label	Property Name	Value to Input
The Docker secret ID that has the password stored of the user account	sn_atf.headless.secret_id	Docker "secret ID" that you obtained earlier, as well as the Docker secret name that you created in Step 2 of the Add secrets to Docker for Headless Browser setup in Microsoft Windows procedure.
The Docker image that is used for headless Client Test Runner	sn_atf.headless.docker_image_name	Image name with tag that you downloaded: `ghcr.io/servicenow/atf-headless-runner:<tagname>`

Note: If you forgot the secret ID, go to Windows host and run the command docker secret list.

6. At the **The absolute path of the secret file on a docker container** field, enter: `C:\ProgramData\Docker\secrets\<YOUR SECRET NAME>`
7. Leave the rest of the fields as they are.
8. Select **Save**.

Verify Headless Browser procedures for ATF in Microsoft Windows

Verify that your Headless Browser setup procedures have been successful.

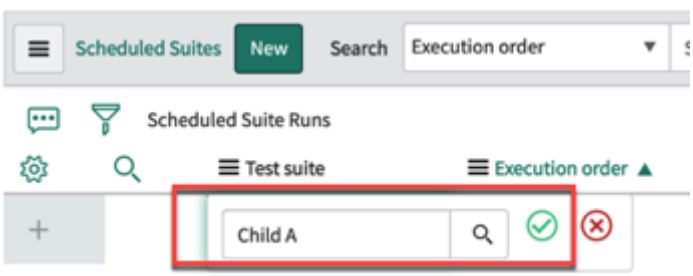
Before you begin

Complete Step 7: Configure Automated Test Framework (ATF) for Headless Browser in Microsoft Windows

Role required: admin

Procedure

1. Go to **ATF Schedules** and next to the Suite Schedules link, select **New**.
2. In the Run box select **On Demand**.
3. Select **Save**.
4. In the Scheduled Suite related list, add a new record for your test suite by double-clicking the **+** sign and choose your browser name of **Chrome** or **Firefox**.
   Use **Child A** as a good example to run.



5. Set the OS name to Windows.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

6. Select **Execute Now**
7. Verify that your test passed.
   When your verification is successful, any suites with UI tests run by scheduled suites or via CICD now automatically create headless test runners without the need to manually open the "scheduled client test runner" page.

Headless Browser system properties

Below is a table of the properties you must have as you set up the ServiceNow® Headless Browser for Automated Test Framework.

System properties

Property name	Type	Default value	Purpose
sn_atf.headless.browser_options	string	"--no-sandbox,--disable-gpu"	The CLI options are passed to the browser on startup
sn_atf.headless.default_browser	string	Chrome	Default browser used when starting up headless requests
sn_atf.headless.default_os	string	Linux	Operating system of host machine
sn_atf.headless.docker_image	image_name		Name:tag of the Docker image on the host machine
sn_atf.headless.docker_init_window_seconds	integer	60	If the Docker container fails to start up twice in this amount of time, the service will not attempt to restart
sn_atf.headless.enabled	true/false	false	Determines whether the instance attempts to create headless client test runners for scheduled UI test runs
sn_atf.headless.heartbeat_enabled	true/false	true	When property is true, Docker container sends a REST API request every minute to the instance to check that the sys_atf_agent is still online, and stops the container if the record status is "offline" or the record no longer exists.
sn_atf.headless.heartbeat_uri	string	/api/now/atf_agent/online	The URL of the heartbeat endpoint so container can

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

System properties (continued)

Property name	Type	Default value	Purpose
			verify browser is still responsive
sn_atf.headless.images_to_verify	true/false	false	When this property is true, the instance verifies that the requested Docker image:tag is present on the host before test execution
sn_atf.headless.login_button_id	string	sysverb_login	The HTML ID of the login page Submit/ Login button
sn_atf.headless.login_page	string	login.do	URL of login page that the browser navigates to
sn_atf.headless.password_field_id	string	user_password	The HTML ID of the input field for the password on the Login page
sn_atf.headless.request_timeout_sec	int	200	Number of seconds that HTTP requests being sent to the Docker host have until timeout
sn_atf.headless.retry_count	int	10	Number of times the instance checks for agent coming online before cancelling the test run
sn_atf.headless.runner_banner_id	string	test_runner_banner	The ID of the element verifying that client test runner page loaded correctly
sn_atf.headless.runner_url	string	atf_test_runner.do?sysparm_nostack=true&sysparm_scheduled_tests_only=true&s	URL of the scheduled client test runner page and its query parameters
sn_atf.headless.secret_gid	string	1000	The GID of the Docker container default user
sn_atf.headless.secret_id	string		The ID of the Docker secret on host machine
sn_atf.headless.secret_name	string		Name of the Docker secret on host machine

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

System properties (continued)

Property name	Type	Default value	Purpose
sn_atf.headless.secret_path	string	/run/secrets/<secret_name>	Path where Docker secret file exists: (Learn more: https://docs.docker.com/engine/swarm/secrets/#how-docker-manages-secrets)
sn_atf.headless.secret_uid	string	1000	The UID of the Docker container default user
sn_atf.headless.service_cleanup_exclude_list	string		Service ID exceptions that should NOT be deleted during the instance service cleanup job. This job runs every night and deletes any services that are on the host and past their expiration time.
sn_atf.headless.service_test_step_delete	true/false	false	On completing of a test execution, if this property is true the instance will NOT send the service delete requests thus keeping the service and containers around on the host. Useful for debugging.
sn_atf.headless.timeout_in_mins	integer	1440	Number of minutes before Docker service automatically shuts down
sn_atf.headless.user_field_id	string	user_name	The HTML ID of the input field for the username on the Login page
sn_atf.headless.username	string		Username of user who logs in to the instance
sn_atf.headless.validation_id	string	headless_vp_validation	The ID of the element confirming verification page has loaded
sn_atf.headless.validation_page	string	atf_headless_validation_page	The URL of the validation page on the instance

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

System properties (continued)

Property name	Type	Default value	Purpose
sn_atf.headless.vp_has_role	string	headless_vp_has_role	The ID of the element verifying that the user has correct roles
sn_atf.headless.vp_success	string	headless_vp_success	The ID of the element signifying entire page loaded correctly

Related topics

Headless Browser for Automated Test Framework

Exploring Automated Test Framework

Headless Browser troubleshooting

These tips can help you troubleshoot your Linux or Microsoft Windows setup of the ServiceNow® Headless Browser for Automated Test Framework.

There are three basic areas to examine when troubleshooting your Headless Browser setup.

Docker container

Error logs: When the headless test completes, if there are errors they are generated in the Docker container. Whether the operation fails or succeeds, the container's **stdout/stderr** logs are placed in the **sn_atf_docker_service** table.

Headless client test runner did not start in the time allotted message: This message generally means an error occurred in the Docker container while it was initializing or starting up. This may indicate something is incorrect in your Docker container setup. Navigate to the **sn_atf_docker_service** table to read the logs and see the error message.

Instance

Error caught running Docker flow when starting the ATF tests messages: Follow the URL address to find the flow context, which contains the error logs. For example, the above error message can display when the instance can't access the Docker host.

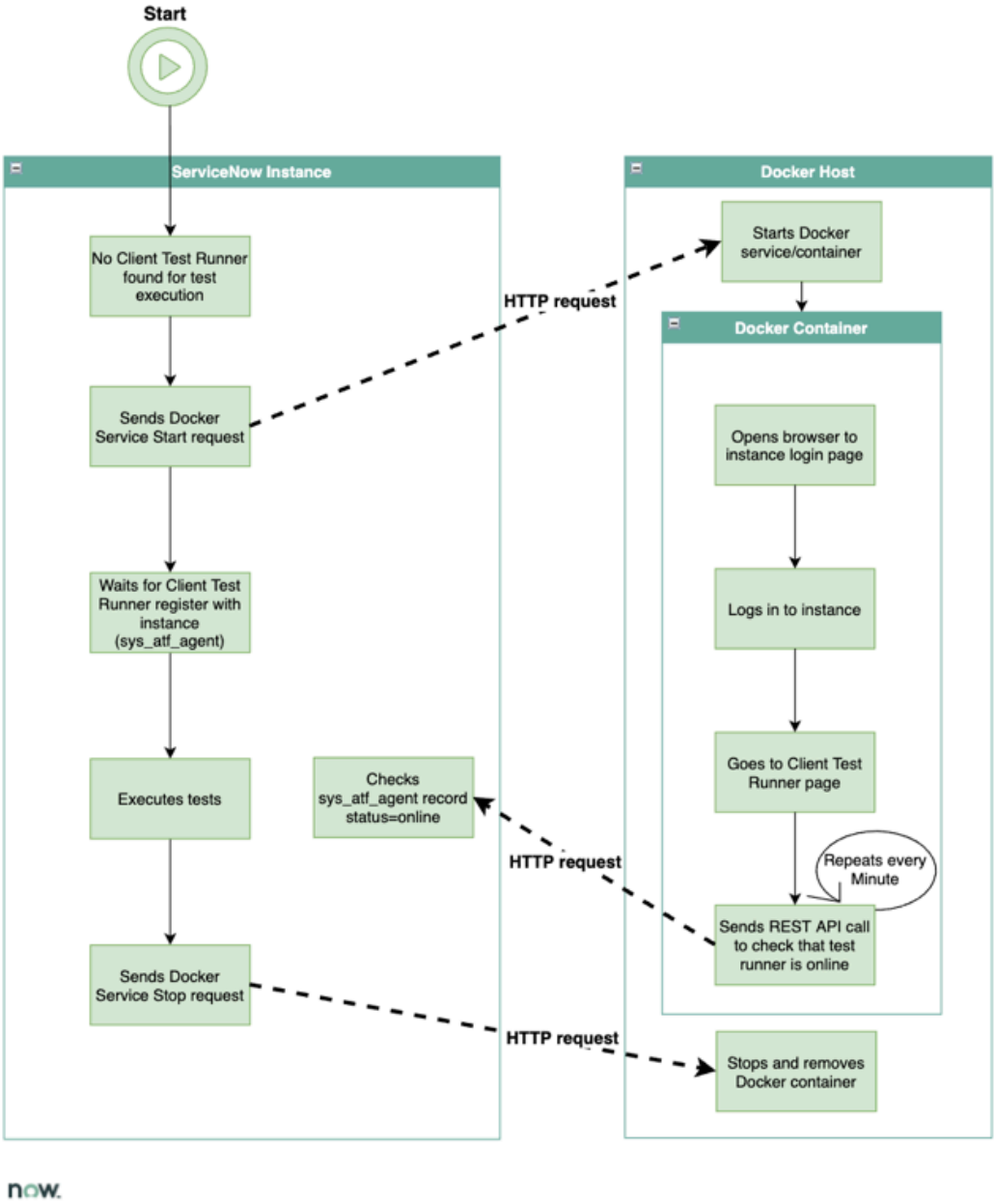
Network errors

Firewalls: Make sure your firewalls are set up so that the instance can access the host and port and the server can access the instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Headless Browser action flow diagram

Example flow diagram



Automated Test Framework (ATF) test building and execution

Build and execute the ATF tests and test suites using the information in this section.

Building and running automated tests with the Automated Test Framework

Basic tasks in the Automated Test Framework.

To build and run any test, you always perform certain operations:

- Create a new automated test
- Add steps to an automated test
- Run an automated test

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- View the progress of automated tests
- View test results and automated test results

You perform other operations often, but not necessarily for every test.

- For tests involving form steps, you might need to View results screenshots from an automated test.
- For some tests, you might need to know about Passing data from one automated test step to another.
- Some steps frequently occur in the same sequence in many different tests, so you can Add a predefined list of steps (template) to an automated test.

Create a new automated test

Create a named automated test containing a series of steps to execute.

Before you begin

Role required: atf_test_admin or atf_test_designer

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Tests**.
2. Click **New**.
3. On the Test new record form, enter a **Name** for your test.
   The system identifies this test by this name wherever it displays a list of tests (for example, in the Tests module).
4. **Optional:** Enable parameterized testing to run a test multiple times with different test data for each run.
   For more information, see Parameterized tests.
5. Enter a **Description** for your test.
6. Click **Save**.
   The system creates a new test record and returns to the list of tests.

What to do next

Add steps to an automated test.

Add a predefined list of steps (template) to an automated test

With test templates you can add a predefined list of steps to a test. Any list of steps that follows a set pattern makes a good candidate for a template.

Before you begin

You must have created the test to which you want to add steps.
Role required: atf_test_admin or atf_test_designer

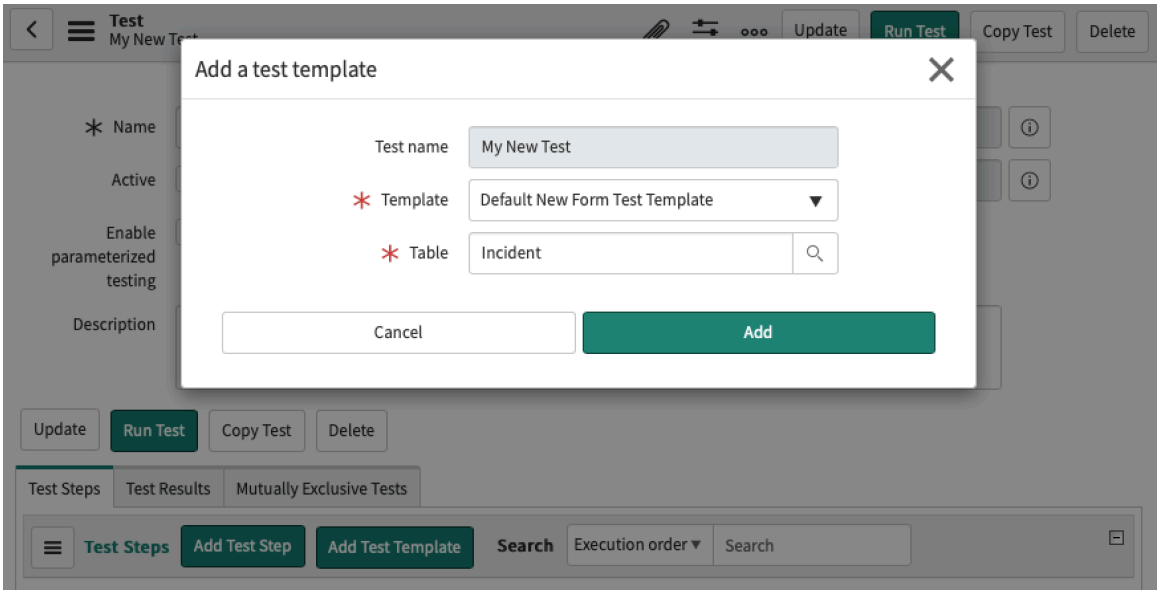
About this task

Many tests follow similar patterns. One common pattern, for example, is to open a form, set some field values, validate some field values, click a UI action, and submit the current form. If a template exists containing these steps, you can add them to a test all at once. The Automated Test Framework comes with default templates. You can also create custom test templates.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Tests**.
2. Click the row for the test to which you want to add steps.
   The system displays the Test form.
3. On the **Test Steps** related list, click **Add Test Template**.
   The system displays the **Add a test template** dialog.



4. From the **Table** field, select the table you want to test with these steps.
5. From the **Template** field, select the template containing the steps you want to add.
6. Click **Add**.
   The system adds the template steps to the test. It also adds to the test description a set of instructions on how to complete the test from this template.
7. Following the instructions in the test description, edit each step added by the template to include the necessary information.

What to do next

Proceed to edit or save the test as you normally would.

Related topics

Create an automated test steps template

Add steps to an automated test

Create a series of steps for an automated test to run in a specified order.

Before you begin

You must create a test before you can add steps to the test.
Role required: atf_test_admin or atf_test_designer

About this task

This procedure lets you add one step at a time. You can also add a batch of steps at once. For more information, see Add a predefined list of steps (template) to an automated test.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Tests**.
2. Click the test that should contain the new test steps.
3. In the Test Steps related list at the bottom of the Test form, click **Add Test Step**.
4. From the left panel of the Add Test Step form, select the test step category, or select **All Steps** to view all available test steps for all test categories.
   For example, if selecting a specific test step category, select **Form** for form-related test steps, **Application Navigator** to test application menu or module visibility, or **Server** for tests executed on the server.

   Note: When selecting certain types of server tests, you can specify whether a positive or negative test result constitutes a successful test outcome. For more information, see Assert type in Exploring Automated Test Framework.

5. Click the type of test step you want to select.
   For example, if you select the **Form** category, select **Set Field Values** to set the field values on a form.

   Start any sequence of steps that work with forms with the **Open a new form** or **Open an existing form** step. Close with the **Submit form** step.

6. If applicable, from the **Insert after** drop-down list, select the step that you want to precede this step.
   If this is the first step in a test, the **Insert after** drop-down list does not appear.
7. Click **Next**.
8. From the **Table** list on the Add Test Step form, select the table that you want to test in this step.
9. **Optional:** In the **Execution Order** field, enter an integer representing the order in which you want the test to execute this step.
   For more information on **Execution Order**, see Edit automated test step order.
10. Fill in the fields that apply to this step.

    For instructions, see Automated Test Framework (ATF) test step categories.

    Some steps return output values that you can pass to the inputs for a subsequent step. For more information, see Pass values from one automated test step to another.

11. Click **Submit**.
    The system creates the step and displays the test record.
12. Repeat Steps 3 through 11 to add additional steps for this test.

What to do next

Run an automated test.

Related topics

Automated Test Framework use case examples

Change automated test step

If necessary, edit a test step after you create it.

Before you begin

Role required: atf_test_admin or atf_test_designer

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Tests**.
2. Click the row containing the test you want to edit.
3. On the **Test Steps** related list, click the row containing the step you want to edit.
   The system displays the **Set Field Values** form.
4. Edit the fields you want to change.
5. Click **Update**.

Edit automated test step order

By default, steps execute in the order in which you created them. You can change this order by editing the **Execution Order** field.

Before you begin
Role required: atf_test_admin or atf_test_designer

About this task
By default, the system assigns the value 1 to **Execution Order** for the first step created for a test. When you add a step, the system assigns it the next-highest available integer value. In other words, if the highest **Execution Order** for any step in the test has the value 7, the system assigns 8 to the new step. By editing these values, you can change the order in which the test executes the steps.

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Tests**.
2. Click the row containing the test you want to edit.
   The system displays the **Test**form.
3. In the **Test Steps** related list, edit the values in the **Execution order** column to determine the new order for the steps.
4. Click **Update**.

Copy automated test

Copy an existing test, which you can then re-name and modify.

Before you begin
Role required: atf_test_admin or atf_test_designer

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Tests**.
2. Click the row containing the test you want to copy top open it.
3. Near the top-right of the Test record form, click **Copy Test**.
   An annotation displays confirming that the system has copied the test. After you dismiss this annotation, the system displays a new test record identical to the copied record, with the exception of the **Name**.
4. In the **Name** field, enter the name you want to assign to this new test.
5. Edit the test steps as desired and proceed as you would for any new test.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

6. When you are finished making changes, click **Update**.

> **Note:** Scope management in ATF tests helps identify and restrict copying of tests in other scopes. If you want to copy a test, you must be in the same scope as the test. See Application Scope for more information.

Run an automated test

After creating an automated test, run it on a non-production instance.

Before you begin

You must have created the test you want to run.
The test execution property must be enabled. You must have an admin or atf_test_admin role to do so.

> **Note:** The test execution property is disabled by default to prevent running tests on a production system. Run tests only on development, test, and other sub-production instances.

Role required: atf_test_admin, atf_test_designer, or admin

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Tests**.
2. If necessary to view the tests list, click **Tests**.
3. Click the row containing the test that you want to run.
   The system displays the **Test** form.
4. Click **Run Test**.

> **Note:** If the test execution property is not enabled, the **Run Test** button does not appear. In this case, see the annotation at the top of the form, and click the link to enable running tests.

If the test includes a form step (any step involving a UI), or other kinds of UI test steps, the Pick a browser dialog appears before executing the tests. Use it to choose among any currently running test clients, or start a new runner. For more information, review Browser recommendations for all tests and suites. If the test only includes server test steps, the system executes the tests without displaying the Pick a Browser dialog.

What to do next

Monitor the progress of the test in the Run Test progress dialog. When complete, click **Go to Results** (on the Run Test progress dialog) to display the **Test Results** list, where you can view and analyze test results.

> **Note:** If your test creates data, the system rolls back that data after all steps in the test complete.

Related topics

Cancelling automated tests and test suites

Implementing breakpoints

Breakpoints allow you to pause your test at any step of a test run in order to troubleshoot and test authoring.

Breakpoints are useful when a test fails. You can add breakpoints to the test steps to debug the test. You can also look at the test steps again to find the reasons of the test failure.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: Breakpoints are user-specific. You can see only your self-created breakpoints.

Rollback of test data

Test runs can also be paused immediately before rollback of test data. Check the **Pause before rollback** option to pause the test just before rollback of test data happens after the completion of the test. See Debug an automated test using breakpoints to learn more about using rollback option in a test run.

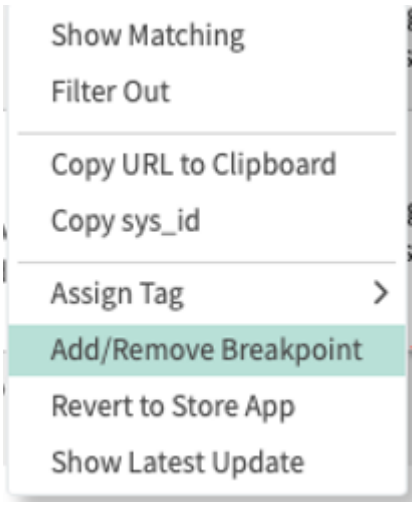
Debug an automated test using breakpoints

Pause a test to troubleshoot failures or unexpected behavior by adding a breakpoint for a particular test step.

Before you begin
Role required: admin

Procedure

1. Navigate to **All** > **Automated Test Framework (ATF)** > **Tests**.
2. Select the test you want to run.
3. Select Test Steps related list to list out all the test steps for the selected test.
4. Right-click the test step where you want to add or remove a breakpoint.
5. Select Add/Remove Breakpoint from the list.

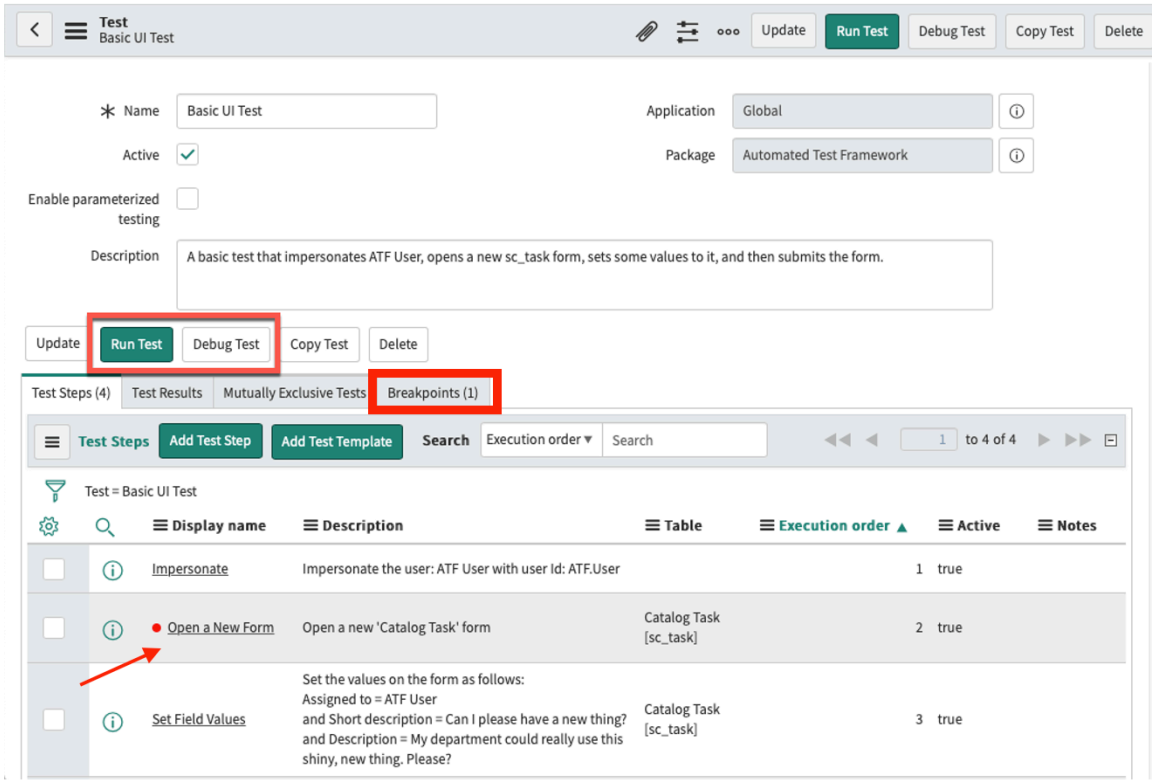


The test step gets marked for the modified breakpoint.

Note: The breakpoint mark on a step states that if the breakpoint is active and the test is run using **Debug Test**, the test pauses just before the breakpoint test step. The breakpoints are not considered if you click **Run Test** to run the test.

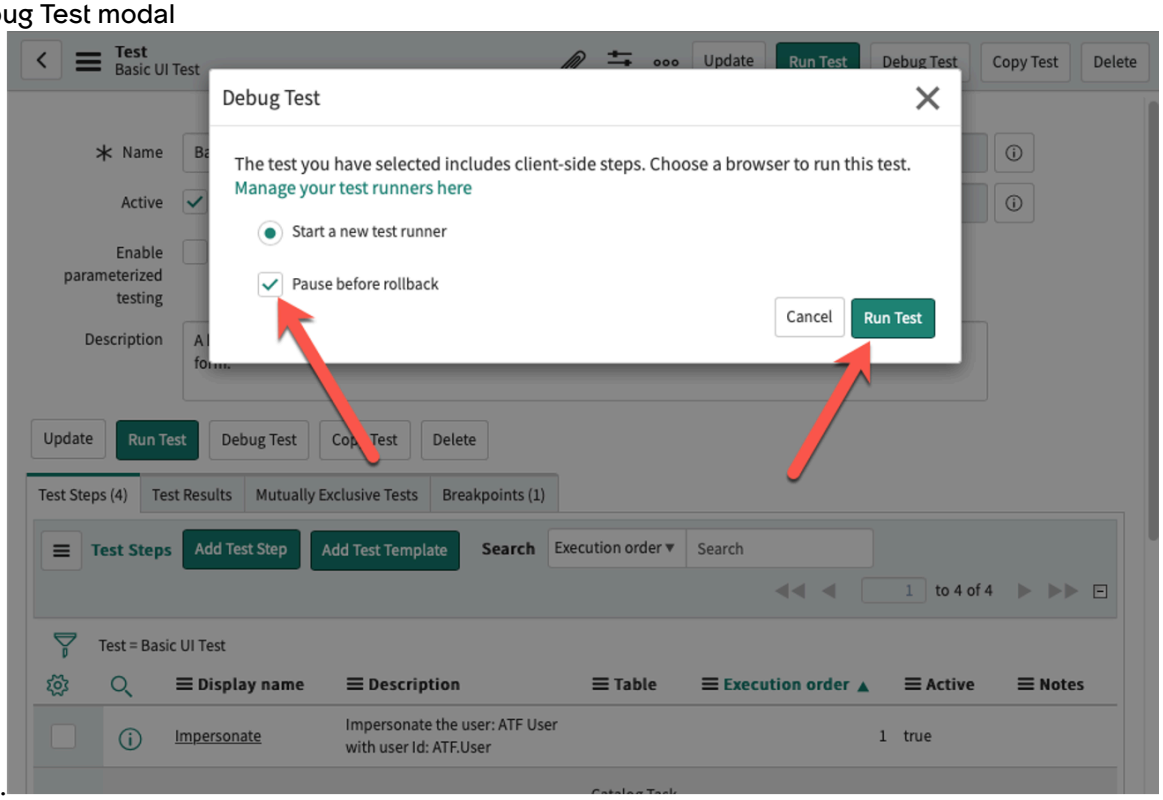
The breakpoints you set on a test are user-specific. Other users can't see and modify your breakpoints on a test, and vice versa.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Note: You can set multiple breakpoints for a test but can have only 1 breakpoint every test step.

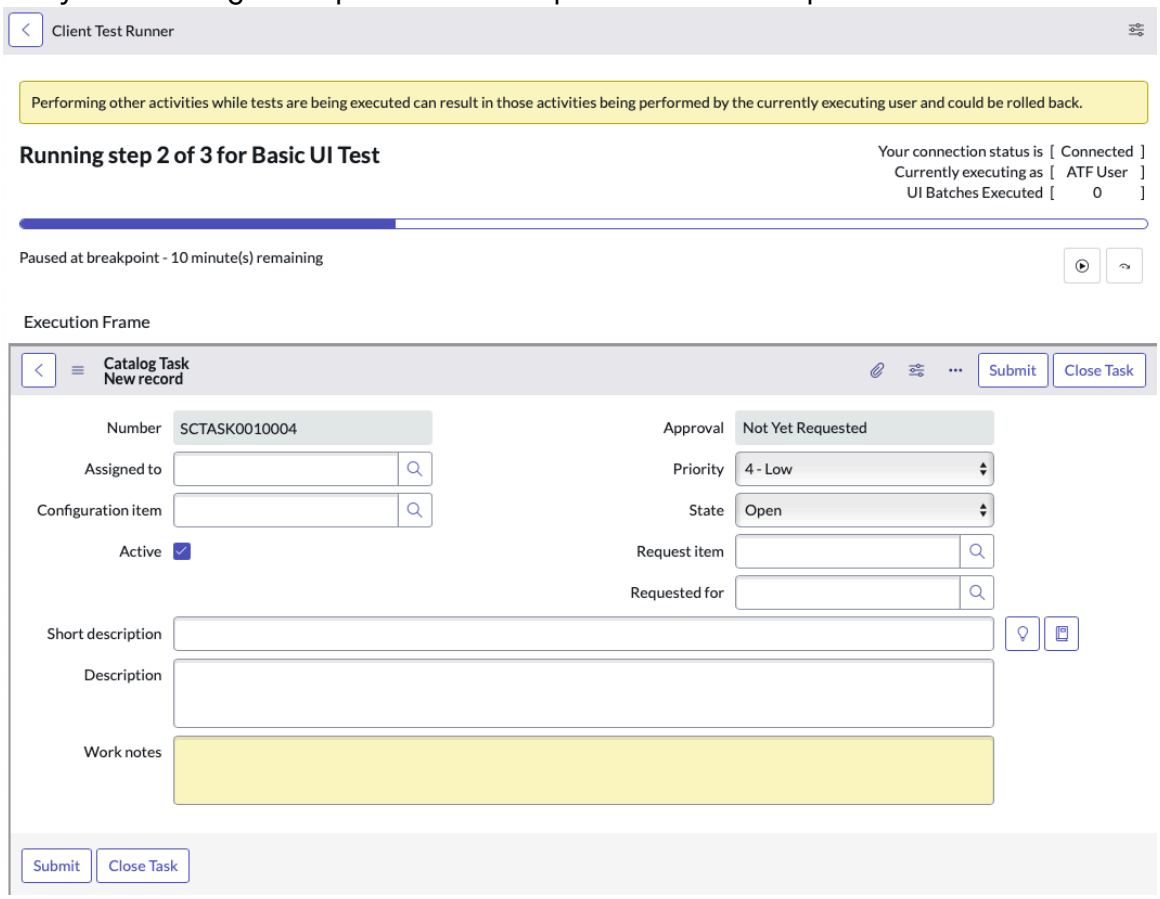
6. **Optional:** Click the Breakpoint related list to show all the test steps within a test that have breakpoints.
7. Click **Debug Test** to run the test with breakpoints.
   The Debug Test modal displays.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Safari – the browser used to run the client-side steps of the selected test
- Pause before rollback – Option to pause the test just before rollback happens after the completion of the test

8. Analyze and debug the steps at which breakpoints have been implemented.



Note: Every breakpoint causes the running of the test to pause for 10 minutes. You can debug the test step within the allocated 10 minutes and then the test starts running again until it reaches the next breakpoint.

The following options display only when the test is paused for a breakpoint:

- Continue: Test execution continues until the next breakpoint.
- Step over: Test pauses at every step. If you don't want to pause in the current step, select the Step over option to jump to the next step.

View test results and automated test results

View test results from completed test and test suite runs. Carefully consider the results of automated test runs and perform any corrective actions required to resolve any revealed issues.

Before you begin

Role required: none

About this task

Test and test suite results show how long it took the system to execute a test and which steps failed. They can include screenshots of form steps. You can also view reports comparing different runs of the same test suites.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

You have multiple options for navigating to the test results, depending on where you are in the user interface. For example, if the Run Test progress dialog is displayed, you can select **Go to Result**. This procedure enables you to view test results from any location in the user interface.

Procedure

1. Navigate to **Automated Test Framework** > **Test Results**.
2. Click a row to access the results for a specific test.
   The system displays the Test results record.
3. View step results for the selected test.
   a. Click the Step Results related list.
   b. Scroll down and click the row for the step result you want to view.
   The system displays the Step results record.
4. View test logs for this test result.
   a. Click the Test Log related list.
   b. Scroll down and click the row for the test log you want to view.
   The system displays the Test logs record.

View automated test results

When viewing test results, step results or test logs, you can allow client errors as ignored or warning entries in the Allowed Client Errors table. This allows test executions to continue past client errors in future test runs.

View results screenshots from an automated test

If the test has a UI component, the system takes screenshots of the UI. View these screenshots to gain further insight into the test results.

Before you begin

You must have run the test whose results screenshots you want to view.

Note: For best results with screen shots, leave the browser zoom level set to 100%.

Role required: atf_test_admin or atf_test_designer

Procedure

1. Navigate to **Automated Test Framework** > **Test Results**.
2. Click the row containing the test whose results you want to view.
   The system displays the **Test Results**form.
3. View a screenshot taken during the test by finding the screenshot you want in the attachments list.
   Screenshots are named with the word **screenshot** followed by the timestamp (always in UTC time) for when the system recorded the shot. You can match the screenshot to the test step by comparing the step and screenshot timestamps.

View automated test suite results

View results from an automated test suite.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

About this task

Note: You have multiple options for navigating to the test suite results, depending on where you are in the user interface. For example, if the Run Test progress dialog is displayed, you can click **Go to Result** instead. The procedure described here enables you to view test suite results from any location in the user interface.

Procedure

1. Navigate to **Automated Test Framework** > **Suite Results**.
2. Click the row containing the test whose results you want to view.
   The system displays the Test suite results record.

Related topics

Identify and resolve client errors

Allow client errors from test results

Allow client errors from step results

Allow client errors from the test logs

Identify and resolve client errors

Identify client errors and resolve them in client-side scripts.

Before you begin

Role required: admin

About this task

When client errors occur, the Automated Test Framework fails the test on the step that was executing when the error occurred. Even though client-side scripts can fail silently on JavaScript errors while procedures are executing, the error may still impact data, and the procedure being executed. The Automated Test Framework considers these errors as validation failures.

Procedure

1. Navigate to **All** > **Application** > **Automated Test Framework** > **Tests** and run an Automated Test Framework test that interacts with a form.
2. In the test results for this test, check for a step result with the following summary:

   ```
   This step failed because the client error 'DETAILED ERROR
   MESSAGE' was detected on the page being tested. See failing Test
   Logs. To ignore these errors in the next test run, use ‘Add all
   client errors to warning/ignored list’ links.
   ```

   This step result appears only on a step that interacts with the UI.

3. To identify and resolve these script errors, open the developer tools browser console on the Client Test Runner page.

   Note: For information about how to open the browser console, see the following article: http://webmasters.stackexchange.com/questions/8525/how-to-open-the-javascript-console-in-different-browsers

4. If you can see the error, try to identify and troubleshoot the problematic client-side script, which may be on any of the following base system tables:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- ServiceNow Client Script
- UI Action
- UI Macro (HTML script)
- UI Page (HTML script)
- UI Policy
- UI Script
- Tables that extend the preceding base system tables

5. Review the script for errors and once you've fixed them, run your test again.

What to do next

Determine the source of the client error by reviewing the script version history. If you customized a base system script, it's possible that the script has new versions that were skipped during upgrade.

Example client errors

There are several types of common client error.

Client JavaScript errors

When a client script causes an error, the browser console displays an error similar to the following example:

```
**************************************************
A script has encountered an error in render events
TypeError: Cannot read property 'id' of undefined
Script ends. Continuing happily
**************************************************
[00:00:00.002] onLoad Modify Comments Label
```

In this example, the client script Modify Comments Label caused the error.

Other client script errors

Any other type of script error reports directly to the browser console with any formatting upon occurrence.

```
TypeError: callbacks(id) is undefined
```

Script resource links by Sys ID

In some cases, the console error provides a link to the script resource file using its Sys ID. Following this link may give context to which script had executed it.

```
Uncaught ReferenceError: myobj is not defined
    at incident.do?sys_id=12345678901234567890123456789012
(anonymous)
 @ incident.do?sys_id=12345678901234567890123456789012  <----
 LINK
```

Script access permissions

While identifying problematic scripts, be sure the script has permission to access data. Check:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Access control rule permissions for tables and fields.
- Application access permissions if the script accesses applications in a private scope.
- Domain separation permissions if domain separation is configured.

Related topics

Allowed client errors

UI test steps

Test user interfaces by mimicking user actions and interacting with the visible components of a page.

Client test runner dependency

UI test steps require an active client test runner to act directly on the visible components of a page. A tester must manually start one or more client test runners for UI testing. Test designers can schedule selecting an open client test runner from a test suite. See Working with client test runners.

Intelligent wait mechanism

UI test steps have an intelligent wait mechanism triggered by UI changes such as clicking a component or setting a value. The wait mechanism requires the UI change to complete before the next UI test step can proceed. Test designers do not need to manually add wait mechanisms between UI test steps.

Custom UI test steps

Test customized user interfaces such as UI pages and UI macros by retrieving their HTML and JavaScript page components and identifying the test actions they support.

Note: For new test development, use the Run UI Test Script test step instead of Custom UI test steps.

Custom UI test steps require the Automated Test Framework to retrieve and identify the testable components from a target web page.

Note: Next UI Experience pages are not supported by the Custom UI test steps, including but not limited to Configurable Workspaces (except for Form steps) and UI Builder.

Testable components

Testable page components consist of standard HTML and JavaScript with these characteristics.

Are set or clicked by user interaction

Testable page components allow users to set a value or click them.

Are accessible from the Document Object Model (DOM)

Testable page components are accessible from the DOM and support JavaScript manipulation of the DOM. Custom UI test steps cannot access page components in the shadow DOM.

Are accessible to JavaScript

Testable page components are accessible to JavaScript. Custom UI test steps cannot access page components that interact directly with the operating system such as file fields. Custom UI test steps cannot access page components that display non-HTML content such as Excel or PDF files.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

New browser tabs or windows are not supported by Custom UI test steps.

Are not excluded from custom UI testing

Testable page components are not excluded from custom UI testing. Automated Test Framework excludes page components that are already testable by other test step categories and also excludes page components associated with ServiceNow AI Platform features.

Are accessible to the Page Inspector

Testable page components must return results when viewed from the Page Inspector. Test designers can use the Page Inspector to identify the testable components of a page.

Examples of testable page components include these UI elements.

- Buttons
- Links
- Page text
- UI controls
- UI macros
  - ui_date
  - ui_date_time
  - ui_reference
- UI pages
- Wizards

Examples of untestable page components include these UI elements.

Example untestable page components

Reason untestable	Untestable page components
Are not settable or clickable	Hidden controls
	HTML comments
	HTML layout elements such as div, section, and span.
	HTML script elements
Are inaccessible from DOM	Dashboards
	Images
	Lists
	Reports
	Shadow DOM
Are inaccessible to JavaScript	Excel files
	File fields
	PDF files
Are ServiceNow AI Platform features	Flow Designer

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example untestable page components (continued)

Reason untestable	Untestable page components
	Studio
	Upgrade Monitor
Are testable by other test step categories	Form field labels
	Form field values
	Service Catalog
	Workspaces

Settable page components

A settable component is a UI element that has a dynamic value such as a text input field. Settable components support these test actions and test steps.

Test options for settable components

Page Inspector actions	Custom UI test steps
Set Component Value	Set Component Values (Custom UI)
Get Component Value	Assert Text on Page (Custom UI)
	Component Value Validation (Custom UI)
Is Component Disabled	Component State Validation (Custom UI)

Settable components have a data type that determines what values a Custom UI test step can set. For example, a page component intended to display a reference to a particular record can have a reference data type to only display Sys ID values.

Automated Test Framework allows UI developers to specify a data type to use during custom UI testing. UI developers can assign page components a data type to ensure that a test step sets a valid value. These data types are supported.

- Date
- Date Time
- Reference

See Override component data type for more information.

Clickable page components

A clickable component is a UI element that users can interact with by clicking, such as inputs of type check box or radio. Clickable components support these test actions.

Test options for clickable components

Page Inspector actions	Custom UI test steps
Click On Component	Click Component (Custom UI)
Get Component Value	Assert Text on Page (Custom UI)

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Test options for clickable components (continued)

Page Inspector actions	Custom UI test steps
	Component Value Validation (Custom UI)
Is Component Disabled	Component State Validation (Custom UI)

Clickable components do not have a data type since they do not have dynamic values.

Retrieved page components

Automated Test Framework stores a list of the retrieved page components for each custom UI page you test. Custom UI test steps display the list of retrieved components from the **Component** and **Component values** fields.

By default, the list of page components is static and is only updated when Test designers manually click **Retrieve Components**. Administrators can enable the system property *sn_atf.page_data_capture.enabled* to refresh the list of page components every time a Custom UI test step is run. Enabling this property during test design ensures that your test designers always have access to the most current list of page components. Disabling this property after test design is complete allows your tests to run faster because test steps can use the previously retrieved list of page components.

The ServiceNow AI Platform treats the list of retrieved page components as data and does not include them in update sets or applications files. When transferring tests from one instance to another, test designers must manually retrieve page components again.

Design considerations

Follow these design considerations when testing custom UI pages and page components.

Use the page inspector to identify testable page components

The page inspector determines which page components are available for custom UI testing. Page components that are unavailable to the page inspector are unavailable to custom UI testing.

Navigate to the custom UI you want to test

Use existing test steps to navigate to the target custom UI. For example, to test a Knowledge Base article, use the existing test steps to navigate to a module or to open an existing record. Most custom UI testing requires using existing test step categories as part of the test.

Use the component area to identify page components

The component area describes the HTML layout element containing the component such as a `<div>` or `<section>` element. The area helps test designers distinguish between components by providing the location in the page layout.

Test your custom UI rather than ServiceNow AI Platform UI

The Automated Test Framework prevents custom UI testing of ServiceNow AI Platform features. For example, you cannot test dashboards or graphical designers. Instead, build tests to validate your custom UI pages and elements since you have direct control over these user interfaces.

Use HTML attributes to override page component testing properties

Change the testing properties of a particular page component using HTML attributes that are specific to Automated Test Framework. See Override component test actions.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

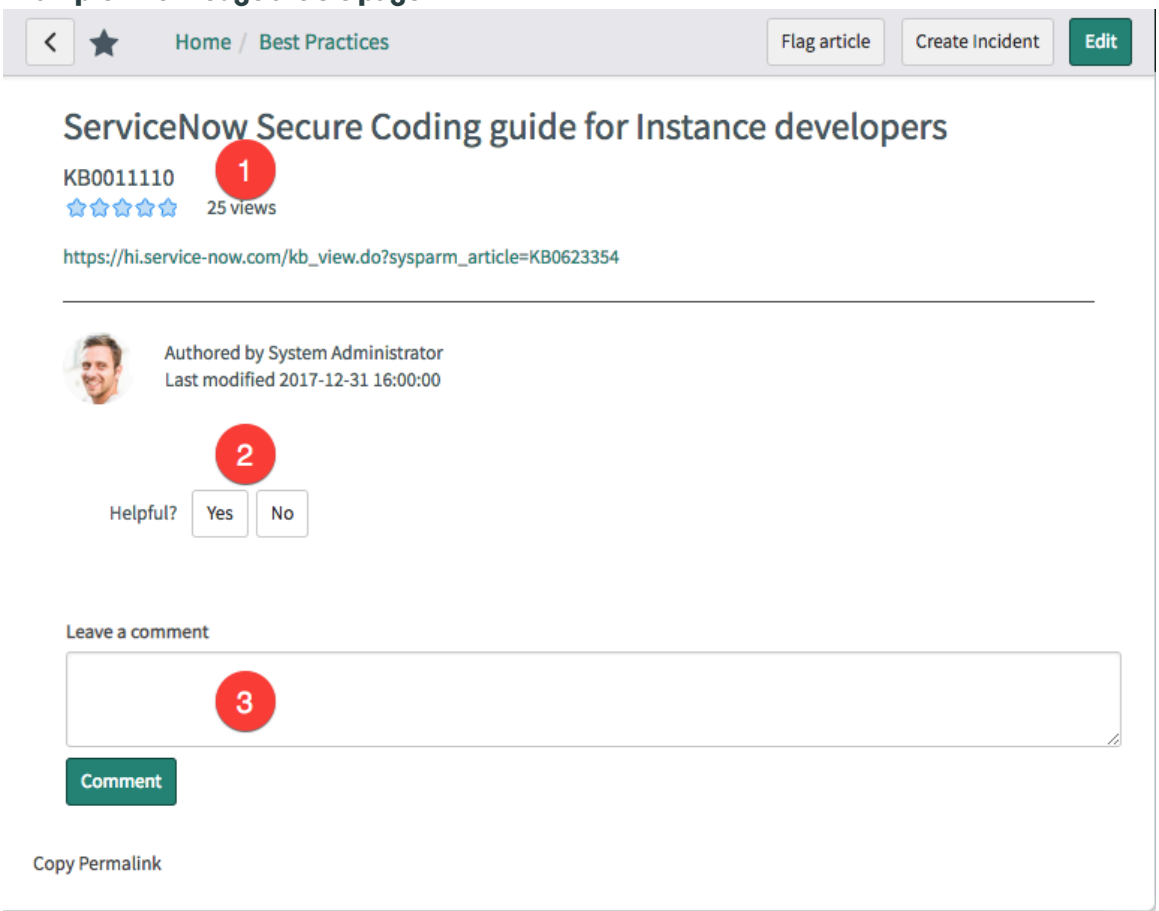
Retrieve page components again when you move tests to another instance

Custom UI test steps do not store UI components as metadata. Testers must manually retrieve page components again when moving tests between instances.

Example custom UI testing

You can use the list of retrieved components to design custom UI test steps. For example, suppose that you want to test reviewing and commenting on a Knowledge Base article. A Knowledge Base article contains several page components that require custom UI steps to test.

Example Knowledge article page



For example, these page components require custom UI test steps.

1. The number of article views.
2. The buttons to mark the article as **Helpful**.
3. The text area to **Leave a comment**.

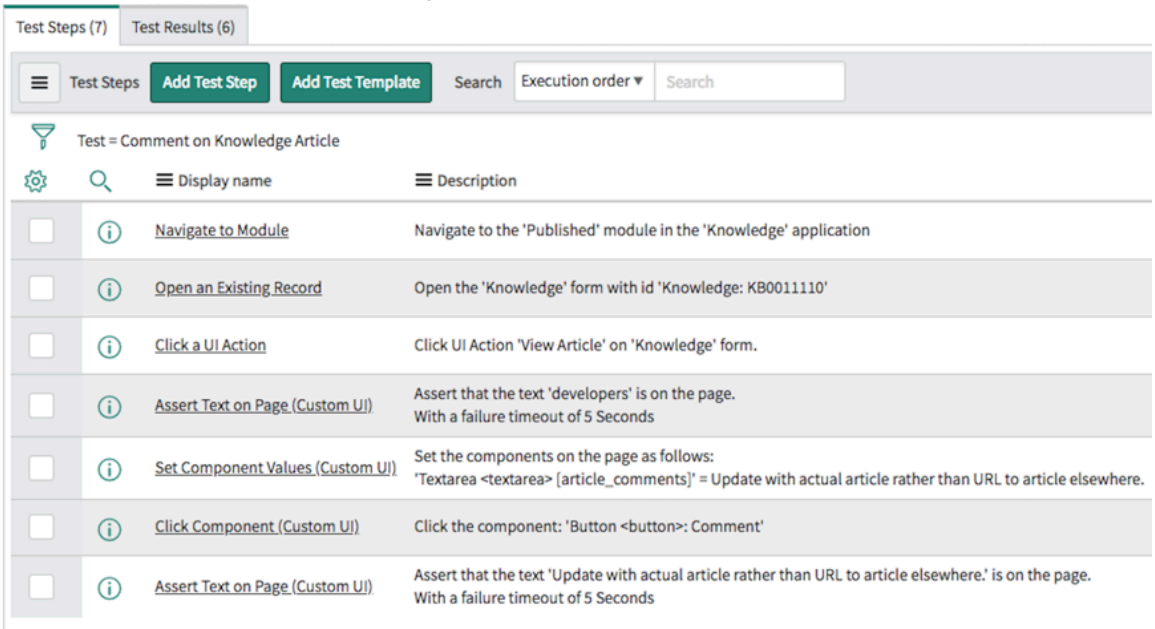
These steps demonstrate custom UI testing on a Knowledge Base article. The example test consists of these existing and custom UI test steps.

1. **Navigate to Module**. Navigate to the 'Published' module in the 'Knowledge' application.
2. **Open an Existing Record**. Open the 'Knowledge' form with id 'Knowledge: KB0011110'.
3. **Click a UI Action**. Click UI Action 'View Article' on 'Knowledge' form.
4. **Assert Text on Page (Custom UI)**. Assert that the text 'developers' is on the page.
5. **Set Component Values (Custom UI)**. Set the components on the page as follows: 'Textarea <textarea> [article_comments]' = Update with actual article rather than URL to article elsewhere.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

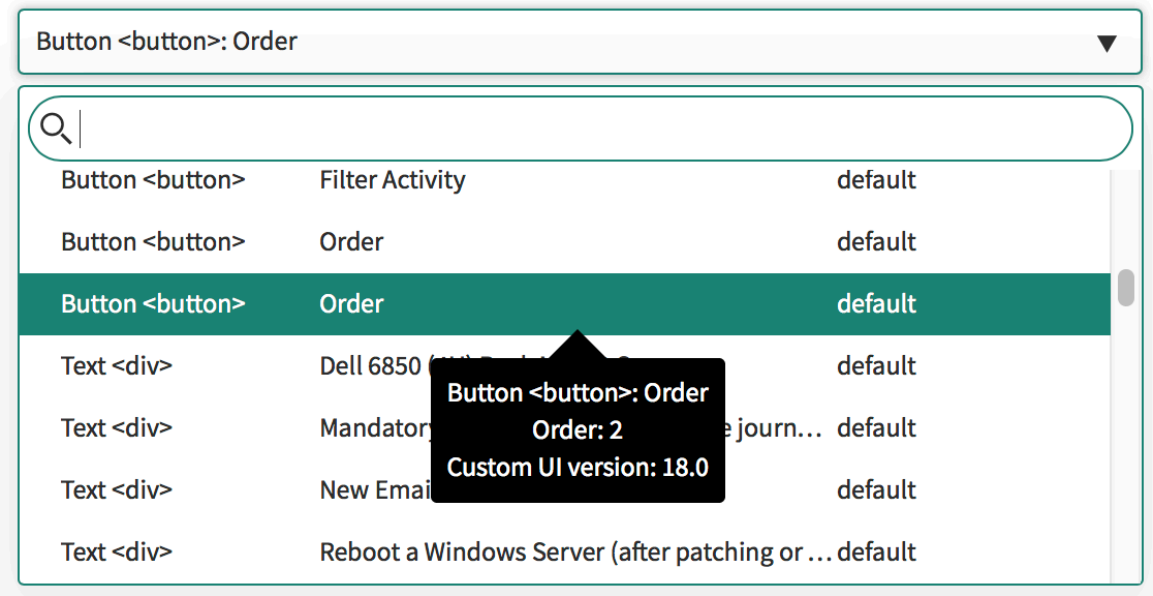
6. **Click Component (Custom UI)**. Click the component: 'Button <button>: Comment'.
7. **Assert Text on Page (Custom UI)**. Assert that the text 'Update with actual article rather than URL to article elsewhere.' is on the page.

Example test steps for a knowledge base article test



Custom UI component version and order

When you select a component in any of the custom UI test steps, the Custom UI version might show up. If there are multiple duplicate components, the order of the components show up.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note:

- The Custom UI version shows up only if there are at least two different components with different Custom UI versions.
- The Order shows up only if there are duplicate components in the component drop-down menu. You can disambiguate them according to the displayed order.
- Both Custom UI version and Order show up if there are multiple duplicate components from different versions.

Identifying components

Implement an alternative way to identify your component by using the *sn-atf-id* attribute. This is useful if the name or id attribute of your component is dynamic and changes every time a test runs. Add the sn-atf-id attribute with a consistent value to allow ATF identify your component when running a test. It also allows you to identify your component when building a test. For example, in a button component

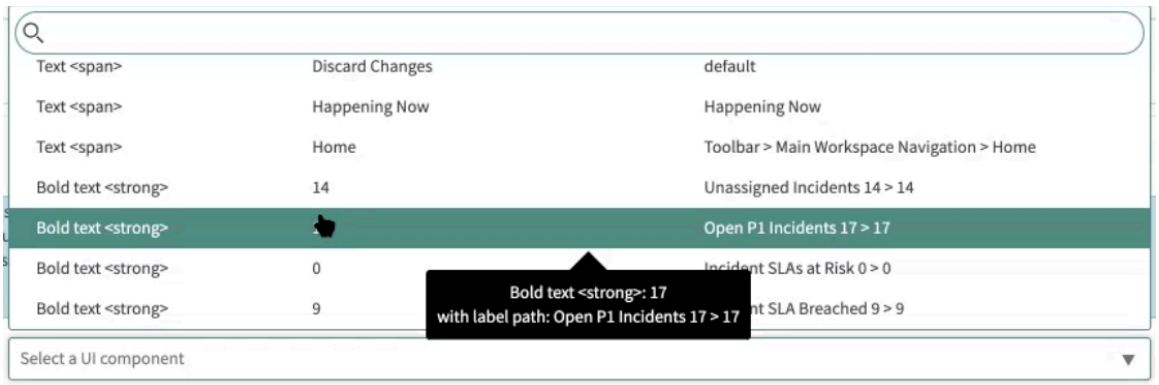
```
<button sn-atf-id="consistentValue">Test</button>
```

Note: Starting with the Rome release, if you have exactly one component on the page that has sn-atf-id attribute, ATF finds that component irrespective of any other attributes on that component.

Improve ATF component identification

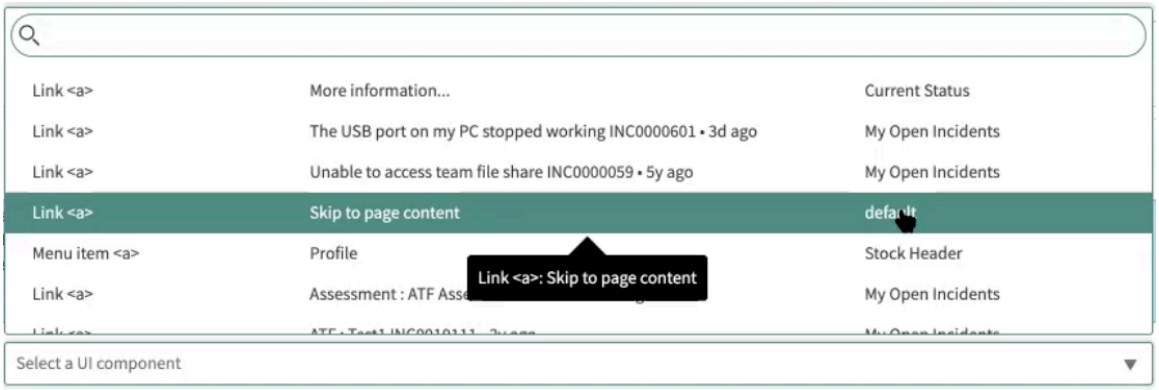
Identify your component using the label path included in the **Page area** column. Starting with the Rome release, the `sn_atf.element.use_label_path` property has been set to true by default.

If a component has the sn-atf-area attribute *sn-atf-area* attribute, the **Page area** column displays the sn-atf-area value *sn-atf-area* value. If the sn-atf-area attribute *sn-atf-area* attribute is not present, the label path for that component is shown in the Page area column.



Note: If you have multiple, identically named **Label** values, the label path helps you identify the correct component. If a component doesn't have an `sn-atf-area` attribute or a label path, the value is displayed as default.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



When you select the required component from the list and click **Submit**, the description of the test step also gets updated with more details.



Page Inspector

Identify the HTML and JavaScript page components in your user interfaces that are available for custom UI testing. Enable automated testing by ensuring that your user interfaces only contain testable page components.

The Page Inspector is a developer setting that opens a new pane beside any currently displayed ServiceNow AI Platform page.

The Page Inspector provides these features.

1. A highlighted frame to identify the currently inspected page.
2. A banner icon to display or hide the Inspector pane.
3. A button to refresh the list of page components.
4. A button to hide the Inspector pane.
5. An inspector button to identify specific components on a page.
6. A list of page components available for custom UI testing.
7. A page component search filter.
8. A row to click to see more information about a page component.
9. A highlighted row and tooltip to preview information about the currently inspected component.
10. A highlighted page component to identify the currently inspected component on the page.
11. A component label and back button to return to the list of page components.
12. A list of actions to take on the current page component.
13. A list of attribute information for the current page component.
14. An attribute name and value pair for a page component.

> **Note:** You can use the Label Path field to show the hierarchical page context around an element while inspecting a component on a page. If there are many "ancestor" label paths of an element, it can display only three. Each label path element is separated by a >. It can be used to differentiate between two similar looking components.

Note: UI buttons with icons are now visible in the page inspector.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Page inspector launcher

The page inspector launcher helps you select and launch a specific page within the page inspector. See Inspect different page types for more information.

Testable page components

The Page Inspector retrieves the list of testable page components when you first load a page. Testable components consist of standard HTML and JavaScript that are accessible to the Automated Test Framework. Test designers can use these components as part of custom UI testing.

Untestable page components

The Page Inspector identifies these types of untestable page components.

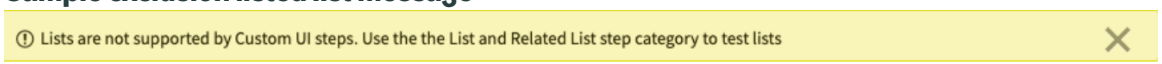
Exclusion listed

Exclusion listed page components can't be tested using any **Custom UI** test steps. Exclusion listed page components typically include specialized ServiceNow AI Platform user interfaces and components already testable by other means. Test designers can't create custom UI tests for exclusion listed page components. To test these components, they must use another supported test category such as Forms or Service Catalog.

In the Page Inspector, the exclusion listed interfaces are indicated by a grey background color when the inspect icon is dragged over them. A detailed message about the step category that needs to be used to test these components is displayed by dropping the inspect icon on an exclusion listed component.

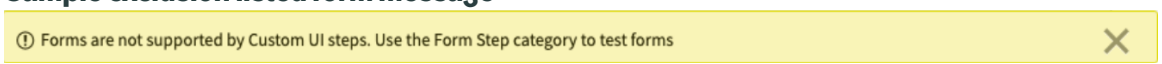
- Lists can't be tested using custom UI test steps. Use List and Related List test step category to test lists.

  Sample exclusion listed list message

- Forms can't be tested using custom UI test steps. To test forms, use Form category test steps.

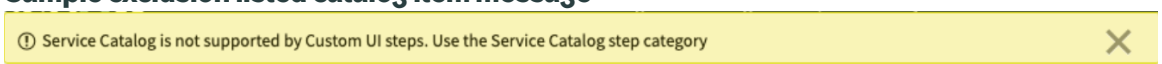
  Sample exclusion listed form message

  Note: The UI formatters within forms can be accessed and tested using custom UI test steps. See Create a custom UI test for more information.

- Service Catalog items can't be tested using custom UI test steps. To test catalog items, use Service Catalog category test steps.

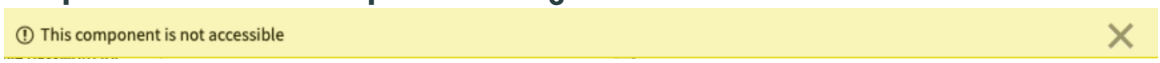
  Sample exclusion listed catalog item message

- Certain components are excluded from custom UI pages and are forbidden from being tested.

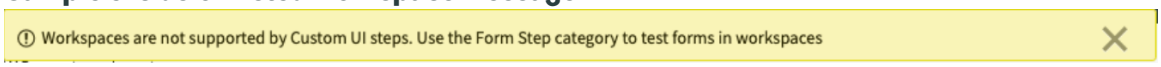
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Sample exclusion listed component message



- Workspaces can't be tested using custom UI test steps. To test forms in an available workspace, use Form category test steps.

Sample exclusion listed workspace message

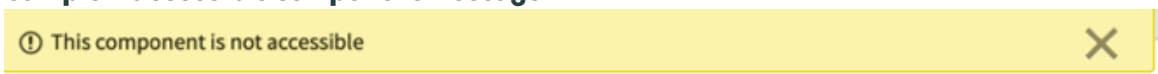


Inaccessible

Inaccessible page components are elements that Automated Test Framework either cannot identify or does not have permission to test because of some configuration issue. Inaccessible page components typically include third-party JavaScript libraries or elements with a Shadow DOM. Test designers cannot create custom UI tests for inaccessible page components, but may be able to redesign the page to use components accessible to Automated Test Framework.

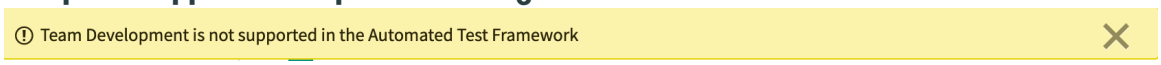
- Certain components are not accessible in the Automated Test Framework

Sample inaccessible component message



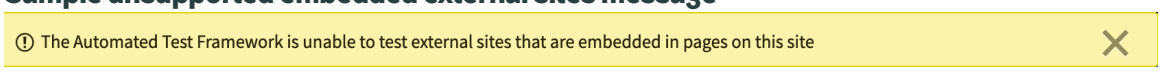
- Team development is not supported in the Automated Test Framework

Sample unsupported component message



- Testing external sites that are embedded in platform pages is not supported in the Automated Test Framework

Sample unsupported embedded external sites message



Inspect different page types

Inspect and troubleshoot the functionality of different page types like **UI Pages**, **Service Portal**, **Standard UI**, and **Custom URL** using the Page Inspector.

Before you begin

Role required: admin, atf_test_designer, atf_test_admin

Procedure

1. Navigate to **All** > **Page Inspector** > **Manual Page Inspector**.
2. Select the **Page Type** you want to inspect.

Inputs

Field	Description
UI Pages	Existing UI pages. Select a starting page from the available list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Standard UI	Standard platform forms, lists, and some UI pages. Select a form or a list or a UI page in the **Starting Page** field. **Note:** The optional **Record** field shows up only if you select a form in the **Starting Page** field.
Service Portal	Any available portal in the instance. Select an available portal and a starting page to be inspected. **Note:** Portal field shows up only if you select Service Portal as the **Page Type**.
Custom	Custom Platform URL to be inspected. **Note:** Don't copy-paste the complete URL onto the **Starting Page URL** field. For example, include only `/home.do` as the input if you want to inspect `https://instance.service-now.com/home.do`

3. Click **Inspect**.

Enable and use the page inspector

Enable a developer setting to inspect UI pages that open within the platform. Use the Manual Page Inspector to inspect pages that open in a new tab, such as Service Portal pages.

Before you begin

Role required: admin

About this task

Identify the HTML and JavaScript page components in your user interfaces that are available for custom UI testing. Enable automated testing by ensuring that your user interfaces only contain testable page components.

Procedure

1. Enable the Page Inspector depending on the type of custom UI page you need to inspect. Note that some custom UI pages open in a new tab and can be inspected only manually.

Option	Procedure
Inspect page opening in the ServiceNow AI Platform.	**a.** Navigate to the record or URL for the page you would like to inspect. **b.** Open System Settings and select the Devel oper tab. **c.** Enable **Automated Test Framework Page In spector**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Procedure
	The page opens in the Page Inspector and the inspect icon appears in the banner frame.
Inspect page opening in a new tab, such as a Service Portal page.	**a.** Navigate to **Automated Test Framework** > **Manual Page Inspector**. **b.** Set the **Page Type** and select the page. If the page type is Service Portal, select the portal in which the page opens. **c.** Click **Inspect**. The page opens in the page inspector.

2. Select a component to inspect.
   - Drag the inspect icon ( ) from the Page Inspector pane to a component on the page.
   - Select an available component from the Page Inspector pane.

   The Page Inspector retrieves the list of testable page components when you first load a page. Testable components consist of standard HTML and JavaScript that are accessible to the Automated Test Framework. Test designers can use these components as part of custom UI testing.
3. View component information and perform actions available in the **Action** field.
   Performing available actions helps you manually confirm which test steps are available to a component. For example, when you select **Click On Component**, the Page Inspector selects the component and displays the resulting page.

What to do next
Create a custom UI test

Create a custom UI test

Test components in custom UI pages.

Before you begin

- Use the Page Inspector to identify testable custom UI components. See Enable and use the page inspector.
- Role required: admin

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Tests**.
2. Click **New**.
3. Enter a name and description for your test.
4. Click **Save**.
5. In the Test Steps related list, click **Add Test Step**.
6. Add test steps to navigate to the target custom UI page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example
Use the **Navigate to Module** step to open a page that has an application navigator module.
Use the **Open Service Portal Page** step to open a portal page.
To open a UI page, use these test steps:

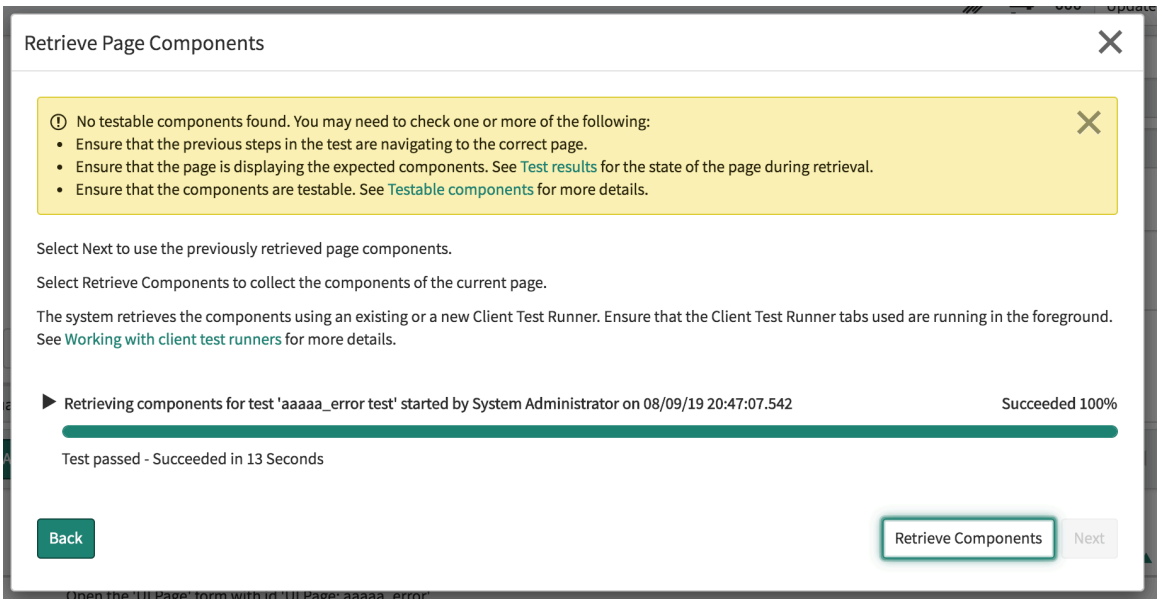
a. **Open an Existing Record**: Open the record for the page.

b. **Click UI Action**: Click the **Try It** UI action to open the page.

7. Add test steps from the Custom UI category to validate the behavior of custom UI components. For a list of available Custom UI test steps, see Custom UI category.
8. Retrieve UI components when adding a test step.

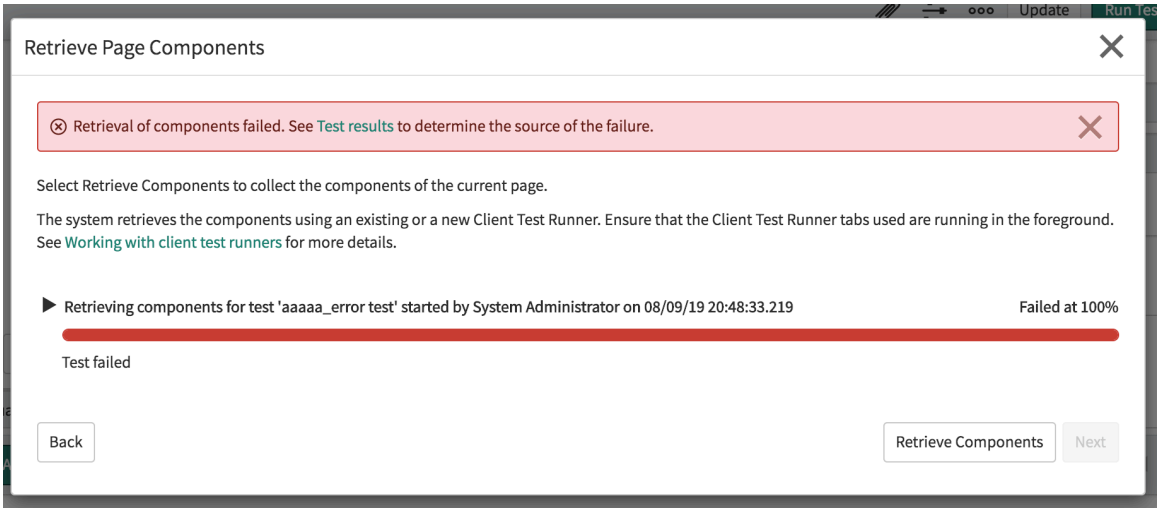
   Identify the testable page components on a custom UI page by retrieving a list of UI components for test steps. You can collect the components of the current page by selecting **Retrieve Components**.

   The system retrieves the components using an existing or a new client test runner. Ensure that the client test runner tabs used in the retrieval process are running in the foreground. For tests that don't yield any components after selecting **Retrieve Components**, the following warning message is displayed and the **Next** button is disabled.

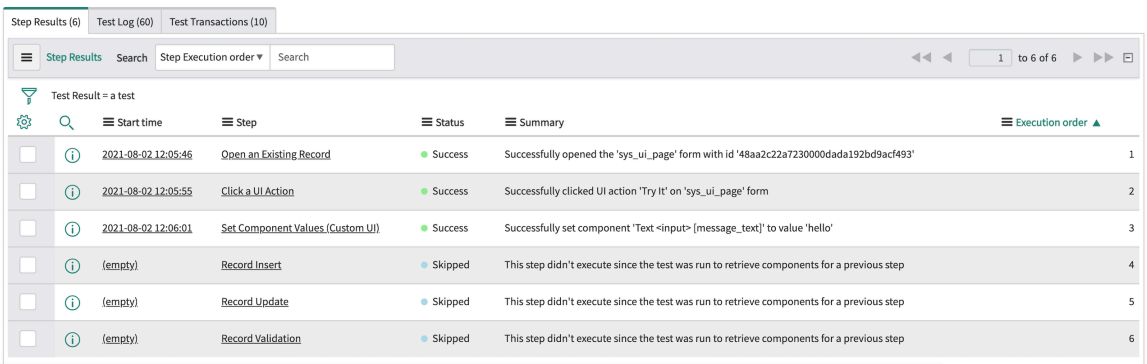


   If you have already retrieved a list of page components, click **Next** to use the previously retrieved list. You can also click **Retrieve Components** to rerun the current steps and refresh the list of testable page components. If the test fails while retrieving components, the following warning notification is displayed. You can review the test results by clicking the link in the notification.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Note: If you need to update a step in a test, click **Retrieve Components** for that step and the test runs only up to that particular step. The steps beyond the selected step don't run. The test passes if all the updated steps pass without failure.



Automated Test Framework stores the list of page components and displays the list in the **Component values** or **Component** field on the test step form. The field displays this information about each component:

Field	Description
Type	HTML element description and tag. For example, `Text<input>`.
Label	HTML component label.
Value	Value of the component. For example, if the component is a search input, the value might be **Search**.
Page area	HTML layout element that contains the component such as a `<div>` or `<section>` element.

If you create a parameterized test that includes Custom UI test steps, the system only uses the first data set to retrieve components.

What to do next

Consider adding tests to a test suite. For more information, see Building and running automated test suites.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Override component test actions

Change the testing properties of a particular page component using HTML attributes that are specific to Automated Test Framework.

Before you begin

Role required: admin

About this task

When Automated Test Framework retrieves a component, it determines which interactions it supports, such as whether it is a settable or clickable component. If the component is settable, Automated Test Framework determines the field type that can be set. If Automated Test Framework incorrectly determines your custom component's actions or field types, or your component contains multiple DOM elements that should be treated as one entity, explicitly set them using HTML attributes that are specific to Automated Test Framework.

Using sn-atf-clickable and sn-atf-settable attributes

Use `sn-atf-clickable` and `sn-atf-settable` attributes to specify that an element and its child elements should be treated as a custom clickable or custom settable component.

Before you begin

Role required: admin

Procedure

1. Open the custom UI page you would like to test.
2. Add the `sn-atf-clickable` or `sn-atf-settable` attribute to the element being tested.

   Example

   ```
   <div sn-atf-clickable="true" sn-atf-disabled
    id="customClickable">
    <button id="customButton">Click me</button>
   </div>
   ```

   ```
   <div sn-atf-settable="true" id="customSettable"
    sn-atf-component-value="A default value">
    <input id="customInput" value="A default value"></input>
   </div>
   ```

   Note: You can use either `sn-atf-clickable` or `sn-atf-settable` attribute to specify if an element should be treated as a custom clickable or custom settable component. You can't use both attributes on the same element.

3. Use either the `sn-atf-clickable` or `sn-atf-settable` attribute.

   - `sn-atf-clickable`: If you added the `sn-atf-clickable` attribute, ATF clicks the component by sending an `sn-atf-click` event to the DOM element with the `sn-atf-clickable` attribute. You should add an event listener (for example, using `addEventListener`) to this DOM element, and implement your custom click logic for the component. You can interact with a custom clickable component using Click Component test step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Clickable component attributes

Attribute name	Description
sn-atf-disabled	The presence of this optional attribute (regardless of its value) tells ATF that this component is disabled **Note:** If this attribute is missing, ATF considers that this component is enabled by default.
sn-atf-component-value	A string or JSON object that tells ATF the current value of this component

- `sn-atf-settable`: If you added the `sn-atf-settable` attribute, ATF sets the component value by sending an `sn-atf-setvalue` event to the DOM element with the `sn-atf-settable` attribute. You should add an event listener (for example, using `addEventListener`) to this DOM element, and implement your custom set value logic for the component. The value that needs to be set can be accessed with `event.detail.newvalue`. The `event` argument is passed to your event handler. You can interact with a custom settable component using Set Component Values test step.

Settable component attributes

Attribute name	Description
`sn-atf-disabled`	The presence of this optional attribute (regardless of its value) tells ATF that this component is disabled. **Note:** If this attribute is missing, ATF considers that this component is enabled by default.
`sn-atf-component-value`	A string or JSON object that tells ATF the current value of this component.
`sn-atf-data-type`	Optional type of field to present to user when building a step. It defaults to string. For example, `glide_date_time`, reference, boolean, etc.
`sn-atf-data-type-params`	JSON object with more data type details.

Example:

```
//A custom clickable component

<div sn-atf-clickable="true" sn-atf-disabled
 id="customClickable">
 <button id="customButton">Click me</button>
</div>
<script>
 var customClickableDiv =
 document.getElementById("customClickable");
 customClickableDiv.addEventListener('sn-atf-click', function()
 {
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
  document.getElementById('customButton').click();
 });
</script>
```

```
//A custom settable component

<div sn-atf-settable="true" id="customSettable"
 sn-atf-component-value="A default value">
 <input id="customInput" value="A default value"></input>
</div>
<script>
 var customSettableDiv =
 document.getElementById("customSettable");
 customSettableDiv.addEventListener('sn-atf-setvalue',
 function(event) {
  var newValue = event.detail.newValue;
  document.getElementById("customInput").value = newValue;
 });
</script>
```

Using sn-atf-class attribute

Use the `sn-atf-class` attribute to specify the JavaScript object to use when testing a custom clickable or settable component. Write a custom JavaScript object to specify the test actions available for a custom component.

Before you begin

Role required: admin

About this task

Test designers can manually specify the test actions available for a custom component by writing a custom JavaScript object and assigning the component a `sn-atf-class` attribute. Set the value of the attribute to the name of the JavaScript object containing the component test actions. Testable custom components must be either clickable or settable, and this classification determines the functions and properties your JavaScript object requires. See Custom UI test steps for testable page component requirements.

Procedure

1. Open the custom UI page you would like to test.
2. Add the `sn-atf-class` attribute to the element being tested and set the value to the name of the JavaScript object embedded in the page that handles getValue(), setValue(), click(), or isDisabled() functions.

   Example

   ```
   <div sn-atf-class="MyClickableComponent">
       <label
    for="a_clickable_checkbox">MyClickableComponent</label>
       <input type="checkbox" id="a_clickable_checkbox"
    checked="true"/>
   </div>
   ```

3. Create the JavaScript object specified in the `sn-atf-class` attribute and add the functions and attribute needed to identify your custom page component as either a clickable or settable page component.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Clickable component functions

Function name	Description
`initialize()`	Gets the initial values of the component. Enter: `$super(element, area)`
`click()`	Selects the component. Returns a JSON object with these properties: `success`: true if the component can be clicked. **Note:** Triggers the UI test step intelligent wait mechanism.
`getValue()`	Gets the value of the element. Returns a JSON object with these properties: ○ `success`: true if the value is retrieved. ○ `result`: the value of the component.
`isDisabled()`	Whether the component is disabled. Returns a JSON object with these properties: ○ `success`: true if the component is disabled. ○ `result`: true if the component is disabled.

Settable component functions

Function name	Description
`initialize()`	Gets the initial values of the component. Enter: `$super(element, area)`
`setValue(newValue)`	Sets the value of the component. See the second example below. Returns a JSON object with these properties: `success`: true if the value is set. **Note:** Triggers the UI test step intelligent wait mechanism.
`getValue()`	Gets the value of the element. Returns a JSON object with these properties: ○ `success`: true if the value is retrieved. ○ `result`: the value of the component.
`isDisabled()`	Whether the component is disabled. Returns a JSON object with these properties: ○ `success`: true if the component is disabled. ○ `result`: true if the component is disabled.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Settable component attribute

Attribute name	Description
`isSettable: true`	Identifies the component as a settable page component.

4. When creating your custom component in Jelly, add `<g2:atf_only>` tags around the JavaScript object specified in the `sn-atf-class` attribute.
   These tags ensure the system only runs the JavaScript object during automated testing.

Example:

```
//A custom clickable component

<form>
     <div sn-atf-class="MyClickableComponent">
         <label
 for="a_clickable_checkbox">MyClickableComponent</label>
         <input type="checkbox" id="a_clickable_checkbox"
 checked="true"/>
     </div>
</form>
<script>
var MyClickableComponent = {

    // The constructor must have this signature, but you can
 perform additional setup after the $super(element, area) call
    initialize: function($super, element, area) {
        $super(element, area);
    },

    click: function() {
        document.getElementById('a_clickable_checkbox').click();
        return {success: true};
    },

    // The function returns an object with a result attribute of
 type String
    getValue: function() {
        var isChecked =
 document.getElementById('a_clickable_checkbox').checked ?
 "true" : "false";
        return {success: true, result: isChecked};
    },

    // The function returns an object with a result attribute of
 type Boolean
    isDisabled: function() {
        if
 (document.getElementById('a_clickable_checkbox').disabled)
            return {success: true, result: true};

        return {success: true, result: false};
    },
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
};
</script>
```

```
//A custom settable component

<form>
    <div sn-atf-class="MySettableComponent">
        <label
 for="a_settable_checkbox">MySettableComponent</label>
        <input type="checkbox" id="a_settable_checkbox"
 checked="true"/>
    </div>
</form>
<script>
var MySettableComponent = {

    // This attribute is required for settable components
    isSettable: true,

    // The constructor must have this signature, but you can
 perform additional setup after the $super(element, area) call
    initialize: function($super, element, area) {
        $super(element, area);
    },

    // The value parameter is a string
    setValue: function(value) {
        document.getElementById('a_settable_checkbox').checked =
 (value == "true");
        return {success: true};
    },

    // The function returns an object with a result attribute of
 type String
    getValue: function() {
        var isChecked =
 document.getElementById('a_settable_checkbox').checked ?
 "true" : "false";
        return {success: true, result: isChecked};
    },

    // The function returns an object with a result attribute of
 type Boolean
    isDisabled: function() {
        if
 (document.getElementById('a_settable_checkbox').disabled)
            return {success: true, result: true};

        return {success: true, result: false};
    },

};
</script>
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Reference and record picker

Use custom UI steps to manipulate the values of the `sn-reference-picker` and `sn-record-picker` angular directives. The value on a reference picker returns the `sys_id` of the chosen record. The value on a record picker returns the value field chosen for that record picker. Both elements can be set by selecting a record to set as their value.

Override component data type

Use the `sn-atf-data-type` and `sn-atf-data-type-params` attributes to override the type of field displayed in a Set Component Value test step.

Before you begin

Role required: admin

About this task

Settable components have a data type that determines what values a Custom UI test step can set. For example, a page component intended to display a reference to a particular record can have a reference data type to only display Sys ID values.

Procedure

1. Open the custom UI page you would like to test.
2. Add the `sn-atf-data-type` attribute to the settable component and set the value to the field type you would like displayed in the Set Component Value test step.

   This attribute contains a string of the testable data type. The available values include those listed in the following table.

Attribute value	Description
glide_date	Contains a string specifying a particular day.
glide_date_time	Contains a string specifying a particular day and time of day.
reference	Contains a Sys ID to a related record. This data type requires specifying additional information in the `sn-atf-data-type-params` attribute.

3. Add the `sn-atf-data-type-params` attribute to provide additional information when the value of `sn-atf-data-type` is `reference`.

   This attribute contains a string of JSON formatted key-value pairs. Available key-value pairs include those listed in the following table.

Key	Value
reference	Name of the table that contains the records you want the reference field to display. For example, add `"reference":"incident"` to display records from the Incident table.
reference_qual	Filter to apply to the query. For example, add `"reference_qual":"active=true"` to display only active Incident records. See Reference qualifiers

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example:

```
<input id="someTextField" value="someSysId"
 sn-atf-data-type="reference"
 sn-
atf-data-type-params='{"reference":"incident","reference_qual":"
active=true"}'/>
```

Select2 functionalities in ATF

Use the Select2 component to search and select your option from a drop-down menu easily.

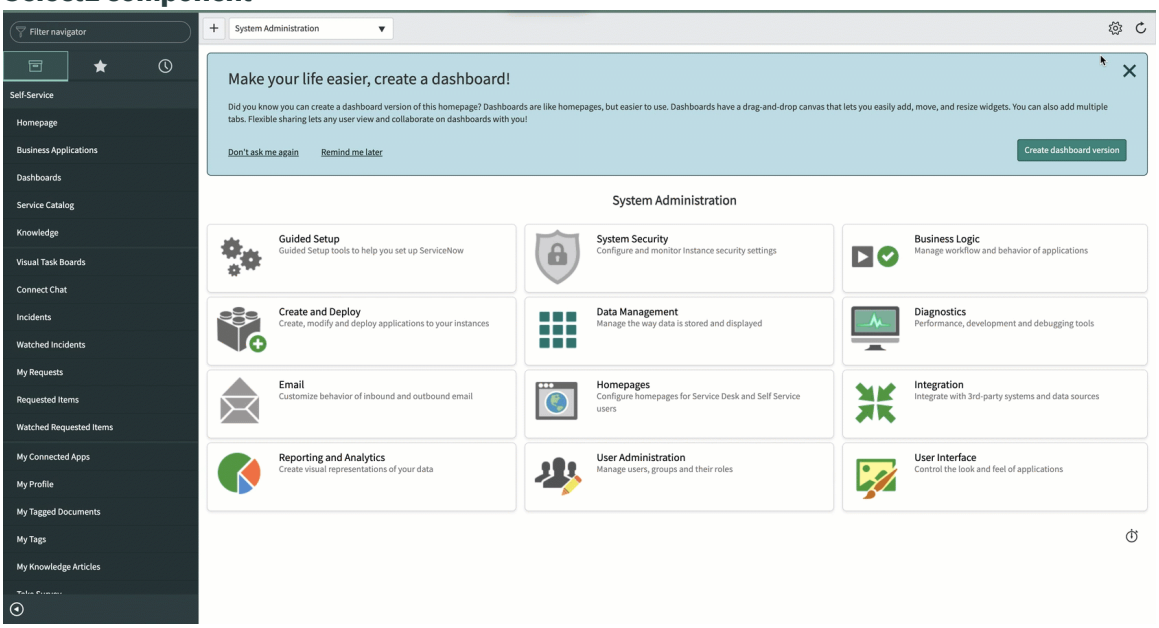
You can set the value of select2 in a test. The test then searches for the value in the search bar of Select2. The first valid result in the component drop-down gets selected.

Drag and drop the page inspector on the required drop-down menu. The right panel for the selected component opens. You can specify the text you want to search in the set component value step within the page inspector.

Select the **Set Component Value** option under **Action** to search for a component. Type in the text in the right panel and click **Submit**. This enables the code to open the select2 to search for that component and pick the first option that shows up.

Note: If multiple, similarly named component options show up on searching a term, only the first option is selected.

Select2 component



Limitations of the select2 support

- Only certain versions (3.5.1-3.5.4 and 4.0.0-4.0.13) of the select2 library are supported.
- The select2 support depends on the jquery library. So, if you try to set a value of select2 and the name of the jquery library is changed, it causes the test to fail.
- Searching through Select2 won't work when there is no search bar.

Design considerations

Search and select your options efficiently with the following design considerations:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- It is recommended to use uniquely searchable test data
- Wait for the current element to be set before proceeding to set the next element while setting the value of Select2 via the Set Component Values (Custom UI) test step
- Prevent failing of tests by avoiding jquery library name change
- Select2 Adapter and Decorator features are not supported

Browser recommendations for Automated Test Framework

Configure client test runner browsers to run automated tests and avoid performance degradations.

Periodic browser restarts

These browsers have memory-management limitations that make it necessary to occasionally close and restart the browser when running the client test runner.

- Internet Explorer
- Edge
- Older versions of Firefox

How often you should close the browser depends on the memory allocation in the browser application.

Browser CPU throttling

Some browsers throttle CPU usage for windows that are out of focus. Follow these guidelines to avoid CPU throttling issues.

- Run each client test runner in its own browser window.
- Ensure that the client test runner browser window is always partially visible on the screen.
- Ensure that the system screen is not locked or shut off.

Browser zoom level

Client test runners take screen shots as they run tests. For best results with screen shots, leave the browser zoom level set to 100%.

OS X CPU throttling

On OS X with the client test runner on Chrome or Safari: If the screen is locked or the client test runner tab is not shown, when the system attempts to run the test suite, tests run significantly slower and may time out. For best performance, run client test runners for scheduled suites in a virtual machine (VM) environment in which the screen does not become locked or disabled.

Rollback in browser sessions

The session cookies roll back all the changes made during a test. When a test is running, everything performed in that session is recorded for rollback. Don't modify your instance when a test is running in the same browser session. For example, if you modify records while a test is running in the same session, the changes are rolled back after the test completes. If you navigate around in other tabs in the same session, your work may be rolled back and interfere with tests that rely on implicit navigation.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Parallel testing

Follow these guidelines to avoid issues when running multiple tests in parallel.

Run each client test runner in an incognito or private window

Because parallel tests roll back all changes tied to the same browser session, it's possible for legitimate changes made in another browser tab to be rolled back during parallel testing. To prevent unwanted rollback of changes, always run client test runners in their own browser session. Opening client test runners in an incognito or private window ensures that they always have their own browser session.

Close client test runner windows when testing is complete

To prevent unwanted rollback of changes, always close client test runners after testing is complete. Closing the browser window ensures that test rollback doesn't revert any legitimate changes made in another browser tab.

Working with client test runners

If an automated test includes steps that involve a form or any other user-interface (UI) element, it runs those steps in a browser tab or window called a test runner or client test runner.

The Automated Test Framework supports two types of client test runners: Client Test Runners for manually started tests and Scheduled Client Test Runners for tests started by a schedule.

When test execution is enabled, clicking the Client Test Runner module opens the client test runner in the current browser session. If tests are waiting to be run, the Client Test Runner runs a waiting test. If no test is running, the message Waiting for a test to run displays in the client test runner.

While the client test runner is idle, it checks every five seconds for waiting tests to start. This ensures that the system runs any tests it may have been unable to start because no client with the proper configuration was available.

Note: The client test runner monitors for tests from the current session and runs those tests as the logged-in user (unless it executes an Impersonate User step). If you start a client test runner, log out from the current session, and then log in again, the client test runner executes using the new session.

When the client runner is active, it displays the activity of the currently running test in the **Execution Frame**.

Note: You are now allowed to execute multiple tests at a given time.

Test execution property

To work with the client test runner module, the test execution property must be enabled.

Note: By default, the system property that is used to run automated tests is disabled to prevent you from accidentally running these tests on a production system. To avoid data corruption or an outage, run tests only on development, test, and other non-production instances.

If the test execution property is disabled when you select this module, the system displays a message and a link to the automated test framework properties page where you can enable it.

Additional debugging functionality

If you have enabled additional debugging functionality, the client test runner module displays two tabs: **Execution Frame** and **Debug Info**. The **Execution Frame** displays the information

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

normally shown by the client test runner and the **Debug Info** displays additional debugging information.

The system takes screen shots from the tests in the **Execution Frame** tab and records them to the test result record.

Browser recommendations for all tests and suites

- Some browsers have memory-management limitations that make it necessary to occasionally close and restart the browser when running the client test runner. These browsers include Internet Explorer, Edge, and older versions of Firefox. How often you should close the browser depends on the memory allocation in the browser application.
- Some browsers have features that throttle CPU time. To avoid problems, follow these guidelines:
  - Run the client test runner in its own browser window.
  - Keep the client test runner at least partially visible on the screen.
  - Make certain the screen is not locked or shut off.
- The client test runner takes screen shots as the tests run. For best results with screen shots, leave the browser zoom level set to 100%.

Browser recommendations for scheduled suites

The client test runners for scheduled suites have additional browser requirements.

- On OS X with the client test runner on Chrome or Safari: If the screen is locked or the client test runner tab is not shown, when the system attempts to run the test suite, tests run significantly slower and may time out. For best performance, run client test runners for scheduled suites in a virtual machine (VM) environment in which the screen does not become locked or disabled.
- The browser must meet the criteria you specified on the Scheduled suite run record.
- A client test runner meeting the criteria you specified on the Scheduled suite run record must be available to run the test suite at the scheduled time. The system cannot automatically open a client test-runner session.

Javascript window command intercepts

The Client Test Runner captures window object commands including console.log, console.error, alert, confirm, and prompt, with default responses where necessary.

- Any script that calls **window.confirm** receives a boolean response of **true**.
- Any script that calls **window.prompt** receives the string response **test value**.
- Any script call to **alert** is ignored.

Active Test Runners table

When you start a client test runner, the system registers that runner in the Active Test Runners table. You can view this table in the Active Manual Test Runners module and the Active Scheduled Test Runners module. These two modules provide views of the same table, filtered to show only manual or only scheduled test runners.

The Active Scheduled Test Runner module is useful when you create a scheduled suite run. For scheduled suite runs, you can specify the browser to use. To determine the name and version

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

of a browser you want to use, start a scheduled test runner with that browser, then inspect that runner's record in the Active Scheduled Test Runners module.

The data in this table is transient. While the runner is active, it reports in to the system at a specified interval. If the runner does not report in at the expected time, the system marks the runner as inactive. After a period of time the system deletes the runner. You can modify these intervals on the Automated Test Framework properties page.

Related topics

- Client test runner
- Scheduled client test runner
- Allowed client errors

Pick a browser

If the test or test suite you are running contains steps that work with a form (any step involving a UI), or any other UI test step element (such as Automated Service Catalog test steps), work with the Pick a Browser dialog.

Before you begin

You must have created the test you want to run.

The test execution property must be enabled. You must have an admin or atf_test_admin role to do so.

Note: The test execution property is disabled by default to prevent running tests on a production system. Run tests only on development, test, and other sub-production instances.

Role required: atf_test_admin, atf_test_designer, or admin

About this task

The Pick a Browser dialog appears after you click **Run Test** or **Run Test Suite**. The dialog asks you to choose among any currently-running test clients or start a new test runner.

Procedure

1. Choose the browser in which the test or test suite should run or be rerun again in the case of failed tests.
   All registered client test runners that are currently active appear in the Pick a Browser dialog. (Current session) indicates that the browser is currently running. See Working with client test runners.
2. Click **Manage your test runners here** as needed to view all client test runners registered for the current user.
   See Active manual test runners.
3. If you want to open a client test runner in this browser session, click **Start a new test runner**, which appears when client test runners are only available in other browsers for the current user.
4. Click the appropriate button to run the test.
   - If you are running a single test, click **Run Test**.
   - If you are running a test suite, click **Run Test Suite**.
   - If you are running failed tests again, click **Re-run failed tests**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Result
The test, test suite or failed tests run in the selected browser or client test runner. The Progress viewer appears for monitoring of the progress of the test run.

Cloud Runner browser

If you are running a test or a test suite, select the Cloud Runner browser option to run your tests in a cloud browser.

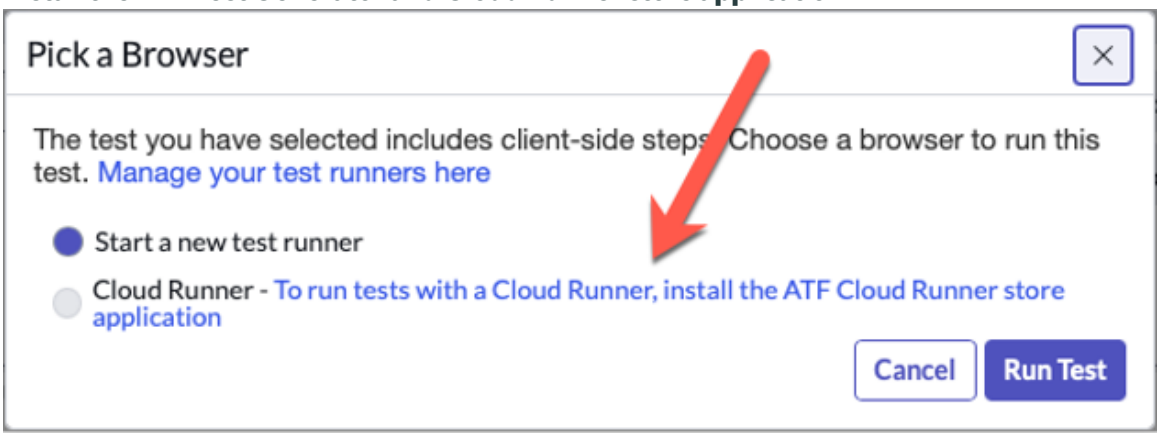
If you want to use the Cloud Runner browser option, you are required to complete the following:

- Install the ATF Test Generator and Cloud Runner store application.
- Configure a user to run and generate a test.

Use cases for Cloud Runner browser

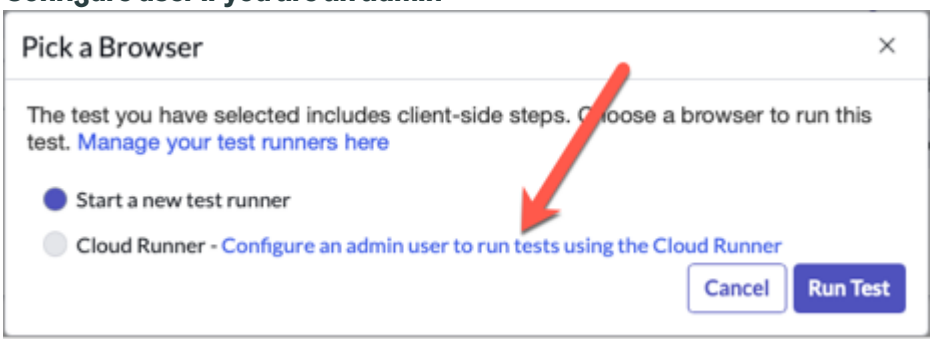
- When you haven't installed the ATF Test Generator and Cloud Runner store application

Install the ATF Test Generator and Cloud Runner store application



- When you haven't configured the user to run and generate tests
  - If you are an admin

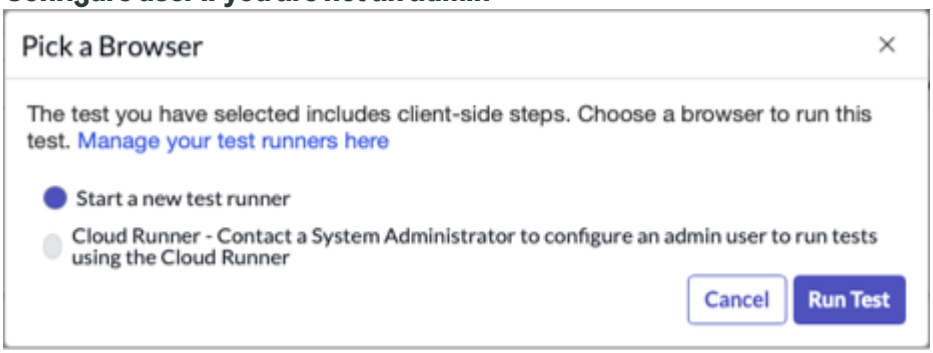
Configure user if you are an admin



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

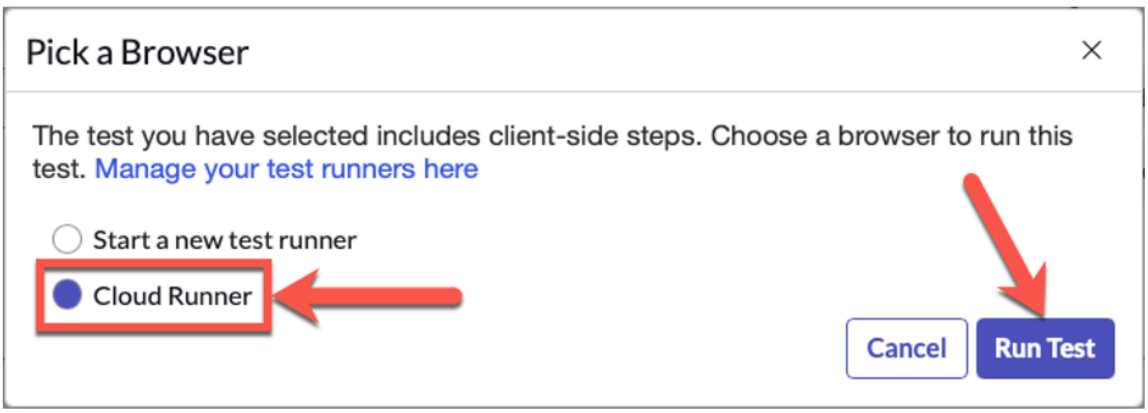
- If you are not an admin

Configure user if you are not an admin



- When the application is installed and the user is configured

Use the cloud runner option



Note: See ATF Test Generator and Cloud Runner for more information.

Server test steps

Test business logic and background processes by performing operations on the server.

Server test steps mimic non-interactive actions such as impersonating users, submitting or saving records, running server-side script, or making REST calls. Since server test steps run directly on the server, they do not require a client test runner. For more information about the server test steps, see Server category.

REST test steps

Test custom inbound web services and backwards compatibility by making REST calls.

REST requests can only be sent to the current instance. You cannot send a request to another instance or third-party at a remote address.

The REST test configuration only supports the XML and JSON response formats. Binary formats are not supported.

You can create tests that include steps from each of the test step configuration categories. The REST test configuration category contains the Send REST Request - Inbound and assert test configurations. Assert steps must immediately follow a **Send REST Request - Inbound** step. You can have multiple REST assert steps following a **Send REST Request - Inbound** step, but the

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

assert steps cannot be separated from the **Send REST Request - Inbound** step by steps from other test categories. For more information about the REST test steps, see REST category.

Authentication

As part of the Automated Test Framework, there are two situations when you send REST requests:

- When you use the REST API Explorer to create and test a request
- When you run a test that contains a **Send REST Request - Inbound** step

When you use the REST API Explorer to create and test a request, and the request requires authentication, the REST API Explorer uses your credentials. When the ATF runs the test, the REST API Explorer uses the credentials of the user who scheduled the test. This means that a test might fail unintentionally because of the difference in privileges between the user who created the test and the user who runs the test.

To address the issue of user credentials, you can create a basic-authentication profile for a test user and then on the **Send REST Request - Inbound** form, specify that the profile be used when the test is run.

Attachment test steps

Test an attachment-dependent business rule by uploading an attachment either from a form or from a server-side API call. For example, you can have a business rule that doesn't let you close an incident without an attachment such as a screenshot.

Upload from form

As a UI test step, the upload attachment step requires navigation to a form, which you can open using either **Open a New Form** or **Open an Existing Record**. Use **Upload Attachments** to select from the attachments that the test step adds to the form. When you select attachments to add to a form, the system waits to load the attachments before proceeding to the next test step. For more information on UI test dependency and wait mechanism, see UI test steps.

Upload from Server API

As a Server test step, the upload attachment step has no UI dependencies. Use **Upload Attachments** to select from the attachments that the test step adds to the record. When you select attachments to add to a form, the system waits for the attachments to be loaded before proceeding to the next test step. For more information, see Server test steps.

Design considerations

Follow these design considerations for attachment test steps:

- All attachment steps require adding one or more attachments.
- The system rolls back any attachments by the step after the test completes.
- The system cannot roll back any existing attachments after the test completes.
- Avoid testing records with existing attachments to eliminate data dependency.
- If UI testing is involved, add the attachment to a form.
- When no UI is involved, add the attachment to the Server API.

List UI actions test steps

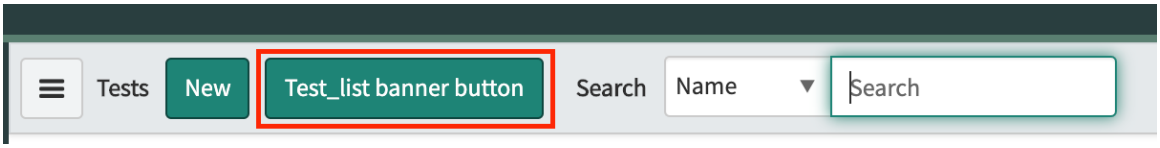
Select a UI action from a list to perform different actions on a list or a related list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

You can create a new UI action of the following types. See Create a UI action for more information.

- **List banner button**: Creates a button on the banner of a list.

List banner button



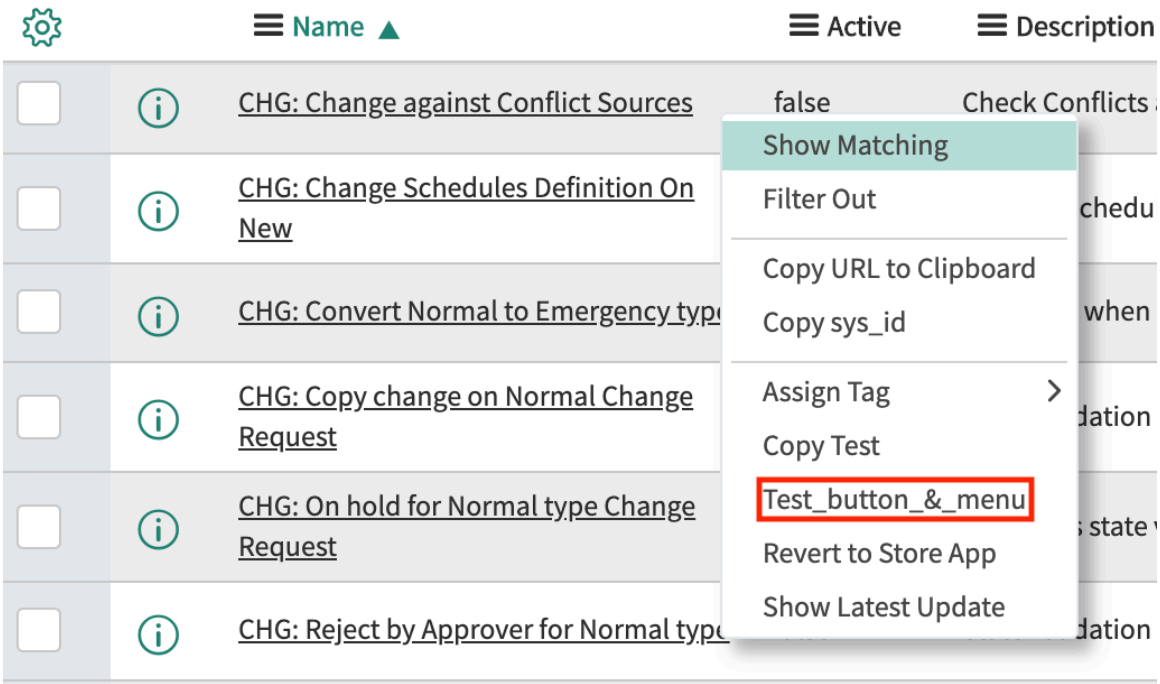
- **List bottom button**: Creates a button at the bottom of the list.

List bottom button



- **List context menu**: Adds an option to the context menu of the list.

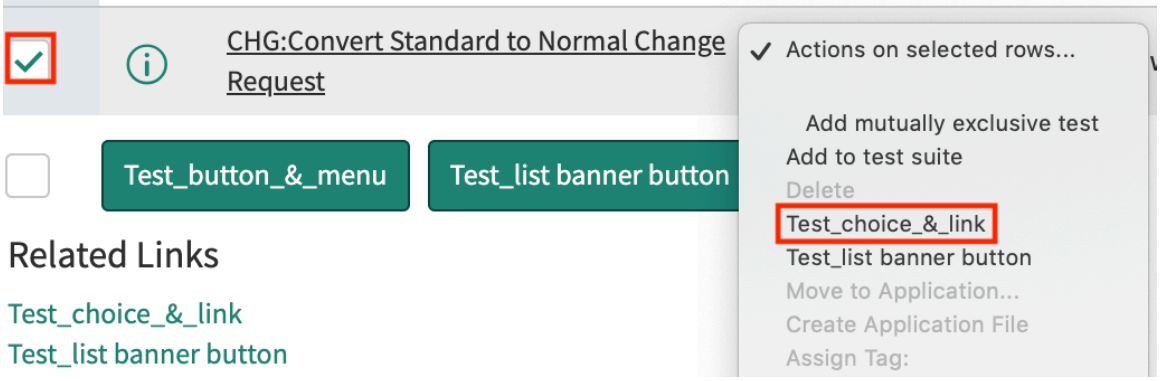
List context menu



- **List choice**: Adds an option to the list choice at the bottom of the list. You need to select one or more tests to enable the recently added list choice.

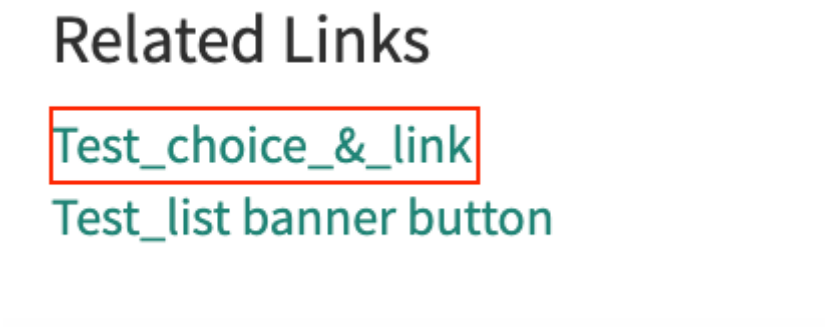
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

List choice



- **List link**: Adds a link to the **Related Links** list.

List link



Design considerations

- To use the **Click a List UI Action** test step, the test first needs to navigate to the list or the form with the related list containing the UI action.
- The **Related list** field appears only when you select Related list as the **List type**.
- **Action type** is mostly auto-filtered depending on the **List action** chosen.
- The **Record** field appears only when you select **Single record** for the UI action to be applied.
- Identify the specific record if you have selected **Single record** to apply the UI action.
- For the **Timeout** field to appear, select **Page reloaded or redirected** as the **Assert** type.

Parameterized tests

Run a test multiple times with different test data for each run. Create parameters to store test data for each test run.

Parameterized testing offers test designers these benefits.

- Eliminates the need to duplicate test steps just to change test data.
- Increases test reuse by separating test actions from test data.
- Produces a separate test result for each data set.

When the test runs, Automated Test Framework replaces the parameters with data set values. For example, you can create a test of the incident form that uses parameter values for the subcategory and priority fields. You can use one data set to test that the Antivirus category

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

produces a high priority incident, and another data set to test that the Email category produces a low priority incident.

Note: The Run Server Side Script test step is not supported in parameterized tests.

Parameterized test components

Parameterized tests consist of these components.

Parameter

A parameter is a variable that stores a particular type of test data. Each parameter has a unique label and a data type. For example, you can create a parameter to store the Sys ID of a reference field or the integer value of a choice field. Define parameters during test design.

Parameters can be shared or exclusive. Shared parameters can be used in any parameterized test. Exclusive parameters can only be used with the test for which they were created. Each shared parameter is a column in the Test Run Data Sets [sys_atf_parameter_set] table. Each exclusive parameter is a record in the Parameter Variables [sys_atf_parameter_variable] table.

Data set

A data set, also known as a test run data set, includes runtime data used when the test runs. You can set a value for every parameter available to the current test. Data sets specify the parameter value during test runs. You can manually create data sets for a test, or import data from a file. Each data set is a record in the Test Run Data Sets [sys_atf_parameter_set] table.

Parameterized tests fail if data sets are not defined.

Design Considerations

Follow these design considerations when creating parameterized tests.

- Parameterized tests support standard Automated Test Framework features, such as reports, test suites, and data rollback. Copying a parameterized test copies all parameters, test run data sets, and test steps.
- If you create a parameterized test that includes Custom UI test steps, the system only uses the first data set to retrieve components.

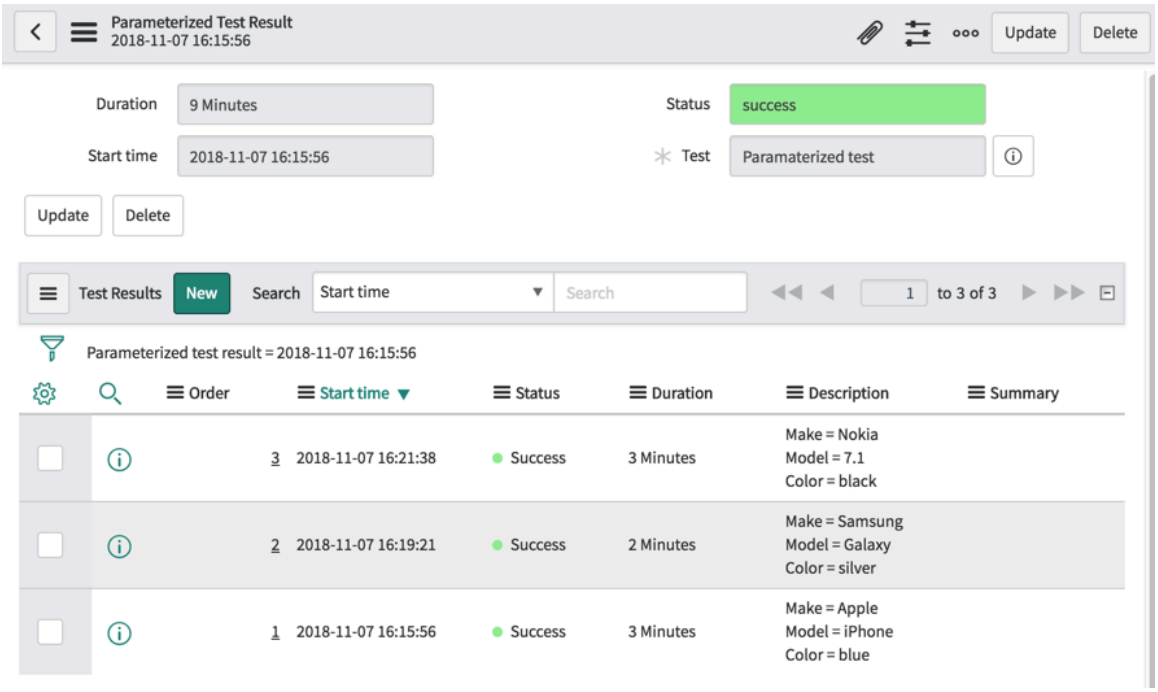
Parameterized test runs

Automated Test Framework runs each parameterized test once per data set, using the same test steps and execution order. For example, if a parameterized test has five data sets, Automated Test Framework runs the test five times, once for each data set.

Parameterized test results

Parameterized tests display test results by the execution order of the data sets. Open each test result record to view the test details.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



The Parameterized Test Result record **Description** field lists the parameters and data sets used in the test run.

Create a parameterized test

Build a test that uses variables to store test data.

Before you begin

Role required: atf_test_admin, atf_test_designer, or admin

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Tests**.
2. Click **New**.
   A blank Test record opens.
3. On the form, fill in the fields.

Field	Description
Name	Enter a name for the test.
Active	Enable
Enable parameterized testing	Enable
Description	Enter a description for the test.

4. Click **Save.**
   Parameterized testing related lists appear.
5. Create a parameter to hold test run data.
   A parameter is a variable that stores a particular type of test data. Each parameter has a unique label and a data type. For example, you can create a parameter to store the Sys ID of a reference field or the integer value of a choice field.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. In the Parameter Definitions related list, add a parameter.

- **Add Exclusive Parameters**: Adds a parameter available to this test only.
- **Add Shared Parameters**: Adds a parameter available to any parameterized test.

b. Define the name of the variable and the data type.
When creating parameters for a form, the parameter data type must match the field data type. For example, if you are creating a parameter to test a reference field on a form, you must create a parameter of type Reference. For more information on ServiceNow AI Platform data types, see Field types.

c. Click **Submit**.

What to do next
Add a parameter to a test step

Add a parameter to a test step

Add a variable to a test step to hold a particular type of data when the test runs.

Before you begin

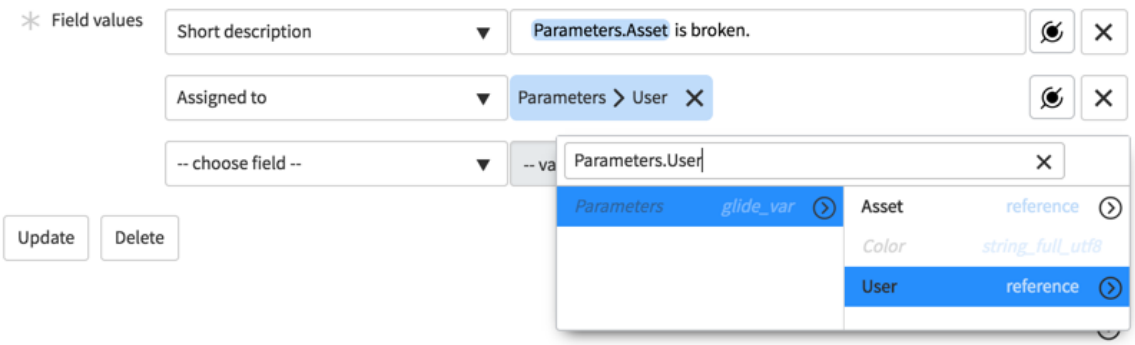
- Create a parameterized test
- Role required: atf_test_admin, atf_test_designer, or admin

About this task

Note: The Run Server Side Script test step is not supported in parameterized tests.

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Tests**.
2. Open a parameterized test.
3. In the Test Steps related list, create a test step or open an existing step.
4. Click and select a parameter to add it to a field.
   The parameter displays in the field.



Result
When the test runs, Automated Test Framework replaces the parameter with test run data.

What to do next
Add parameterized data sets.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add parameterized data sets

Add or import test data to specify parameter runtime values.

Before you begin

- Create a parameterized test
- Add a parameter to a test step
- Role required: atf_test_admin, atf_test_designer, or admin

About this task

A data set, also known as a test run data set, includes runtime data used when the test runs. You can set a value for every parameter available to the current test. Data sets specify the parameter value during test runs.

Test designers can add data sets manually, or import data from a file.

Note: If you create a parameterized test that includes Custom UI test steps, the system only uses the first data set to retrieve components.

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Tests**.
2. Open a parameterized test.
3. In the Test Run Data Sets related list, add or import data sets.

Option	Description
Add	The New Parameter Value Set form opens. This form includes all parameters available to the current test. **a.** Define the **Order** field to set the order in which data sets run. **b.** Specify the parameter value when the test runs. **c.** Click **Submit**.
Import	The Import Test Run Data Sets form opens. **a.** Define the import behavior. ▪ **Add**: Adds the data in the file to existing test run data sets in the Test Run Data Sets table. For more information, see [illegible]. ▪ **Replace**: Downloads a template that includes any existing data sets in the Test Run Data Sets table. Replaces all existing data sets that have matching Sys IDs in the Test Run Data Sets table with data from the uploaded file. For more information, see [illegible]. **b.** In the Upload Test Run Data File section, select **Yes** to download an Excel template with the required fields, then upload a completed file. Alternately, select **No** to upload a completed file. Each row in the file becomes an entry in the Test Run Data Sets table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

What to do next

Click **Run Test**. Automated Test Framework runs each parameterized test once per data set, using the same test steps and execution order. For example, if a parameterized test has five data sets, Automated Test Framework runs the test five times, once for each data set.

Parameterized tests display test results by the execution order of the data sets. Open each test result record to view the test details.

Allowed client errors

Add known client errors to the allowed client errors list to allow tests and steps to continue running when a specific error occurs. Set the report level to specify what the Automated Test Framework does when the error occurs in future tests.

Scenarios for allowing client errors

Test designers and developers typically allow client-side JavaScript errors to prevent certain types of known failures from impacting test design and results. Scenarios to allow client errors include:

Timing constraints

Temporarily allow a client error until your developers have time to investigate and resolve the issue. For example, when testing an old form containing a longstanding bug.

Minimizing the impact of old libraries

Ignore client errors that cannot be fixed or are unimportant to your operations to eliminate their impact on future test runs. For example, when you find a bug in an old library.

Test design time

Temporarily allow client errors until you finish writing tests and have time to investigate the error. For example, one of your developers modifies a UI policy and the change generates an error.

Possible platform bug

Temporarily allow client errors until a fix is available. Prior to reporting a platform bug to ServiceNow Technical Support, investigate the error, verify it is not a customization error, and identify the type of platform bug involved. For example, a UI policy generates an error during a test. Your investigation verifies that the issue is not a customization error and identifies a platform issue with the UI policy.

Note: While adding client errors to the allowed client errors list allows the test framework to continue testing, it does not guarantee that your tests pass. Test designers and developers should always investigate client errors to determine if there are issues with your business process. For more details, see Identify and resolve client errors.

Report levels for allowed client errors

The report level indicates whether the test framework reports future occurrences of the error as a warning or ignores them altogether. You can change the report level of an allowed error at any time. For example, if you originally add an error as a warning, you can later change the report level to ignored.

Report level	Description
Warning	Test steps containing the allowed client error report a status of `Success with warning(s)`. The error message appears in

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Report level	Description
	the test result output, and is recorded in the test logs with the status `Warning`.
Ignored	Test steps containing the allowed client error report a status of `Success`. The error is recorded in the test logs with an `Ignored` status.

Matching process

The Automated Test Framework identifies allowed client errors using a contains search rather than an exact string match. A match occurs when a client error contains a message from an Allowed Client Error [sys_atf_whitelist] record. For example, if you create an Allowed Client Error record for the error message "`Test message`" with a report level of Ignored, then any client error containing this string is ignored.

Note: When you create or modify an Allowed Client Error record, the Client Test Runner automatically gets the update.

Identifying and resolving client errors

When client errors occur, the Automated Test Framework fails the test on the step that was executing when the error occurred.

Allow client errors from test results

Allow client errors as you review test results.

Before you begin

Role required: atf_test_admin, atf_test_designer, or admin

About this task

You can allow multiple or individual client errors. For each client error, you must decide how to report future instances of the client error. Report level options include:

- Warning: Test steps containing the allowed client error report a status of `Success with warning(s)`. The error message appears in the test result output, and is recorded in the test logs with the status `Warning`.
- Ignored: Test steps containing the allowed client error report a status of `Success`. The error is recorded in the test logs with an `Ignored` status.

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Test Results**.
2. Select the test result for a specific test.
   The system displays the Test Result record.
3. From the Related links, select one of the following options:
   - **Add all client errors to warning list**: Allow all client errors in this test with a report level of Warning.
   - **Add all client errors to ignored list**: Allow all client errors in this test with a report level of Ignored.

   The Automated Test Framework allows the selected client errors and displays a status message at the top of the form.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Related topics

Allowed client errors

Allow client errors from step results

Allow client errors as you review step results.

Before you begin

Role required: atf_test_admin, atf_test_designer, or admin

About this task

You can allow multiple or individual client errors. For each client error, you must decide how to report future instances of the client error. Report level options include:

- Warning: Test steps containing the allowed client error report a status of `Success with warning(s)`. The error message appears in the test result output, and is recorded in the test logs with the status `Warning`.
- Ignored: Test steps containing the allowed client error report a status of `Success`. The error is recorded in the test logs with an `Ignored` status.

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Test Results**.
2. Select the test result for a specific test.
   The system displays the Test Result record.
3. Select the client errors to be allowed.

Option	Description
Specific step result from Step Results related list	**a.** In the Step Results related list, right-click the step result containing client errors you want to allow. **b.** From the context menu, select one of the following options: ▪ **Add all client errors to warning list**: Allow all client errors in this step with a report level of Warning. ▪ **Add all client errors to ignored list**: Allow all client errors in this step with a report level of Ignored.
Multiple step results from Step Results related list	**a.** In the Step Results related list, select the check box in the first column for each step result containing client errors you want to allow. **b.** From the **Actions on selected rows** list, select one of the following options. ▪ **Add all client errors to warning list**: Allow all client errors in this step with a report level of Warning. ▪ **Add all client errors to ignored list**: Allow all client errors in this step with a report level of Ignored.
Specific step result from Step Result record	**a.** In the Step Results related list, select the Step Result record containing a client error you want to allow. **b.** From the Related links, select one of the following options:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Description
	▪ **Add all client errors to warning list**: Allow all client errors in this step with a report level of Warning. ▪ **Add all client errors to ignored list**: Allow all client errors in this step with a report level of Ignored.

The Automated Test Framework allows the selected client errors and displays a status message at the top of the form.

Related topics

Allowed client errors

Allow client errors from the test logs

Allow client errors as you review test logs.

Before you begin

Role required: atf_test_admin, atf_test_designer, or admin

About this task

You can allow multiple or individual client errors. For each client error, you must decide how to report future instances of the client error. Report level options include:

- Warning: Test steps containing the allowed client error report a status of `Success with warning(s)`. The error message appears in the test result output, and is recorded in the test logs with the status `Warning`.
- Ignored: Test steps containing the allowed client error report a status of `Success`. The error is recorded in the test logs with an `Ignored` status.

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Test Results**.
2. Select the test result for a specific test.
   The system displays the Test Result record.
3. Select the client errors to be allowed.

Option	Description
Specific test log from the Test Log related list	**a.** In the Test Log related list, right-click the test log client error you want to allow. **b.** From the context menu, select one of the following options: ▪ **Add client error to ignored list**: Allow the client error with a report level of Ignored. ▪ **Add client error to warning list**: Allow the client error with a report level of Warning.
Multiple test logs from the Test Log related list	**a.** In the Test Log related list, select the check box in the first column for each test log client error you want to allow. **b.** From the **Actions on selected rows** list, select one of the following options:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Description
	▪ **Add client error to ignored list**: Allow the client error with a report level of Ignored. ▪ **Add client error to warning list**: Allow the client error with a report lev el of Warning.
Specific test log from Test Result Item record	**a.** In the Test Log related list, select the test log (Test Result Item record) containing a client error you want to allow. **b.** From the Related links, select one of the following options: ▪ **Add all client errors to warning list**: Allow all client errors in this step with a report level of Warning. ▪ **Add all client errors to ignored list**: Allow all client errors in this step with a report level of Ignored.

The Automated Test Framework allows the selected client errors and displays a status message at the top of the form.

Related topics

Allowed client errors

Manually allow client errors

Manually create allowed client error entries as needed in the Allowed Client Errors table.

Before you begin

Role required: atf_admin, atf_test_designer, or admin

About this task

You normally allow client errors directly from a test result, step result, or test log, but you can manually allow them.

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Run** > **Allowed Client Errors**.
2. Click **New**.
3. On the form, fill in the fields.

Allowed Client Error form

Field	Description
Report level	Report action to take when the client error is encountered. Options include: ○ **Warning – Step & Test will report Success with Warning(s)**: Test steps containing the allowed client error report a status of `Success with warning(s)`. The error message appears in the test result output, and is recorded in the test logs with the status `Warning`. ○ **Ignored – Step & Test will report Success**: Test steps containing the allowed client error report a status of `Success`. The error is recorded in the test logs with an `Ignored` status.
Active	Check box to enable or disable allowing a client error.
Error message	Message of the client error you want to allow.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Description	Description of the error you want to allow. If this client error was allowed from a test result, step result, or a test log, the test log description is copied into this field.

4. Click **Submit**.

Related topics

Allowed client errors

View the progress of automated tests

When an automated test is running, view its progress in the Run Test progress dialog.

Before you begin

The system must be running a test.
Role required: atf_test_admin or atf_test_designer

About this task

When you execute a test or test suite, the system automatically displays the Run Test progress dialog. If you close this dialog, you can re-display it from either the results list or the results page for the currently-running test or test suite.

Procedure

1. If necessary, navigate to the Test Results list or Test Results page for the currently-running test or test suite.
   - If the system is currently running a test suite, navigate to **Automated Test Framework** > **Suite Results**. If desired, you can click the row for the running test suite to view the Suite Results page for that test suite.
   - If the system is currently running a test that's not part of a test suite, navigate to **Automated Test Framework** > **Test Results**. If desired, you can click the row for the running test to view the Test Results page for that test.
2. Display the Run Test dialog.
   - If you are currently viewing the Test Results page or the Suite Results page, click **Show Progress** under Related Links.
   - If you are currently viewing the Test Results list or Suite Results list, right-click the row for the running test or test suite, then click **Show Progress**.

Passing data from one automated test step to another

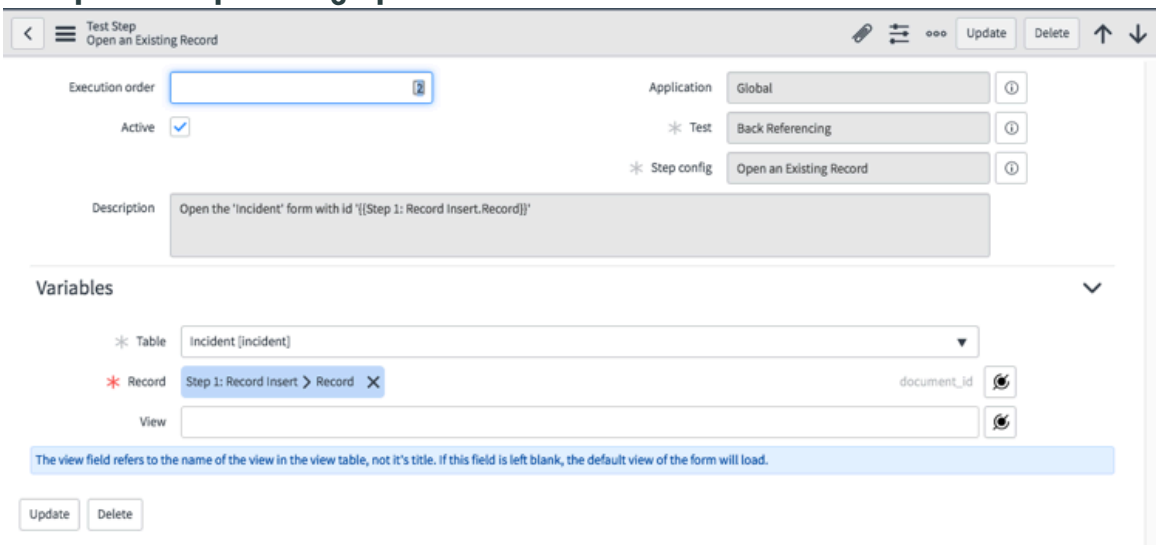
Some automated test steps create data that you can use as an input to a subsequent step.

You can pass data from one test step to another using input variables and output variables.

The term input variables is another name for the field values associated with a step. These values are input variables because they provide the input the step needs to accomplish its task. For example, the Open Form step has these input variables: **Table**, **Record**, and **View**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example test step showing input variables

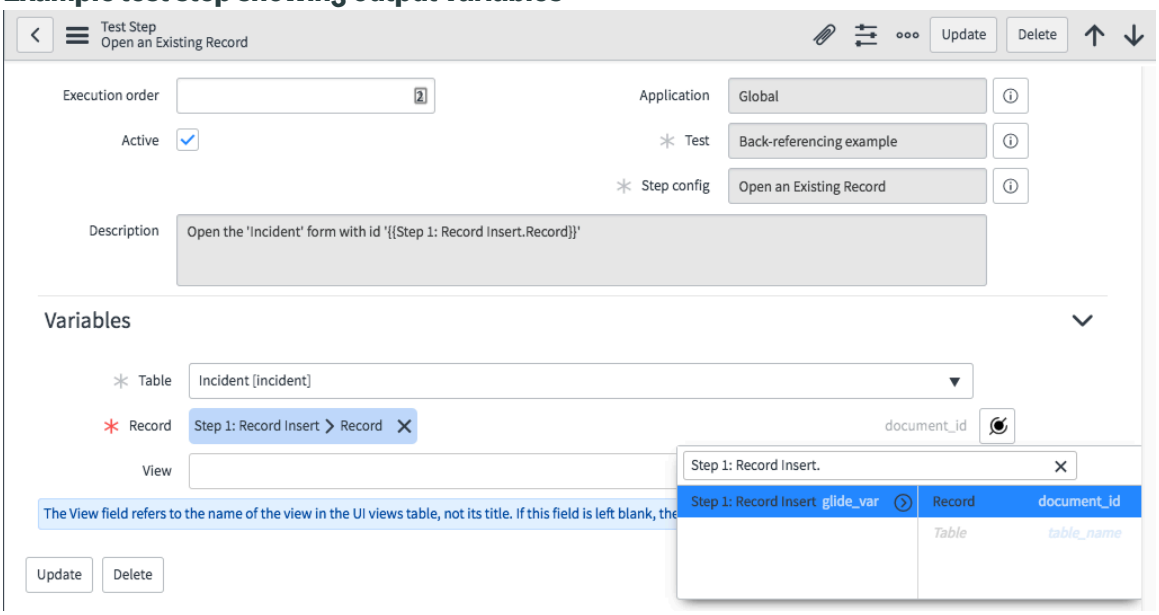


Some step types also have output variables. These are the values that later steps in the same test can use as input. For example, the Record Insert step has an output variable called Record which contains the sys_id of the newly created record.

Note: Input data can also be passed to an automated test case from an external source such as a .csv file.

The test step form doesn't indicate if a test step has output variables or not. You can easily tell if any output variables are available to provide a value to any given input variable. If you can map the value of one step's output variable to the current step's input variable, the system displays the mapping icon ( ) to the right of that input field. When you click the variable mapping icon, the system displays a tree giving you access to any available output variables from previous steps.

Example test step showing output variables



For step-by-step instructions on how to assign the value of an output variable to another step's input variable, see Pass values from one automated test step to another.

For an example of a test that passes variables using input and output variables, see Automated Test Framework use case: reference a value from a previous step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Pass values from one automated test step to another

Assign a form field the value of an output variable returned from a previous step.

Before you begin

You must have a previous test step that returns an appropriate output variable.

Role required: admin, atf_test_admin and atf_test_designer

Procedure

1. To the right of the field whose value you want to assign, click the input value icon ()
   The input value mapping control lists previous steps that create an output variable. If no previous steps create an output variable, the control displays the message: There are no elements to show .
2. Click the row for the step that contains the output variable you want to use as an input.
3. Click the output variable you want to use.
   If the output variable is an id for a glide record, the control displays a tree picker providing access to fields for this record.
4. Navigate through the tree picker hierarchy until you find and select the value you want.

Related topics

Automated Test Framework use case: reference a value from a previous step

Building and running automated test suites

Run a group of tests in a specific order to test an application or a group of related features.

A test suite can contain both individual tests and other test suites. A test suite that contains another test suite is called a parent, and the test suite contained within the parent is called a child. While a test suite can have both individual tests and test suites as children, tests cannot have other tests as children. Tests can only contain test steps.

Benefits

Grouping tests into test suites offers these benefits.

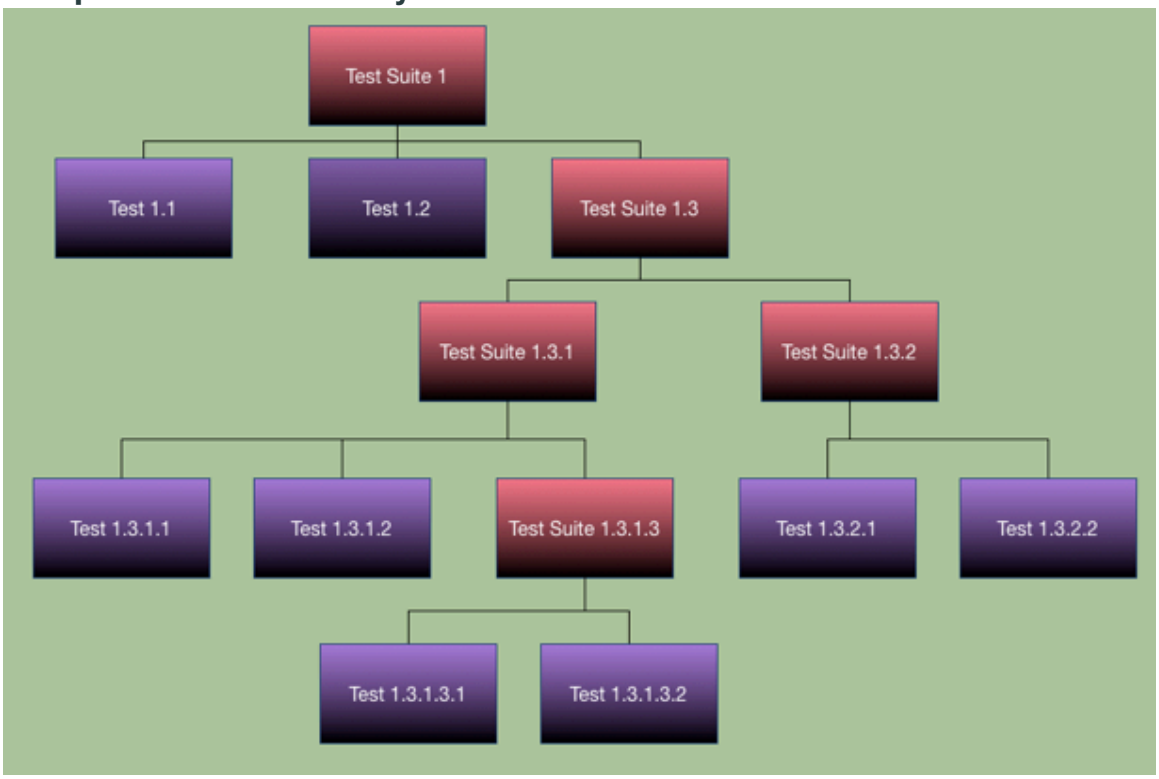
- Allows testers to run every test in a test suite with one action.
- Allows testers to run all child test suites in a parent test suite.
- Allows testers to see test results for every test in a test suite.
- Allows test designers and testers to schedule when to run test suites.
- Allows test designers and testers to schedule starting client test runners to support test runs.

Hierarchies

Automated Test Framework supports building a multi-level hierarchies where a test suite can be both a parent and child. For example, this figure illustrates Test Suite 1 as the parent at the top of the hierarchy. Test Suite 1.3 is a child of Test Suite 1 and also a parent of Test Suite 1.3.1 and Test Suite 1.3.2.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example of Test Suite Hierarchy



Schedules

To schedule a test suite, you need three components:

- a test suite record
- a schedule record specifying when you want the system to run the test suite
- a scheduled suite run record that associates the test suite to run with the schedule for running it

With this model, you can associate a schedule with many different test suites, and vice versa.

Note: You can schedule only test suites, not individual tests. Scheduled tests will run only if there is an open Scheduled Client Test Runner page matching the scheduled suite's browser conditions. Scheduled tests cannot run on a machine that is locked, powered down, or does not already have the browser open.

The watchlist on the test suite run record also allows you to specify users to receive an email when the system finishes executing the test suite run.

If the test suite contains one or more form steps (steps involving a user interface), you must ensure that a scheduled client test runner is actively running in a browser when the schedule triggers the suite run.

Note: See Browser recommendations and requirements for recommendations and requirements for running the client test runner.

For step-by-step instructions on how to schedule a test suite, see Schedule an automated test suite.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Filters

Automate the creation of test suites by using a filter to dynamically add tests to a test suite when they match the filter conditions. Reduce the time that your test designers spend manually creating and maintaining test suites.

Related topics

Suites

Step results record

Create an automated test suite

Group automated tests into a suite you can execute as a batch.

Before you begin

The tests you want to include in the test suite must exist.
Role required: atf_test_admin or atf_test_designer

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Suites**.
2. Click **New**.
   The system displays the Test Suite New Record form.
3. In the **Name** field, enter a name for this suite.
4. Select the tests to be included in this suite.

Option	Description
Filter	Add tests dynamically using a filter: **a.** In the **Filter** field, use the condition builder to create the conditions a test must match for inclusion in the test suite. **b.** Click **Save**. All tests that match the filter conditions appear in the **Test Suite Tests** related list. Because the suite is dynamic, any new test that matches the conditions is automatically added to the suite.
Test Suite Tests	Add tests manually: **a.** In the **Test Suite Tests** related list, click **Insert a new row...**. **b.** In the **Test** field, enter the name of the test to add to this test suite.

5. In the **Description** field, enter a description for this test suite.
6. In the **Test Suite Tests** related list, specify options for a test.

Option	Description
Execution order	Enter a value to specify the order in which this test should execute within the test suite.
Abort on Failure	Enter a value that indicates whether you want the test suite to stop or continue if this test fails.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Description
	○ By default, the system assigns the value **false** to this field. If this test fails, the system still executes any further tests in the test suite. ○ Set the value to **true** to stop the test suite if this test fails.

7. Repeat steps 4 - 6 for every test you want to include in this test suite.
8. Click **Submit**.

Copy an automated test suite

Reduce time when creating tests by copying an entire test suite. Rename and modify the test suite after copying. The **Copy Test Suite** button on the Test Suite form copies all the nested tests and child test suites within the text suite.

Before you begin

Role required: atf_test_admin or atf_test_designer

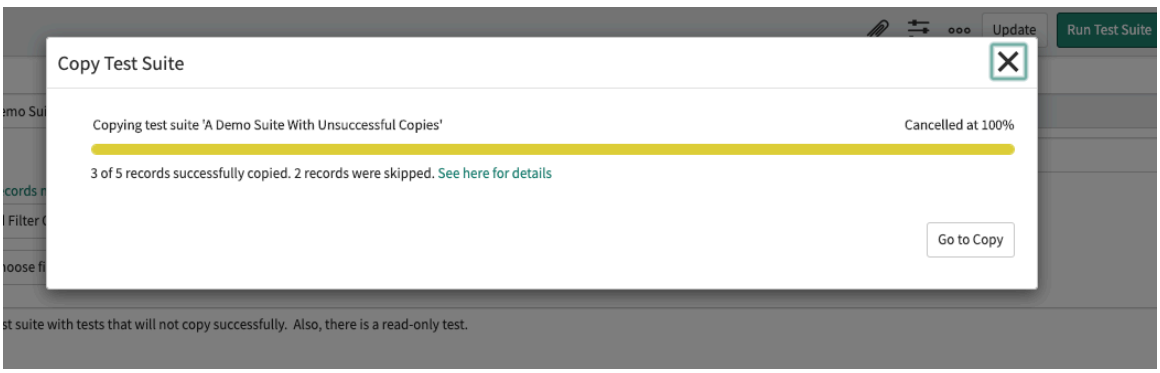
Procedure

1. Navigate to **All** > **Automated Test Framework** > **Suites**.
2. Select the row containing the test suite that you want to copy.
3. Near the top corner of the Test Suite form, select **Copy Test Suite**.
   A progress tracker appears to confirm the status. If you select **Go to Copy**, the system displays a new test suite record identical to the copied record, except for the **Name**.

   Note: If you apply filter conditions, copied tests are added to the test suite if the conditions are met.

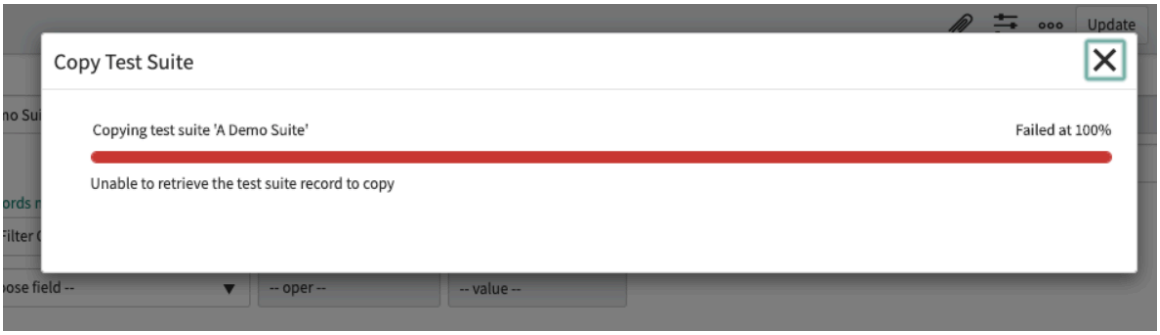
   If any tests or child test suites fail to copy, they are skipped and copying of the remaining tests and child test suites continues. A warning message showing a partial success appears below the progress bar.

   Note: You can copy a test suite only when it's in the same scope as the current scope. The **Copy Test Suite** button appears only when the test suite is in the same scope. The scopes of the tests in a suite are preserved when copying a test suite. See Application Scope for more information.



   If the copy operation fails, an error message appears below the progress bar.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



4. In the **Name** field, enter the name that you want to assign to this new test suite.
5. Edit the tests and test suites within the copied test suite.
6. When you are finished making changes, select **Update**.

Result
A copy of the test suite along with all the nested tests and child test suites is created.

Add tests to a suite with a filter

Automate the creation of test suites by using a filter to dynamically add tests to a test suite when they match the filter conditions. Reduce the time that your test designers spend manually creating and maintaining test suites.

Before you begin
The tests you want to include in the test suite must exist.
Role required: atf_test_admin or atf_test_designer

Procedure
1. Navigate to **All** > **Automated Test Framework** > **Suites**.
2. Click **New**.
   The system displays the Test Suite New Record form.
3. In the **Name** field, enter a name for this suite.
4. For **Filter**, use the condition builder to specify the conditions a test must meet to be added to the test suite.

   Note: If you don't have a filter condition set, it is ignored and you can build the suite manually.

5. Click **Save**.

Result
All tests that match the filter conditions appear in the **Test Suite Tests** related list. Because the suite is dynamic, any new test that matches the conditions is automatically added to the suite.

Add test to an existing automated test suite

Add a test to a test suite that already exists.

Before you begin
The tests you want to include and the test suite must exist.
Role required: atf_test_admin or atf_test_designer

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Suites**.
2. Click the row containing the test suite you want.
   The system displays the Test Suite form.
3. In the **Test Suite Tests** related list, click **Insert a new row...**.
4. In the **Test** field, enter the name of the test to add to this test suite.
5. In the **Order** field for this row, enter a value to determine the order in which this test should execute within the test suite.
   By default, the system assigns a value to this field according to the order in which you add the tests.
6. In the **Abort on Failure** field for this row, enter a value that indicates whether you want the test suite to stop or continue if this individual test fails.
   By default, the system assigns the value **false** to this field. **False** means that if this test fails, the system still executes any further tests in the test suite.
7. Repeat steps 3 - 6 for every test you want to include in this test suite.
8. Click **Submit**.

Add child test suite to parent test suite

Add to a multi-level test suite by including a child test suite within a parent test suite.

Before you begin

The parent test suite must exist.
Role required: atf_test_admin or atf_test_designer

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Suites**.
2. Access a child test suite.
   - If the child test suite exists, open the child test suite form for editing.
   - If the child test suite does not yet exist, create it.
3. In the test suite form in the **Parent suite** field, enter the name of the test suite you want to act as the parent to this child.
4. If desired, add one or more tests to the child test suite.
5. Click **Submit**.

Run an automated test suite

After creating an automated test suite, run it in a non-production instance.

Before you begin

You must have created the test suite you want to run.
The test execution property must be enabled. You must have an admin or atf_test_admin role to do so.

Note: The test execution property is disabled by default to prevent running tests on a production system. Run tests only on development, test, and other sub-production instances.

Role required: atf_test_admin, atf_test_designer, or admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

About this task

This procedure outlines how to start a test suite manually. You can also schedule test suites to run at a later time. For more information, see Working with scheduled test suites.

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Suites**.
2. If necessary to view the Test Suites list, click **Test Suites**.
3. Click the row containing the test suite you want to run.
   The system displays the Test Suite form.
4. Click **Run Test Suite**.

   Note: If the test execution property is not enabled, the **Run Suite** button does not appear. In this case, see the annotation at the top of the form, and click the link to enable running tests.

   If the tests associated with the test suite include a form step (any step involving a UI), or other kinds of UI test steps, the Pick a Browser dialog appears before executing the tests.

5. Choose among any currently running test clients, or start a new runner.

   For more information, review Browser recommendations for all tests and suites.

   If the tests associated with the test suite only include server test steps, the system executes the tests without displaying the Pick a Browser dialog.

What to do next

Monitor the progress of the tests. When complete, click **Go to results** on the progress dialog window to display the **Test Results** list, where you can view and analyze the results.

Schedule an automated test suite

Schedule one or more test suites to run at a specific date and time.

Before you begin

You must have created the test suites you want to schedule.
Role required: atf_test_admin or atf_test_designer

About this task

To schedule a test suite, you need three components:

- A test suite record
- A schedule record specifying when you want the system to run the test suite
- A scheduled suite run record that associates the test suite to run with the schedule for running it

For more information about the capabilities of and requirements for scheduled test suites, see Working with scheduled test suites.

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Schedules**.
   The system displays the list of existing test suite schedules.
2. To create a new schedule, click **New**.
   The system displays the Suite Schedule record form.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

3. On the Suite Schedule record form, enter the name of the schedule, the frequency with which to run associated suites, the time at which to run associated suites, and the timezone for this schedule.
4. **Optional:** Specify a condition that must be met for running associated test suites by selecting **Conditional**, then fill in the **Condition** text box with the appropriate script.
5. Add a test suite to run by navigating to the Scheduled Suites related list, then clicking **New**. The system displays the Scheduled Suite Run record form.
6. On the Scheduled Suite Run record form enter the appropriate data.
   a. Enter the test suite to run.
   b. If the suite contains UI steps, enter any client constraints you wish to apply (such as browser to use).
   For more information on client constraints, see Scheduled Suite Run.
   c. Add to the record's watchlist users you want the system to inform (by email) when the scheduled suite has finished.
   d. Click **Submit**.
7. **Optional:** Add more test suites to this schedule by repeating steps 5 and 6.
8. Click **Update**.
9. If your suite includes any test steps that work with a form – or any other element on the client side, open a browser window for running the client portion of the scheduled tests.
   a. Review Browser recommendations and requirements for all tests and suites, as well as those that apply only to scheduled suite runs.
   b. In the **Navigator**, right-click **Scheduled Client Test Runner**, and then click the option to open in a separate tab or window, as you prefer.
   c. Leave open the browser window that's running the client test runner and return to the browser window that contains the **Navigator**.

Run a scheduled test suite using a script

Execute a scheduled UI test suite immediately using a script without having to wait for the scheduled time. You can use this method while trying to automate the process of running a test.

Before you begin

You've created and scheduled the test suites that you want to run. See Create an automated test suite and Schedule an automated test suite, for more information.

Role required: atf_test_admin or atf_test_designer

About this task

The following steps might not be in line with your software configurations.

Procedure

1. Spin up a virtual machine (VM) on an operating system with the necessary browsers.
2. Open a browser on the instance and navigate to the Scheduled Client Test Runner.
3. Call the scriptable method `new sn_atf.ScheduledRunsExecutor().setScheduleSysId`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

`("SYS_ATF_SCHEDULE_SYS_ID_HERE").start();` to run the scheduled suite immediately.
To run a test only when the script is called, set the **Run** field to **On Demand** in the Schedule form.

Note: The `start()` method returns `sys_progress_worker.sys_id` of the progress worker.

Re-run failed tests in an automated test suite

Re-run failed tests within a test suite without rerunning the entire suite.

Before you begin
Role required: atf_test_admin, atf_test_designer, or admin

About this task
The **Re-run failed tests** button appears on the Suite Result form and on the Suite Execution Progress Viewer after a suite with failed tests completes. It does not appear if test execution is disabled, the suite is deactivated, the suite passed, or the user does not have one of the required roles.

The **Re-run failed tests** button re-runs all non-passing tests, which includes tests with the following status: canceled, skipped, failure, and error. It does not include the test results with a Success with warning(s) status.

Procedure
1. Navigate to either the suite result form or suite execution progress viewer for the completed suite that had failed tests.
2. Click **Re-run failed tests**.
   If the re-run tests include a form step (any step involving a UI) or other kinds of UI test steps, the Pick a Browser dialog appears before executing the tests.
3. In the dialog box, choose among any currently running test clients or start a new runner.
   If the re-run tests include only server test steps, the system executes the tests without displaying the Pick a Browser dialog. For more information, review Browser recommendations for all tests and suites and Server test steps.

Result
The system re-runs the failed tests.

- The system creates a new suite result hierarchy for the re-run tests. The Progress Workers, Test Result, and Test Suite Result forms show the same suite hierarchy as the previous test suite. They do not include the tests or suites that passed in a previous run.
- If you delete or deactivate a child suite or test that failed, and then re-run it, the system does not execute that suite or test in the re-run.
- If you add a child suite or test to the suite to a failed test, and then re-run it, the system does not execute the added suite or test in the re-run.

What to do next
To view the results from the previous run of a test or suite, click **Previous test result** on the test result form or **Previous suite result** on the test suite result form. These fields only appear for tests and suites that have been re-run.

Auto-generate ATF tests

Auto-generate ATF tests by selecting the auto-generate option either from the Auto-generate Tests module or Tests/Suites modules.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin
Role required: admin

Procedure
1. Access the Auto-generate Tests feature using either of the following ways.

Option	Steps
Using Auto-generate Tests module	**a.** Navigate to **All** > **Automated Tests Frame work (ATF)** > **Auto-generate Tests**. **Note:** The ATF Test Generator and Cloud Runner store app link shows up. The Auto-generate Tests module shows up only if you don't have the ATF Test Generator and Cloud Runner store app installed. **b.** Select the **Check out the app on Store** op tion. **Note:** The ATF Test Generator and Cloud Runner store app page shows up.
Using Tests or Suites module	**a.** Navigate to **All** > **Automated Tests Frame work (ATF)** > **Tests**. **Note:** You can also select **Suites** modules instead of **Tests**. **b.** Scroll down to select the **Auto-generate Tests** link. **Note:** This link takes you to the Test Generator home page if you have the store app installed.

2. Optional: Install the ATF Test Generator and Cloud Runner store app.

Note: This step is applicable only if the store app is not installed on your instance. You will be redirected to install the store app.

Cancelling automated tests and test suites
You can cancel automated tests and automated test suites that are running or are queued to run.

How you cancel an automated test or automated test suite depends on whether the test or test suite is currently running or is queued to run.

Cancel queued automated test suite
You can cancel an automated test suite that is queued but has not yet run.

Before you begin
Role required: atf_test_admin or atf_test_designer

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Run** > **Waiting/Running Test Runs**.
2. Select the action check boxes for the test suites to cancel.
3. From the action choice list, select **Delete**.

Cancel running and pending tests in an automated test suite

Cancel running and pending tests in a running suite test.

Before you begin
Role required: atf_test_admin or atf_test_designer

Procedure

1. If necessary, display the Run Test progress dialog.
2. Click **Cancel Pending Steps**.
   The system displays a dialog asking you to confirm that you want to cancel this test execution.
3. In the Confirmation dialog, click **Yes**.

Result
The system cancels any running and pending tests in this test suite and rolls back any data that tests in the test suite may have changed.

Cancel running automated test

You can cancel a running test from the Run Test progress dialog.

Before you begin
The system must be running a test.
Role required: atf_test_admin or atf_test_designer

Procedure

1. If necessary, display the Run Test progress dialog.
2. Click **Cancel Pending Steps**.
   The system displays a dialog asking you to confirm that you want to cancel this test execution.
3. In the Confirmation dialog, click **Yes**.

Result
The system cancels the running test and rolls back any data changes the test made.

Cancel waiting automated test

You can cancel a waiting automated test from the Waiting/Running Test Runs module.

Before you begin
Role required: atf_test_admin or atf_test_designer

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Run** > **Waiting/Running Test Runs**.
2. Select the action check boxes for the tests to cancel.
3. From the action choice list, select **Delete**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Automated Test Framework (ATF) test types and techniques

Experience the different types of ATF tests and testing techniques.

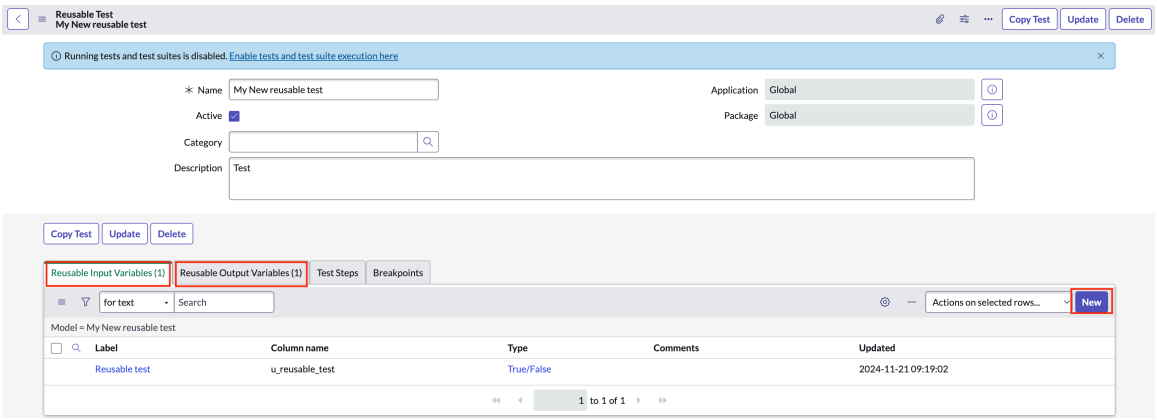
Reusable tests

Leverage reusable tests to simplify test maintenance and streamline the management of large tests and test suites. Reusable tests reduce redundancy, ensuring consistent and reliable test execution across your system.

Note: A reusable test can be invoked only from another test (a regular test or a reusable test). The reusable test can't be added to test suites individually.

Use the following related lists to pass variables from one test to another. Select **New** to create either the reusable input or output.

- Reusable Input Variables: Contains the reusable input variables that are passed in from the parent test. The variable is passed from the test calling the current reusable test.
- Reusable Output Variables: Contains the reusable output variables that are passed in from the child test. The variable is intended to be passed back up to the parent test calling the current reusable test.



Note: The reusable input and output variables that you create for the test are shown under the Reusable Input Variables and Reusable Output Variables related lists. The input variables are passed in from the parent test that is calling the current reusable test and is used by the reusable test. The output variables are passed back to the parent test and are used only by its test steps.

Use the Reusable Test test step category to access the created reusable test records. See Reusable Tests category for more information.

Create a reusable test

Create a reusable test to avoid redundancy, ensuring better test maintenance and reliable test execution across the instance.

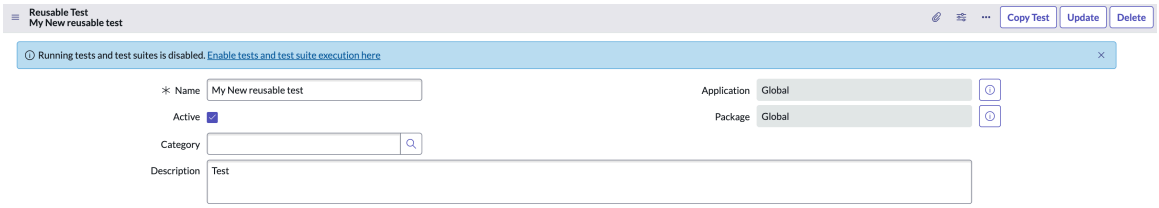
Before you begin

Role required: atf_test_admin or atf_test_designer

Procedure

1. Navigate to **All** > **Automated Test Framework (ATF)** > **Reusable Tests**.
   A list of reusable tests created on the instance shows up.
2. Select **New** to create a new reusable test.

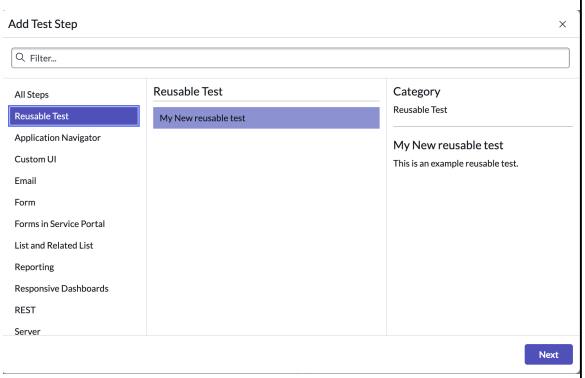
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



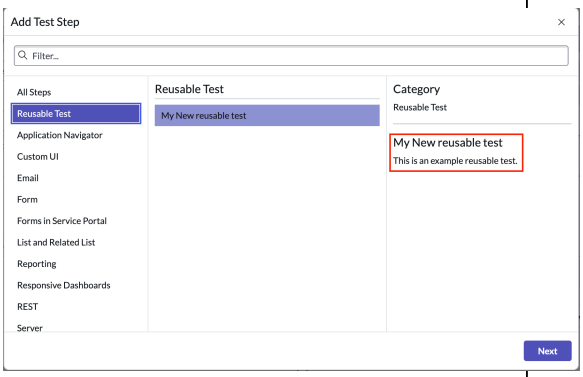
The Reusable Test form shows up.

3. On the form, fill in the fields.
   The read-only fields are filled up automatically.

Inputs

Field	Description
Name	Name of the reusable test.
Application	Read-only field with the scope of the test.
Active	Option to define the test as active or not.
Category	The test step category in which the test shows up.  **Note:** If you leave this field empty, the test shows up under the Reusable Tests test category by default.
Description	Description of the reusable test.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
	Note: The description of the reusable test is displayed in the step creator modal.  See Reusable Tests category for more information.

4. Select **Save** to create the reusable test.
   The reusable test shows up.

   Note: Reusable tests can't run independently. They can run only when it becomes a part of another test. They also can't be added to a test suite independently. They must be invoked in another test within a test suite.

Mutually exclusive tests

Prevent conflicting tests from running in parallel by marking them as mutually exclusive. For example, when the system identifies tests that modify the same record, the system makes these tests mutually exclusive. You can also manually mark tests as mutually exclusive.

The system marks tests as mutually exclusive when there is a potential resource conflict. When the system can't detect resource conflicts automatically, you can create your own mutual exclusion rules that can prevent conflicting tests from running in parallel. For example, if a test changes a sys_properties record, the record shows up under **Records Modified** of that test. If the validation path of another test depends on the same sys_properties record without any change, that test fails. This can occur if the previous test runs at the same time.

You can view mutually exclusive tests on the Mutually Exclusive Tests related list in the test form. This related list shows all tests that don't run in parallel with the current test and the reason. The same test might appear more than once in the list if there are multiple reasons.

Mutually exclusive tests reasons

Tests are marked mutually exclusive for the following reasons.

- Two or more tests modify the same record.
- A test that runs in parallel with itself.
- You can create your own mutual exclusion rules when the system can't detect resource conflicts automatically.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Mark tests as mutually exclusive

You can mark tests as mutually exclusive using any of the following methods.

Mutually Exclusive Tests tab

Select a test from the Tests list and navigate to **Mutually Exclusive Tests** > **Add Mutual Exclusion** to make the selected test mutually exclusive with another test.

Tests list

Select one or more tests from the Tests list and choose **Add mutually exclusive test** from the Action on selected rows context menu. Enter one test in the Add mutually exclusive test dialog box to make the selected tests mutually exclusive with the current test.

Parallel Test Runs tab

When two or more tests run in parallel, navigate to **Test Results** > **Parallel Test Runs**. Select one or more tests and choose **Add mutually exclusive test** from the Action on selected rows context menu to mark the selected tests as mutually exclusive.

Note: The **Parallel Test Runs** tab is visible only if the test runs in parallel with one or more tests.

Quick start tests

Copy and customize quick start tests provided by the ServiceNow AI Platform® to validate that your instance works after you make any configuration changes. For example, if you apply an upgrade or develop an application.

The tests can only produce a pass result when you run them with the default demo data that's provided with the application or feature plugin.

Note: If your QST fails, it can be that the test doesn't match your instance customizations. Use the **Notes** field for each test step to update the test to pass with your data. If you have customized the business process associated with a test, revise the test to match your customizations.

To apply a quick start test to your instance-specific data, copy the quick start test and add your custom data.

Note: You can copy either an individual quick start test or the entire quick start tests suite.

See Getting started with quick start tests for more information.

Activation

Each application or feature has its own plugin activation requirements for enabling quick start tests. See Available quick start tests by application or feature for activation information.

Managing copies

When you copy a test, Automated Test Framework populates the **Copied from** field with the name of the copied test. When an upgrade changes, a quick start test Automated Test Framework notifies test designers about the change in a notification on the test form. Test designers can revert the copied test to the upgraded version with the **Revert Copy to Base System** UI action.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Tip: If a QST is updated, the previously copied versions of the QST does not have the recent changes. Review the old copies using **Copies to Review** for all the tests that have an associated warning message.

Available quick start tests by application or feature

Validate that your instance still works after you make any configuration change such as apply an upgrade or develop an application. Copy and customize the ServiceNow -provided quick start tests to pass when using your instance-specific data.

Danger: By default, the system property that is used to run automated tests is disabled to prevent you from accidentally running these tests on a production system. To avoid data corruption or an outage, run tests only on development, test, and other non-production instances. See Enable or disable executing Automated Test Framework tests.

Agile Development 2.0

Agile Development 2.0 quick start tests require activating the Agile Development 2.0 plugin (com.snc.sdlc.agile.2.0) and the Agile Development 2.0 - ATF Tests plugin (com.snc.sdlc.agile.2.0.atf).

Agile Development 2.0 quick start test suite

Test	Description	Release version
Verify that global rank is populated when a story is created	Verify the global rank of a story after creation.	Madrid
Verify that closing a sprint with active stories is prevented	Verify that a sprint with active stories cannot be closed.	Madrid
Verify that sprints cannot overlap in the same group	Verify that sprints in the same group do not overlap.	Madrid
Verify that sprint points are updated	Verify that changes to stories produce accurate sprint point totals.	Madrid
Verify that only one sprint in a group can have the current state	Verify sprint statuses.	Madrid
Verify sprint end date is after the sprint start date	Verify sprint start and end dates.	Madrid
Verify that any update on story rolls up to Epic	Verify that adding, estimating, removing, deleting, updating, or cancelling a story updates the epic-level roll-ups correctly.	Orlando
Verify changes to the scope of a current sprint do not alter the value of the Total Committed Points	Verify that the value of Total Committed Points does not change with change in the scope of a sprint after its state is changed to Current.	Orlando
Verify active flag is set false when Agile Story state is	Verify that active flag of an Agile story is set to the following:	Orlando

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Agile Development 2.0 quick start test suite (continued)

Test	Description	Release version
changed to Completed/ Cancelled	• False, if the state is changed to Completed or Cancelled • True, for all other states	
Verify that updating the team/ group capacity overrides the capacity on all the future sprints	Verify that any update to the **Group capacity** field of the assignment group results in the following changes to the **Group capacity** field of the various sprints associated with this assignment group: • For the sprints that are in the Draft and Planning state: ○ The group capacity is updated to the new value. ○ The **Group capacity** field is editable. • For the sprints in the Current, Complete, or Cancelled state: ○ The group capacity remains the old value. ○ The **Group capacity** field is read-only. For the sprints in the Draft or Planning state, you can individually edit the group capacity of the sprint anytime later. This would not change the group capacity of the assignment group associated with this sprint.	Paris
Verify create and edit functionality of an epic backlog on the scrum program board	• Verify that you can create an epic backlog for scrum programs from the Backlog tab of Agile Board. • Verify that you can update an existing epic backlog for scrum programs from the Backlog tab of Agile Board. • The epics listed in the backlog must belong to the selected epic backlog.	Quebec

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Agile Development 2.0 quick start test suite (continued)

Test	Description	Release version
Verify a Demand is converted to a scrum story	If the PPM Standard plugin (com.snc.financial_planning_pmo) is active, verify that a Demand can be converted to an Agile 2.0 story using the **Create Story** related link on the Demand form. For more information on how to create an Agile 2.0 story from a demand, see Create an artifact from a demand.	Quebec
Verify a Demand is converted to a scrum epic	If the PPM Standard plugin (com.snc.financial_planning_pmo) is active, verify that a Demand can be converted to an Agile 2.0 epic using the **Create Epic** related link on the Demand form. For more information on how to create an Agile 2.0 epic from a demand, see Create an artifact from a demand.	Quebec

To learn more about Agile Development 2.0, see Agile Development 2.0.

Enterprise Architecture (formerly Application Portfolio Management)

Enterprise Architecture quick start tests require enabling the Enterprise Architecture – ATF Tests plugin (com.snc.apm.atf).

Enterprise Architecture: Create Business application and capability test suite

Test	Description	Release version
Enterprise Architecture: Create Business Application	Verify the creation of an application category and then the creation of a business application with users having apm_user role.	Madrid
Enterprise Architecture: Create Business Capability	Verify the creation of a parent and child business capability and verify its field values.	Madrid
Enterprise Architecture: Test relating Business Service, Business Application, and Software Models	Verify the creation of a business application, business service, using the existing software model, and a relationship between them.	Orlando

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Enterprise Architecture: Create Business application and capability test suite (continued)

Test	Description	Release version
Enterprise Architecture: Test for Indicator Score and Application Score generation	Verify the creation of indicator, scoring profile, and generation of indicator scores and application scores.	Paris
Enterprise Architecture: Business Application with Information Object and Data Domain	Verify the creation of business application, information object, and addition of the CRUD operations in relation attributes.	Quebec

To learn more about Enterprise Architecture (formerly Application Portfolio Management), see Enterprise Architecture .

Assessments and Surveys

Assessments and Surveys quick start tests require activating the Automated Test Framework for Survey plugin (com.glide.automated_testing_impl. Survey).

Test Suite for Survey

Name	Description	Release version
Survey: Basic Platform Based Test	Create a survey using Platform UI actions.	Madrid
Survey: Platform test for Dynamic Validation	Validate a survey dynamically.	Madrid
Survey: Clone Action	Clone a survey and validate the records of the original survey and the cloned survey.	New York
Survey: Question Bank Flow	Verify the addition of a question bank to a survey.	New York
Survey: Survey Creator Work Flow	Survey creator can create a survey and assign to the user who can take the survey and submit it.	New York
Assessment: Assign assessment to assessor	Assign an assessment to an assessor and verify that the instance is created for the assessor.	Orlando
Assessment: Create assessment as survey creator	Create an assignment as a survey creator.	Orlando
Assessment: Scale and Template support	Create, publish, and assign the assessment. Then take the assessment.	Rome
Assessment: Basic test flow	Verify the basic flow of an assessment.	Orlando

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

To learn more about Assessments and Surveys, see Service administration.

Change Management

Change Management quick start tests require activating the Change Management - ATF Tests plugin (com.snc.change_management.atf).

CHG: Emergency Type Change Request test suite

Test	Description	Release version
Emergency Type Change Request workflow	Process an emergency change request from new to closed.	Madrid
On Hold for Emergency type Change Request	Validate the approval state of an on-hold emergency change request.	Madrid
Copy Change For Emergency type Change Request	Validate the state of a copied emergency change request.	Madrid
Reject By Approver for Emergency type	Validate the state of a rejected emergency change request.	Madrid
Revert to new for emergency type	Validate the state of an emergency change request after using the Revert to new UI action.	Madrid
Convert Emergency to Normal type	Validate the conversion of an emergency change request to a normal change request.	Madrid
Cancel Change Request For Emergency Type	Validate the state of a canceled emergency change request.	Madrid

CHG: Normal Type Change Request test suite

Test	Description	Release version
Normal Type Change Request Workflow	Process a normal change request from new to closed.	Madrid
Convert Normal to Emergency type	Validate the conversion of a normal change request to an emergency change request.	Madrid
Copy change on Normal Change Request	Validate the state of a copied normal change request.	Madrid
On hold for Normal type Change Request	Validate the approval state of an on-hold normal change request.	Madrid
State validation when Reject Normal type Change request by Approver.	Validate the state of a rejected normal change request.	Madrid
Revert to New Functionality for Normal Type Change Request	Validate the state of a normal change request after using the Revert to new UI action.	Madrid

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

CHG: Normal Type Change Request test suite (continued)

Test	Description	Release version
Cancel Change Request For Normal type	Validate the state of a canceled normal change request.	Madrid

CHG: Standard Change Proposal test suite

Test	Description	Release version
Standard Change Proposal	Determine whether a user can successfully perform standard change proposal creation, approval, and template publishing processes.	Madrid

CHG: Standard Type Change Request test suite

Test	Description	Release version
Standard Change Request Workflow	Process a standard change request from new to closed.	Madrid
Convert Standard to Emergency Change Request	Validate the conversion of a standard change request to an emergency change request.	Madrid
Convert Standard to Normal Change Request	Validate the conversion of a standard change request to a normal change request.	Madrid

CHG: Unauthorized Change Request and Outage test suite

Test	Description	Release version
Unauthorized change request Workflow	Process a unauthorized change request from new to closed.	Orlando
Create Outage of type planned outage from change request	Validate the creation of an outage of type planned outage from a change request.	Orlando
Create Outage of type outage from change request	Validate the creation of an outage of type outage from a change request.	Orlando

CHG: Risk Conditions with Best practice plugin test suite

Test	Description	Release version
Moderate Risk with UI Action Property	Process a moderate risk with UI action property.	Orlando
Low Risk with UI Action Property	Process a low risk with UI action property.	Orlando
Leave Alone Risk with UI Action Property	Process a leave alone risk with UI action property.	Orlando

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

CHG: Risk Conditions with Best practice plugin test suite (continued)

Test	Description	Release version
High Risk with UI Action Property	Process a high risk with UI action property.	Orlando
High Risk with Business Rule Property	Process a high risk with Business rule property.	Orlando
Moderate Risk with Business Rule property	Process a moderate risk with Business rule property.	Orlando

CHG: Change Request against Conflict Sources test suite

Test	Description	Release version
Change against Blackout window	Process a change request against a blackout window.	Orlando
Check conflicts for CI Already Scheduled	Validate the conflicts for CI already scheduled.	Orlando
Change against Conflict Sources	Validate the change request against conflict sources.	Orlando

CHG: Change Schedule Definition and Sharing test suite

Test	Description	Release version
Change Schedules Definitions on New	Process the creation of change schedules definitions from New button on Change Schedules landing page.	Orlando
Share Panel On Change Schedules Definition	Validate the share panel on change schedules definition.	Orlando
Share Change Schedule Definition	Validate the sharing of change schedules definitions.	Orlando
Create Standard Change via Service Portal	Create Standard Change from Service Portal	Yokohama

To learn more about Change Management, see Change Management.

Cloud Provisioning and Governance

Cloud Provisioning and Governance quick start tests require activating the following Cloud Provisioning and Governance plugins:

- Cloud Provisioning and Governance (com.snc.cloud.mgmt)
- Domain Support- Domain Extensions Installer plugin (com.glide.domain.msp_extensions.installer)
- Service Catalog- Domain Separation plugin (glideapp.servicecatalog.domain_separation)

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Cloud Provisioning and Governance: Azure test suite

Test	Description	Release version
CMP: Add Azure Credentials	Add Azure credentials by inserting the fields into the Credentials table.	Orlando
CMP: Azure Service Account	Add Azure service account by inserting the fields into the service account table.	Orlando
CMP: Update datacenter type	Update the Cloud Service account with the datacenter type.	Orlando
CMP: Creating Resource Group	Creating Resource Group to provision Azure stack via ARM Template.	Orlando
CMP: Create Azure Logical datacenter.	Create Azure Logical datacenter for provisioning resources.	Orlando
CMP: Create "Contains: Contained by" relationship.	Create "Contains: Contained by" relationship between Resource Group and Azure datacenter.	Orlando
CMP: Create 'Hosted on' relationship.	Create hosted on relationship between logical datacenter and Cloud Service account.	Orlando
CMP: Create Cloud Account	Create cloud account by inserting the fields into the cloud account table.	Orlando
CMP: Cloud account and Logical datacenter association	Create an association between CMP cloud account and Logical datacenter.	Orlando
CMP: Create Catalog Item	Create Cloud Catalog Item with ARM Template.	Orlando
CMP: Create template version.	Create Cloud Template Version for Cloud Catalog Item.	Orlando
CMP: ARM template body	Update Cloud Template Version with ARM template body.	Orlando
CMP: Open the 'ServiceNow Cloud Template Versions'	Open the 'ServiceNow Cloud Template Versions' GUI page.	Orlando
CMP: Activate Cloud Template Version	Activate Cloud Template Version to create resource block, blueprint, and catalog.	Orlando
CMP: Verify the Cloud Template Version.	Validate whether status is Success and state is active in	Orlando

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Cloud Provisioning and Governance: Azure test suite (continued)

Test	Description	Release version
	the Cloud Template Version after Activation operation.	
CMP: Activate Cloud Catalog Item	Activate Cloud Catalog Item to display catalog order form to the cloud user.	Orlando
CMP: Add stack and resource group name in Catalog item	Set the default value for stack name and resource group in the catalog item.	Orlando
CMP: Verify whether stack is present	Validate whether stack is available with name "ATFARMStack".	Orlando
CMP: Open Cloud Catalog Item page	Open Cloud Catalog Item page in the Service Portal.	Orlando
CMP: Submit Catalog form	Submit the Catalog Form to start provisioning ARM stack.	Orlando
CMP: Verify whether stack status is active	Validate the status of a stack after provisioning the stack.	Orlando
CMP: Verify stack status after Day2 - Stop operation	Validate the status of the stack status changed from 'On' to 'Off' after the Stop operation.	Orlando
CMP: Verify stack status after Day2 - Start operation	Validate the status of the stack status changed from 'Off' to 'On' after the Start operation.	Orlando
CMP: Verify stack status after Day2 - Deprovision operation	Validate the status of the stack status changed from 'On' to 'Terminate' after the de-provisioning operation.	Orlando

Cloud Provisioning and Governance: AWS test suite

Test	Description	Release version
CMP: Add AWS Credentials	Add AWS credentials by inserting the fields into the credential table.	Orlando
CMP: AWS Service Account	Add AWS service account by inserting the fields into the service account table.	Orlando
CMP: Update datacenter type	Update the cloud service account with the datacenter type.	Orlando
CMP: Create logical datacenter	Create Logical datacenter for provisioning the resources.	Orlando

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Cloud Provisioning and Governance: AWS test suite (continued)

Test	Description	Release version
CMP: Create hosted on relationship	Create hosted on relationship between logical datacenter and Cloud Service account.	Orlando
CMP: Create Cloud Account	Create cloud account by inserting the fields into the cloud account table.	Orlando
CMP: Cloud account and Logical datacenter association	Create an association between CMP cloud account and logical datacenter.	Orlando
CMP: Create Catalog Item	Create Cloud Catalog Item with the CFT Template.	Orlando
CMP: Create template version	Created Cloud Template Version for a Cloud Catalog Item.	Orlando
CMP: CFT template body	Update Cloud Template Version with the CFT template body.	Orlando
CMP: Open the ' ServiceNow Cloud Template Versions'	Open the ' ServiceNow Cloud Template Versions' GUI page.	Orlando
CMP: Activate Cloud Template Version	Activate Cloud Template Version to create resource block, blueprint, and catalog.	Orlando
CMP: Verify the Cloud Template Version	Validate the Cloud Template Version whether status is Success and state is active after Activation operation.	Orlando
CMP: Activate Cloud Catalog Item	Activate Cloud Catalog Item to display the catalog order form to the cloud user.	Orlando
CMP: Add stack name in Catalog item	Set the default value for stack name in the catalog item.	Orlando
CMP: Verify whether stack is present	Validate whether stack is available with name "ATFCFTStack".	Orlando
CMP: Open Cloud Catalog Item page	Open the Cloud Catalog Item page in the Service Portal.	Orlando
CMP: Submit Catalog form	Submit the Catalog Form to start provisioning CFT Stack.	Orlando
CMP: Verify whether stack status is active	Validate the status of the stack after provisioning the stack.	Orlando
CMP: Verify stack status after Day2 - Stop operation	Validate the status of the stack status changed from 'On' to 'Off' after the Stop operation.	Orlando

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Cloud Provisioning and Governance: AWS test suite (continued)

Test	Description	Release version
CMP: Verify stack status after Day2 - Start operation	Validate the status of the stack status changed from 'Off' to 'On' after the Start operation.	Orlando
CMP: Verify stack status after Day2 - De-provision operation	Validate the status of the stack status changed from 'On' to 'terminate' after the de-provisioning operation.	Orlando

To learn more about Cloud Provisioning and Governance, see Cloud Provisioning and Governance.

Coaching

Coaching quick start tests requires activation of the Coaching plugin (com.sn_coaching).

Coaching test suite

Test	Description	Release version
Coaching: Create an assessment manually when logged in as a coach.	As a coach, verify that you can create an assessment and assign it to trainees.	Orlando
Coaching: Add skills to an opportunity and verify those skills awarded to trainee.	Add skills to a coaching opportunity and verify that those skills have been awarded to trainees after they complete an assessment.	Orlando
Coaching: Complete an assessment as a virtual coach.	Verify that the virtual coach completes an assessment and provides feedback to the trainee when a virtual coach is attached to an opportunity.	Orlando
Coaching: Add skills to an assessment and verify those skills awarded to trainee.	Add skills to a coaching assessment and verify that those skills have been awarded to trainees after they complete an assessment.	Orlando
Coaching: Add skills to a recommendation and verify those skills awarded to trainee.	Add skills to a coaching recommendation and verify that those skills have been awarded to trainees after they complete an assessment.	Orlando
Coaching: Move assessments from one state to another when logged in as a coach.	As a coach, verify that you can move an assessment from one state to another.	Orlando
Coaching: Attach a recommendation learning to an assessment.	Verify that a recommendation learning on an opportunity gets attached to an	Orlando

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Coaching test suite (continued)

Test	Description	Release version
	assessment when an assessment is generated.	
Coaching: When a coaching opportunity is inactive, assessments are not generated.	Verify that when a coaching opportunity is in inactive state, it does not generate assessments.	Orlando
Coaching: Submit Coaching survey as a Coach user.	Verify that coach can submit survey for a trainee.	Quebec
Coaching: Submit Coaching survey as a Traineer user.	Verify that trainee can submit survey for a coach.	Quebec

To learn more about coaching, see Coaching.

Communities

Communities quick start tests require activating the Customer Communities plugin (com.sn_customer_communities) and the Communities Demo Data plugin (com.sn_communities_demo).

Communities: Community test suite

Test	Description	Release version
Post question and validate feed	Verify that a question is posted in the community and validate whether it appears in the content feed.	Madrid
Post video	Verify that a video is posted in the community.	Madrid
Forum membership approval	Verify that a membership request to a forum is approved.	Madrid
Forum membership reject	Verify that a membership request to a forum is rejected.	Madrid
Approve a moderation task	Verify that a moderation task is approved.	Madrid
Reject a moderation task	Verify that a moderation task is rejected.	Madrid
Topic subscription and activity feed validation	Verify that a topic is subscribed to and the topic activities appear in the activity feed.	New York
Approve content approval workflow task on question	Verify the content approval workflow of a question.	New York
Question auto-subscription and activity feed validation	Verify that the author is automatically subscribed to the question and the question	New York

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Communities: Community test suite (continued)

Test	Description	Release version
	activities appear in the activity feed.	
Follow a user	Verify that a community user is able to follow another community user.	New York
Reject content approval workflow task on question	Verify the content approval workflow of a question when content is rejected.	New York
Gamification on video posting	Verify the gamification points gained when posting a video.	New York
Forum subscription and activity feed validation	Verify that a forum is subscribed to and the forum activities appear in the activity feed.	New York
Question subscription and activity feed validation	Verify that a question is subscribed to and the question activities appear in the activity feed.	New York
Post a video with 'Disable comment' option	Verify a video is posted with *`Disable comments`* option as selected.	Orlando
Pin a video as Featured	Verify that a video is marked as featured by community administrator in the forum and community home page.	Orlando

To learn more about Communities, see Communities.

Configuration Compliance

Configuration Compliance quick start tests require activating the Configuration Compliance application (sn_vulc) and loading the demo data.

Configuration Compliance quick start tests

Test	Description	Release version
Configuration Compliance Assignment Rules	Create an assignment rule.	Yokohama
Configuration Compliance - Reapply Group Rule	Reapply a test result group rule.	Yokohama
Configuration Compliance - Delete Group Rule	Delete a test result group rule.	Yokohama

To learn more about Configuration Compliance, see Configuration Compliance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Configuration Management Database (CMDB)

Configuration Management Database (CMDB) quick start tests require activating the Configuration Management (CMDB) plugin (com.snc.cmdb) and the CMDB - ATF Tests plugin (com.snc.cmdb.atf).

CMDB BSM: Dependency Views test suite

Test suite to check functionality of Dependency Views APIs.

Test	Description	Release version
CMDB BSM: Dependency Views	Test functionality of Dependency Views APIs. These APIs retrieve Dependency Views map and associated map items such as context menu items, for a given CI sys_id and using sn_cmdb_editor or itil user role.	New York

CMDB HEALTH: CMDB Health Dashboard test suite

Test suite to check whether CMDB CMDB Health Dashboard is functional at a basic level.

Test	Description	Release version
CMDB HEALTH: Health Job Status	Checks whether any CMDB Health dashboard jobs, which were started 30 or more days ago, are still in progress.	New York
CMDB HEALTH: CMDB Health Completeness/ Recommended	Checks whether the recommended metric (included in the CMDB Health completeness KPI) is fully functional. This test validates: • Creation of a health inclusion rule for the recommend metric. • Creation of a recommended field that satisfies the health inclusion rule. • Validate that the health inclusion rule is correctly applied to a test record with missing data in the recommended field.	New York

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

CMDB IRE: Identification Reconciliation Engine test suite

Test suite to check Identification and Reconciliation Engine (IRE) functionality.

Test	Description	Release version
CMDB IRE: IRE Validation	Validate CI identifiers and reconciliation definitions and check indexes for CI identifiers.	Madrid
CMDB IRE: Reconciliation Rule	Check operations on a reconciliation rule, in CI Class Manager, using sn_cmdb_editor, sn_cmdb_admin, itil, and itil_admin roles. Operations include create, edit, and delete a reconciliation rule. Also, check for active and not active setting, and derived rules.	Paris
CMDB IRE: Identification Rule	Check operations on an identification rule, in CI Class Manager, using sn_cmdb_editor, sn_cmdb_admin, itil, and itil_admin roles. Operations include create, edit, and delete an identification rule. Also, check for active and not active setting, and derived rules.	Paris

CMDB QB: Query Builder test suite

Test suite to verify CMDB Query Builder functions such as create query, read query, and execute query using two related user roles - cmdb_query_builder and cmdb_query_builder_read.

Test	Description	Release version
CMDB QB: Query Builder - cmdb_query_builder Role	Verify that cmdb_query_builder user role can save queries, and access and run all saved queries, in CMDB Query Builder.	New York
CMDB QB: Query Builder - cmdb_query_builder_read Role	Verify that cmdb_query_builder_read user role can access and run all saved queries, and cannot	New York

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

CMDB QB: Query Builder test suite

Test suite to verify CMDB Query Builder functions such as create query, read query, and execute query using two related user roles - cmdb_query_builder and cmdb_query_builder_read.

(continued)

Test	Description	Release version
	save any query, in CMDB Query Builder.	

CMDB REL: Relationship Editor and Formatter test suite

Test suite to verify functionality of Relationship Editor and Relationship Formatter.

Test	Description	Release version
CMDB REL EDITOR:Relationship Editor	Check addition of relations to a CI and deletion of relations from a CI using sn_cmdb_editor and itil user roles.	New York
CMDB REL FORMATTER:Relationship Formatter	Check CI information, relationship types, relationships, associated records, and CMDB views displayed in relationship formatter using sn_cmdb_editor and itil user roles.	New York

CMDB SDK: SDK REST API test suite

Test suite to verify functionality of CMDB SDK Rest APIs.

Test	Description	Release version
CMDB SDK: Query CMDB Metadata	Test querying CMDB metadata.	New York
CMDB SDK: Create a relation for a CI using REST APIs	Test creation of a relationship for a CI in the CMDB using the CMDB REST APIs.	New York
CMDB SDK: Delete a relation for a CI using REST APIs	Test deletion of a relationship for a CI using CMDB REST APIs.	New York
CMDB SDK: Create a CI using REST API	Test creation of a CI using CMDB REST APIs.	New York
CMDB SDK: Query CMDB using REST APIs	Test querying the CMDB using CMDB REST APIs.	New York
CMDB SDK: Update a CI using REST APIs	Test updating of a CI using CMDB REST APIs.	New York

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

CMDB SDK: SDK REST API test suite

Test suite to verify functionality of CMDB SDK Rest APIs.

(continued)

Test	Description	Release version
CMDB SDK: Query for a CI using REST APIs	Test querying a CI using CMDB REST APIs.	New York

To learn more about Configuration Management Database, see Configuration Management Database.

Continual Improvement Management

Continual Improvement Management quick start tests require activating the Continual Improvement Management Automated Tests plugin (com.sn_cim.atf)

Continual Improvement Management tests

Test	Description	Release version
CIM: Inbound integration from Assessments	Create an improvement initiative from an assessment.	Paris
CIM: Inbound integration from Incident Management	Create an improvement initiative from an incident.	Paris
CIM: Inbound integration from Problem Management	Create an improvement initiative from a problem.	Paris
CIM: Inbound integration from Survey Management	Create an improvement initiative from a survey.	Paris

To learn more about Continual Improvement Management, see Continual Improvement Management.

Customer Service Management

All Customer Service Management quick start tests require activating the Customer Service Management Demo Data plugin (com.snc.customerservice.demo). Some quick start tests also require activating the following plugins:

- Business Location (com.snc.business_location)
- Case Playbook for Complaints (sn_complaint)
- Case Playbook for Onboarding (sn_onboarding)
- Consumer Service Portal (com.glide.service-portal.consumer-portal)
- CSM Contributor User (com.snc_csm_contributor_user)
- Customer Proxy Contacts (com.snc.csm_proxy_contacts)
- Customer Household Data Model (com.snc.household)
- Customer Service Management for Orders (com.snc.csm.order)
- Customer Service Portal (com.glide.service-portal.customer-portal)
- Customer Service with Request Management (com.sn_cs_sm_request)
- Customer Service with Service Management (com.sn_cs_sm)

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Guided Decisions Experience (com.snc.guided_decisions_playbook_experience)
- Guided Decision - Next Best Action (com.snc.next_best_action)
- Major Issue Management (com.sn_majorissue_mgt)
- Proactive Customer Service Operations with Event Management (com.snc.proactive_cs_itom)
- Skill Determination (com.snc.skill_determination)
- UX ATF Action (com.glide.ux.atf_action)
- Walk-Up for CSM (com.snc.walkup_for_csm)

CSM: Case Management test suite

Test	Description	Release version
CSM: Create Product Case	Create a case for a product.	Madrid
CSM: Assign Case to an Agent	Create a case and assign it to a customer service agent.	Madrid
CSM: Assign Asset on Case	Assign an asset to a case.	Madrid
CSM: Assign Entitlement	Assign an entitlement to a case.	Madrid
CSM: Escalate an Account	Escalate an account.	Madrid
CSM: Escalate a Case	Escalate a case.	Madrid
CSM: Create Special Handling Notes	Create special handling notes for a case.	Madrid
CSM: Close a Case	Close a case.	Madrid
CSM: Time Recording	Record the time worked on a case.	Madrid
CSM: Create CHG from Case	Create a change record from a case.	Madrid
CSM: Create Incident from Case	Create an incident record from a case.	Madrid
CSM: Create Order Case	Create a case for an order. **Note:** Requires Customer Service Management for Orders.	Madrid
CSM - Create Order Case as Customer from CSM Portal	Create an order case as a customer from the Customer Service Portal. **Note:** Requires the Customer Service Portal. Also requires that the test be run as admin.	Madrid
CSM: Create Problem from Case	Create a problem record from a case.	Madrid

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

CSM: Case Management test suite (continued)

Test	Description	Release version
CSM: Create Major Case and its Child Cases	Create a major case and the associated child cases for the customer accounts in the recipient list. **Note:** Requires Major Issue Management.	Orlando
CSM: Agent Creating Request for Customer	As a customer service agent, create a request for a customer. **Note:** Requires Customer Service Request Integration.	Orlando
CSM: Register New Case Type	Register a case type and verify the record is created.	Orlando
CSM - Create Project for an Account	Project Manager creates project for an account. **Note:** Requires Customer Project Management.	Orlando
CSM - Identify Project Contact	Project manager identifies customers to a project. **Note:** Requires Customer Project Management.	Orlando
CSM - Create Case from Project	Create a case from a project.	Orlando
CSM: Advanced Type Skill Determination Rule Test	Create an advanced type skill determination rule. **Note:** Requires Skill Determination.	Orlando
CSM - Create Case from Project Task	Create a case from a project task.	Orlando
CSM: Create Task from Case	Create a case task from a case.	Orlando
CSM: Assign Case Task to Case Task Agent	The customer service agent creates and assigns a case task to a case task agent.	Rome

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

CSM: Case Management test suite (continued)

Test	Description	Release version
CSM: Case Task Agent views assigned Case Task	Verifies that the case task agent can view an assigned case task.	Rome
CSM: Case Task Agent Completes Assigned Task	The case task agent completes an assigned case task.	Rome
CSM-Create a case using 'Create a case (POST)' API	Create a case using 'Create a case (POST)' API. **Note:** Requires a user authentication record assigned to the **Basic authentication** field in step 1 of the ATF.	Rome
CSM-Query a case using 'Query a case (GET)' API	Query a case using 'Query a case (GET)' API. **Note:** Requires a user authentication record assigned to the **Basic authentication** field in step 1 of the ATF.	Rome
CSM: Address sharing across Accounts	Implement reusable addresses for accounts to support complex customer operations and business models.	Washington DC
CSM: Manage Consumer Profile Locations	Create and manage consumer locations, and map them to consumer profiles.	Washington DC
CSM: OCS Manager creating New OSP	Verify that a new Outsourced Service Provider (OSP) is created by the OCS internal manager. **Note:** Requires Outsourced Customer Service.	Rome
CSM: Case Creation by OCS Agent	Verify that an Outsourced Customer Service agent is able to create a case. **Note:** Requires Outsourced Customer Service.	Rome

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

CSM: Case Management test suite (continued)

Test	Description	Release version
CSM - Create Proactive Case by NOC Operator	Verify whether a proactive case is created. **Note:** Requires Proactive Customer Service Operations with Event Management.	New York
CSM - Employee creating case OBO customer	As an employee with the proxy contact role (sn_customerservice_proxy_contact), create a case from the self-service portal on behalf of a customer. **Note:** Requires the Customer Proxy Contacts.	New York
CSM - Lookup Type Skill Determination Rule Test	Create a lookup type skill determination rule. **Note:** Requires Skill Determination.	New York
CSM - Manager creating request OBO customer from CSM portal	As a user with the case manager role, create a case on behalf of a customer from the Customer Service Portal. **Note:** Requires Customer Service Request Integration.	New York
CSM - Simple Type Skill Determination Rule Test	Create a simple type skill determination rule. **Note:** Requires Skill Determination.	New York
CSM - Project Manager create project task and assign to a customer	Project manager creates project task and assigns to a customer. **Note:** Requires Customer Project Management.	Orlando
CSM - Agent Create Cases from a Project	As a customer service agent, create a case from a project.	Orlando

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

CSM: Case Management test suite (continued)

Test	Description	Release version
	Note: Requires Customer Project Management.	
CSM - Agent Create Cases from a Project task	As a customer service agent, create a case from a project task. **Note:** Requires Customer Project Management.	Orlando
CSM - Agent Create Change Requests for a project	As a customer service agent, create a change request for a project.	Orlando
CSM - Service Contracts covered under Sold Product	Create sold products and service contracts and associate service contracts to a sold product. Verify the association between the active contracts and the sold product. **Note:** Requires Customer Service Install Base Management.	Orlando
AWA - Create New Service Channel	Create a new service channel in the Advanced Work Assignment application. **Note:** Requires Advanced Work Assignment for CSM.	Orlando
CSM-ITOM - Create Child Cases for Proactive Major Case	Create a child case for a proactive major case using recipient list. **Note:** Requires Proactive Customer Service Operations with Event Management.	Orlando
CSM - Create Outage from Case	Validate if newly created outage is linked to a case.	Paris
CSM: Create Sold Product on Household	Create a sold product on a household and its member.	Quebec

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

CSM: Case Management test suite (continued)

Test	Description	Release version
	Note: Requires Customer Service Install Base Management and Customer Household Data Model with **Load demo data** enabled.	
CSM: View Health Status of Install Base Item from Account and Case pages	Validates the functionality of the **Refresh Install Base Health** button on the Account and Case record pages.	Quebec
CSM: Report a knowledge gap from a case in Agent Workspace	Verify that a knowledge gap related to a case was reported in Agent Workspace	Rome
CSM-Create a case using 'Create a case (POST)' API	Create a case using 'Create a case (POST)' API. **Note:** Requires a user authentication record assigned to the **Basic authentication** field in step 1 of the ATF.	Rome
CSM-Query a case using 'Query a case (GET)' API	Query a case using 'Query a case (GET)' API. **Note:** Requires a user authentication record assigned to the **Basic authentication** field in step 1 of the ATF.	Rome
CSM: OCS Manager creating New OSP	Verify that a new Outsourced Service Provider (OSP) is created by the OCS internal manager. **Note:** Requires Outsourced Customer Service.	Rome
CSM: Case Creation by OCS Agent	Verify that an Outsourced Customer Service agent is able to create a case. **Note:** Requires Outsourced Customer Service.	Rome

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

CSM: Case Management test suite (continued)

Test	Description	Release version
CSM: Manage Consumer Profile Locations	Create and manage consumer locations, and map them to consumer profiles.	Washington DC
CSM: Address sharing across Accounts	Implement reusable addresses for accounts to support complex customer operations and business models.	Washington DC
CSM: Create Case as Location Contributor from BLSP Portal	As a location contributor,create a case from the BLSP portal.	Yokohama
CSM: Create a Business Location And Onboard a Member To It	Verifies the creation of a business location and onboarding a member to it.	Yokohama
CSM: Escalate or de-escalate account workflow	Escalate or de-escalate account workflow	Zurich

CSM Configurable Workspace test suite

Test	Description	Release version
CSM Workspace: Case creation Workflow	Tests the creation of a case record by a customer service manager in CSM/FSM Configurable Workspace.	Xanadu
CSM Workspace: Case follow up workflow	Tests the requesting of information from a customer by a customer service manager in CSM/FSM Configurable Workspace.	Xanadu
CSM Workspace: Report knowledge gap workflow	Tests the creation of a knowledge feedback task for a case record by a customer service manager in CSM/FSM Configurable Workspace.	Xanadu
CSM Workspace: Case resolution workflow	Tests the resolution of a case record by a customer service manager in CSM/FSM Configurable Workspace.	Xanadu
CSM Workspace: Create Incident from Case	Tests the creation of an incident record from a case record by a customer service agent in CSM/FSM Configurable Workspace.	Xanadu
CSM Workspace: Create Problem from Case	Tests the creation of a problem record from a case record by a customer	Xanadu

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

CSM Configurable Workspace test suite (continued)

Test	Description	Release version
	service agent in CSM/FSM Configurable Workspace.	
CSM Workspace: Create Normal Change from Case	Tests the creation of a change record from a case record by a customer service agent in CSM/FSM Configurable Workspace.	Xanadu
CSM Workspace: Case creation from All Cases list	Tests the creation of a case from the All cases list by a customer service manager (sn_customerservice_manager) in CSM/FSM Configurable Workspace.	Yokohama
CSM Workspace: Case Creation from My Cases list	Tests the creation of a case from the My Cases list by a customer service manager (sn_customerservice_manager) in CSM/FSM Configurable Workspace.	Yokohama
CSM Workspace: Case Creation from My Open cases list	Tests the creation of a case from the My Open list by a customer service manager (sn_customerservice_manager) in CSM/FSM Configurable Workspace.	Yokohama
CSM Workspace: Interaction creation from My Interaction list	Tests the creation of an interaction record from the My Interactions list by a customer service manager (sn_customerservice_manager) in CSM/FSM Configurable Workspace.	Yokohama

CSM: Case Types - Complaint

Test	Description	Release version
Create a Complaint Case	Tests that a user can create a case that is of type complaint in the sn_complaint_case table.	Paris
Escalate a Complaint Case	Tests that a user can escalate a case that is of type complaint in the sn_complaint_case table.	Paris

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

CSM: Case Types - Onboarding

Test	Description	Release version
Create an Onboarding Case	Tests that a user can create a case that is of type onboarding in the sn_onboarding_case table.	Paris
Escalate an Onboarding Case	Tests that a user can escalate a case that is of type onboarding in the sn_onboarding_case table.	Paris

CSM: Operations Dashboard test suite

Test	Description	Release version
awa_admin_operations_dashboard	Verify whether user with role awa_admin is able to view the Advanced Work Assignment menu under Operations Dashboard and unassigned interactions and unassigned task work items modules.	Orlando
awa_manager_operations_dashboard	Verify whether user with role awa_manager is able to view the Advanced Work Assignment menu under Operations Dashboard and unassigned interactions and unassigned task work items modules.	Orlando

CSM Portal test suite

Test	Description	Release version
CSM - Create Product Case as Customer from CSM Portal	Create a product case as a customer from the Customer Service Portal. **Note:** Requires Customer Service Portal. Also requires that the test be run as admin.	Madrid
CSM - Create Product Case as Partner from CSM Portal	Create a product case as a partner from the Customer Service Portal. **Note:** Requires Customer Service Portal. Also requires that the test be run as admin.	Madrid

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

CSM Portal test suite (continued)

Test	Description	Release version
CSM - Search on Homepage	Search for information from the Customer Service Portal. The search includes cases, Knowledge articles, and Community threads.	New York
CSM - Update Support Profile	Update a contact's profile from the Customer Service Portal.	New York
CSM - Provide requested info on case	From the Customer Service Portal, the contact can provide information for a case that was requested by the agent.	New York
CSM - Accept Proposed Solution On Case	Accept a proposed solution for a case from the Customer Service Portal.	New York
CSM - Provide Feedback on Survey	Provide feedback on a survey after a case is closed from the Customer Service Portal.	New York
CSM - View All Desktop Notifications	View all Customer Service Management specific desktop notifications.	Orlando
CSM - View Publications on CSM Portal	View publications on the Customer Service Management portal.	Orlando
CSM - Create Contact on CSM Portal	Create contacts on the Customer Service Management portal.	Orlando
CSM: Validate Outage widgets in CSM Portal	Validates various types of outages and the corresponding widgets shown on the Customer Service Portal home page and the Install Base page.	Quebec
CSM - Create case from Portal as Consumer Contributor	Creates a customer case from the portal by a user with the consumer contributor user role (sn_customerservice.consumer_contributor). **Note:** Requires Business Location, Customer Household Data Model, and CSM Contributor User.	Rome
CSM - Create case from Portal as Account Contributor	Creates a customer case from the portal by a user with the	Rome

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

CSM Portal test suite (continued)

Test	Description	Release version
	account contributor user role (sn_customerservice.account_contributor). **Note:** Requires Business Location, Customer Household Data Model, and CSM Contributor User.	
CSM - Add related parties on Case and perform action on case as Related party	Verify the ability to add related parties to the case and perform actions on the case as related parties through the CSM Portal.	Rome
CSM - Create product case as an account consumer from the CSM Portal	Create a product case as an account from the customer service portal. **Note:** Requires Customer Service Portal. Also requires that the test be run as admin.	Australia

CSP Portal test suite

Test	Description	Release version
CSP - Create product case as consumer from CSP Portal	Create a product case as a consumer from the Consumer Service Portal. **Note:** Requires the Consumer Service Portal. Also requires that the test be run as admin.	Madrid
CSP - Search on homepage	Search for information from the Consumer Service Portal. The search includes cases, knowledge articles, and Community threads.	New York
CSP - Update support profile	Update a consumer's profile from the Consumer Service Portal.	New York
CSP - Register your product	Register a product from the Consumer Service Portal.	New York
CSP - Provide requested info on case	From the Consumer Service Portal, the consumer can provide information for a case	New York

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

CSP Portal test suite (continued)

Test	Description	Release version
	that was requested by the agent.	
CSP - Accept proposed solution on case	Accept a proposed solution for a case from the Consumer Service Portal.	New York
CSP - Provide feedback on Survey	Provide feedback on a survey after a case is closed from the Consumer Service Portal.	New York
CSP - View publications on CSP Portal	View publications on Customer Service portal.	Orlando
CSP - Create product case as an internal unified consumer from CSP Portal	Verifies creating a product case as an internal unified consumer from the CSP portal. Requires the Customer Service and Consumer Service Portal plugins with the Customer Service Management (CSM) demo data plugin.	Yokohama
CSP - Create product case as an external unified consumer from CSP Portal	Verifies creating a product case as an external unified consumer from the CSP portal. Requires the Customer Service and Consumer Service Portal plugins with the Customer Service Management (CSM) demo data plugin.	Yokohama
CSP - Create household member as an account consumer from the CSM Portal	Create a household member as an account from the customer service portal. **Note:** Requires Customer Service Portal. Also requires that the test be run as admin.	Australia

Guided Decision - Next Best Action ATF test suite

Test	Description	Release version
Validate Next Best Action List- Guided Decisions	Validates the list of next best actions recommended based on the configured rules.	San Diego
Validate Next Best Action Ranking- Guided Decisions	Validates the list of next best actions recommended based on the configured rules.	San Diego

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

TC: Targeted Communications test suite

Test	Description	Release version
TC - Create Recipient List	Create a recipient list with the required parameters. Verify the new recipient list in the related list "Recipients".	Orlando
TC - Create Publication	Create a publication. The publication is published based on the publication date and verify if an user in the recipient list gets the publication.	Orlando

CSM Walkup Experience tests

Test	Description	Release version
CSM Walkup: Check-in as a walkup user, look at the queue, and submit a survey	As a walk-up user, perform check-in, look at the queue, and submit feedback through a survey. **Note:** This test works when the Seattle location, available with demo data, is available.	Rome

CSM: Customer Project Management tests

Test	Description	Release version
CSM PPM: Case creation from project workflow	As a customer service agent, create a case from a CSM project record in CRM Workspace.	Yokohama
CSM PPM: Case creation from project task workflow	As a customer service agent, create a case from a CSM project task record in CRM Workspace.	Yokohama

To learn more about Customer Service Management, see Customer Service Management.

Dashboards

Dashboards quick start tests require activating the Automated Test Framework - Responsive Dashboards plugin (com.glide.automated_testing_impl.dashboards). This plugin is active on zBoot of the instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Dashboards quick start tests

Test	Description	Release version
Responsive Dashboard Sharing	Confirm dashboard sharing by impersonating users.	Madrid
Responsive Dashboard Visibility	Confirm dashboard visibility by impersonating users.	Madrid

To learn more about Dashboards, see Dashboards.

DevOps

DevOps quick start tests are available when you install the DevOps application from ServiceNow Store.

DevOps test suite

Test	Description	Release version
DevOps Code Tool Flow	Verify the Workflow Studio flow for a DevOps coding tool.	Madrid
DevOps Orchestration Flow with CR	Verify the Workflow Studio flow for a DevOps orchestration tool that includes a change request.	Madrid
DevOps Orchestration Tool Flow	Verify the Workflow Studio flow for a DevOps orchestration tool.	Madrid
DevOps Plan Tool Flow	Verify the Workflow Studio flow for a DevOps planning tool.	Madrid

To learn more about DevOps, see DevOps Change Velocity.

Employee Center

Employee Center quick start tests

Test	Description	Release version
Create Adhoc Delegation for Approval Task	Ensures an adhoc delegation for an approval task can be created from the My Tasks form in the Employee Center.	San Diego
Create Adhoc Delegation for HR Task	Ensures an adhoc delegation for an HR task can be created from the My Tasks form in the Employee Center.	San Diego
Employee Center - My Favorites	Verifies the favorite functionality in the Employee Center. Also verifies when a	San Diego

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Employee Center quick start tests (continued)

Test	Description	Release version
	KB article or catalog item is added to the Favorite widget.	
Employee Center - Approval Hub Approve Request	Verifies the approval functionality in the Employee Center. Also verifies that when a request is approved, it appears in the user's **Completed** column.	San Diego
Employee Center - Approval Hub Reject Request	Verifies the rejection functionality. Also verifies that when a request is rejected, it appears in the user's **Completed** column.	San Diego
Employee center - Employee Profile Generation	Verifies that employee profiles are generated as per the employee definition.	Xanadu
Employee Center - Topic Page	Associates a KB article or catalog item to a topic and verifies they display on the topic page.	San Diego
Employee Center - Validate home page widgets	Validates the following widgets are present on the Employee Center home page: • Relevant for you • Popular topics • My Active Items • Quick tasks • Homepage Search	San Diego
ESC: Employee can see ticket updates	Employees can view updates to their tickets.	Quebec
ESC: Post General HR Inquiry questions on ESC portal	Verifies an HR employee can create a General Inquiry case and post general HR inquiry questions on the ESC.	Quebec
ESC: Submit a Record Producer which creates Universal Request and HR Case	Verifies a user can submit an HR catalog item that creates a Universal Request. Also verifies the Universal Request and HR case are created and linked.	Quebec
ESC: Verify Standard Ticket page on ESC for HR Case	Creates a general inquiry case from the service portal and verifies it appears on the standard Ticket page.	Quebec

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Employee Center quick start tests (continued)

Test	Description	Release version
ESC: Verify widget contents in Catalog items	Verifies the widget content in a Catalog page.	Orlando
ESC: Verify widget contents in knowledge pages	Verifies the widget content in a Knowledge page.	Orlando
HR: Search catalog items & KBs in ESC	Verifies the search functionality in the ESC.	Quebec
Taxonomy and Topic Creation	Creates taxonomy and topics and then adds content to the topic.	San Diego

Essential SAFe

Essential SAFe quick start tests require activating the Agile - Scaled Agile Framework - Essential SAFe plugin (com.snc.sdlc.safe) and the Agile - Scaled Agile Framework - Essential SAFe - ATF Tests plugin (com.snc.sdlc.safe.atf).

Essential SAFe test suites

Test	Description	Release version
Essential SAFe: Feature tests	Verify feature global rank updates.	Madrid
Essential SAFe: Feature tests	For a SAFe feature, verify that: • Actual start date is populated after the state is changed to Implementation, Validation on Staging, or Deployment. • Actual end date is populated after the state is changed to **Released** or **Cancelled**. • Active flag is set to the appropriate value: ○ **False**, if the state is changed to **Released** or **Cancelled**. ○ **True**, for all other states.	Orlando
Essential SAFe: Feature tests	If the PPM Standard plugin (com.snc.financial_planning_pmo) is active, verify that a Demand can be converted to a SAFe feature using the **Create SAFe Feature** related link on the Demand form.	Quebec

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Essential SAFe test suites (continued)

Test	Description	Release version
	For more information on how to create an SAFe feature from a demand, see Create an artifact from a demand.	
Essential SAFe: Program increment tests	Verify program increment date overlapping.	Madrid
Essential SAFe: Sprint tests	Verify the generation of ART sprints and team sprints as well as updates to sprint points and dates.	Madrid
Essential SAFe: Sprint tests	Verify that any update to the **Group capacity** field of the assignment group results in the following changes to the **Group capacity** field of the various sprints associated with this assignment group: • For the sprints that are in the **Draft** or **Planning** states: ○ The group capacity is updated to the new value. ○ The **Group capacity** field is editable. • For the sprints in the **Current**, **Complete**, or **Cancelled** states: ○ The group capacity remains the old value. ○ The **Group capacity** field is read-only. For the sprints in the **Draft** or **Planning** state, you can individually edit the group capacity of the sprint anytime later. This would not change the group capacity of the assignment group associated with this sprint.	Paris
Essential SAFe: Story tests	Verify story global rank updates.	Madrid

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Essential SAFe test suites (continued)

Test	Description	Release version
Essential SAFe: Story tests	Verify that active flag of the SAFe story is set to the appropriate value: • **False**, if the state of the state is changed to Completed or Cancelled. • True, for all other states.	Orlando
Essential SAFe: Story tests	Verify that adding, estimating, removing, deleting, updating, or cancelling a SAFe story updates the SAFe feature-level and then the epic-level roll-ups correctly.	Orlando
Essential SAFe: Story tests	Verify that adding, updating, or deleting the feature on a SAFe story updates the Epic field on the SAFe story form.	Orlando
Essential SAFe: Story tests	If the PPM Standard plugin (com.snc.financial_planning_pmo) is active, verify that a Demand can be converted to a SAFe story using the **Create SAFe Story** related link on the Demand form. For more information on how to create a SAFe story from a demand, see Create an artifact from a demand.	Quebec
Essential SAFe: Team tests	Verify team association with an ART.	Madrid
Essential SAFe: Epic tests	For a SAFe epic, verify that: • Actual start date is populated after the state is changed to **Implementation**. • Actual end date is populated after the state is changed to **Complete**. • Active flag is set to the appropriate value:	Orlando

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Essential SAFe test suites (continued)

Test	Description	Release version
	○ **False**, if the state is changed to **Released** or **Cancelled** states. ○ **True**, for all other states.	
Essential SAFe: Epic tests	If the PPM Standard plugin (com.snc.financial_planning_pmo) is active, verify that a Demand can be converted to a SAFe epic using the **Create SAFe Epic** related link on the Demand form. For more information on how to create a SAFe epic agile from a demand, see Create an artifact from a demand.	Quebec
Essential SAFe: Program PI Objective tests	Verify the functionality of creating and updating PI objectives • Verify that you can create a program increment-level objective • Verify that you can update the created PI objective with the planned business value (PBV) and actual business value (ABV) • Verify that the percentage of business value achieved is computed as the percentage of ABV complete for committed PI objectives compared to program PBV	Rome
Essential SAFe: Team PI Objective tests	Verify the functionality of creating and updating team PI Objectives • Verify that you can create a team-level PI objective • Verify that you can update the created PI objective with the planned business value (PBV) and actual business value (ABV)	Rome

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Essential SAFe test suites (continued)

Test	Description	Release version
	• Verify that the percentage of business value achieved is computed as the percentage of ABV complete for committed PI objectives compared to team PBV	

To learn more about Essential SAFe, see Essential SAFe.

Event Management

Event Management quick start tests require activating the Event Management plugin (com.glideapp.itom.snac).

Event Management quick start tests

Test	Description	Release version
EMSelfMonitoring	Track problems with the Event Management plugin after upgrade. To understand the details of issues found, open the All Alerts list. Search for alerts that failed, whose source is `EMSelfMonitoring`, and that were opened or reopened since the upgrade. The exact time for filtering can be found in the error message.	Madrid

To learn more about Event Management, see Event Management.

Field Service Management

Field Service Management quick start tests are available when you enable the Field Service Management plugin (com.snc.work_management). Enable the demo data plugin (com.snc.work_management.demo) in a non-production instance to start using the quick start tests available with your application. Activate Field Service Contractor Management plugin (com.snc.fsm_contractor_management) to execute the External contractor related tests. You can also modify existing data and customize it to run the quick tests.

FSM: Field Service Management test suite

Test	Description	Release version
FSM: Create Initiate Qualify Dispatch and assign Work Order Task	• Create a work order. • Initiate and qualify a work order.	Madrid

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

FSM: Field Service Management test suite (continued)

Test	Description	Release version
	• Dispatch a work order. • Assign a work order task to an agent.	
FSM: Part Sourcing	• Source any part to an agent's stock room from the work order task. • Create a transfer order line for the part sourcing. • Use this part for any work order to consume it using the **Part Usage** action.	Madrid
FSM: Part Usage	• Source any part to an agent's stock room from the work order task. • Use this part for any work order to consume it using the Part Usage action.	Madrid
FSM: Questionnaire	Create a questionnaire and associate it with a work order task.	Madrid
FSM: Field Service Configuration	Verify that the default configuration such as task assignment method, qualification requirement, PDF summary and agent's ability to accept or reject tasks is preserved.	New York
FSM: Planned Maintenance	Create a planned maintenance work order with weekly interval time for printer maintenance.	New York
FSM: Appointment Booking Configuration	Verify that the default configuration for appointment booking is preserved and that the point of sale service and catalog item exists in the system.	New York
FSM: Create Work or Personal Schedules	Create personal or work schedule for agents.	Orlando
FSM: Field Service Property Settings	Verify that the field service system properties preserve expected values.	Orlando

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

FSM: Field Service Management test suite (continued)

Test	Description	Release version
FSM: Work Groups	Verify that field service agents can be added to work groups.	Orlando
FSM: Dynamic Scheduling - Preferred Technician assignment with mandatory parts reservation	With Dynamic scheduling: • Preferred technician should be picked for task assignment. • Mandatory parts should be reserved in the agent stock room. **Note:** Activate Customer service management demo data plugin.	Paris
FSM: Create Time Card	Verify that a time card is created for an agent in the work order task.	Quebec
FSM: Create incidental	Verify that an incidental is created for an agent in the work order task.	Quebec
FSM: Onboarding contractor company	Verify that a contractor company is onboarded with assignment group, external manager, and external agent. **Note:** Activate Field Service Contractor Management plugin.	Quebec
FSM: Onboarding contractor agent by external manager	Verify that an external contractor manager can onboard external agents from the Field Service Contractor Management portal.	Quebec
FSM: External contractor manager fulfil the work order task	Verify that a contractor manager or an agent is able to fulfil the assigned work order task.	Quebec
FSM: Assign work order task to Vendor group	Verify that a work order task is assigned to the external assignment group based on the defined criteria, such as task location and configuration parameters.	Quebec

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

FSM: Field Service Management test suite (continued)

Test	Description	Release version
FSM: Pause and Resume work order task	Verify that an agent can pause and resume work for a work order tasks.	Rome
FSM: Dynamic Scheduling - Assign technician with matching skills	With Dynamic Scheduling: Verify that a work order task is assigned to a technician who possesses all mandatory skills mentioned in the task.	Rome
FSM: Off boarding contractor agent by external manager	Verify that an external contractor manager can off board external agents from the Field Service Contractor Management portal.	Rome
FSM: External contractor manager asset sourcing and usage	Verify that an external contractor manager is able to perform the following actions: • Request parts from stockroom. • Pick the part. • Close the transfer order. • Perform work using the part.	San Diego
FSM: External contractor agent fulfil the work order task	Verify that a contractor agent is able to fulfil the work order task that is assigned by the external contractor manager.	San Diego
FSM: Capacity Scheduling - Assign work to field service agents based on capacity	Verify that the rules and values mentioned in the capacity definition and capacity assignment module are evaluated to assign work to the agent based on capacity.	Tokyo
FSM: Onboarding external contractor agent as additional manager for the external assignment group	Verify that when onboarding an external agent for the contractor company, you can assign the additional manager role to the agent.	Tokyo
FSM: Dispatcher Workspace - Assign crews to work order tasks that require a group of agents to work on them	• Verify that a dispatcher with the crew moderator role can create a crew in the Dispatcher Workspace. • Verify that a dispatcher with the crew moderator role can	Tokyo

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

FSM: Field Service Management test suite (continued)

Test	Description	Release version
	assign a work order task to the planned crew in the Dispatcher Workspace	
FSM: Work order task supports Multi-day task scheduling	Verify the Multiday task scheduling functionality: • Create Work order, work order task, and agent schedule records. • Select the Assign across schedule entries option in the work order task. • Assign an agent to the work order task for which the task duration is more than a day. • Verify that work order task is assigned successfully to the agent and the estimated end time is populated correctly.	Tokyo
Verify creating a work order from a case and assigning a work order task to an agent from CSM/ FSM Configurable Workspace	• Verify you can create a work order from a case. • Qualify the work order and verify a work order task is created. • Assign the work order task to an agent and verify the agent it is assigned to is updated correctly.	Xanadu
FSM: Appointment Booking unified UI modal validation in platform	• Create work order and work order task. • Launch Seismic Appointment Booking calender in Platform. • Verify that **Today** and **Next week** options displays the correct calender.	Yokohama
Create and qualify a work order, then assign the work order from Dispatcher Workspace as a dispatcher	• As a dispatcher, from the CSM/FSM Configurable workspace go to the **List** view and create a new work order.	Yokohama

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

FSM: Field Service Management test suite (continued)

Test	Description	Release version
	• Then as an administrator, qualify the work order and verify if a work order task is created. • Then as a dispatcher, assign the work order task to a Field Service agent and verify the Assigned to field is update correctly.	
Clone a work order task	• From from the CSM/FSM Configurable workspace, open any work order task record page. • Select **Clone Work order task**, and verify that a new work order task is created with all details populated from the original work order task.	Yokohama
Create a time worked entry as a dispatcher or qualifier	• From the CSM/FSM Configurable workspace, go to the List view and create a new work order. • Qualify the work order and verify that a work order task is created. • Select **Record Time**, then fill in the time worked entry and select **Save**. • Verify that the time worked entry is created successfully.	Brazil
Flag a work order task from CSM/FSM Configurable workspace	• From the CSM/FSM Configurable workspace, as a user with the Dispatcher or Qualifier role, go to the List view and create a new work order. • Qualify the Work order and verify if a Work order task is created. • Go to the Work order task you created and select **Flag**.	Brazil

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

FSM: Field Service Management test suite (continued)

Test	Description	Release version
	Verify the work order task is flagged and you see a **Unflag** button. • Select **Unflag** and verify work order task is now unflagged, and the **Flag** button is available.	

Finance Close Automation

Finance Close Automation quick start tests are available when you install the Finance Close Automation application from ServiceNow Store.

FCA test suite

Test	Description	Release version
FCA: Applicable Months sub-set tests	Verify that the Applicable months can only be sub-set of functional workbook.	New York
FCA: dates validations	Verify that the date related calculations are correct.	New York
FCA: task relationships	Verify that dates and states are updated correctly for the tasks having relationships.	New York
FCA: relationship with parent tasks	Verify that dates and states are updated correctly for tasks that have relationship with parent tasks.	New York
FCA: Reset Task	Verify the FCA workflow when the task is reset.	New York
FCA: Reject Task	Verify the FCA workflow when the task is rejected.	New York
FCA: Create JE Task	Verify that ERP source is available for a Journal Entry task type.	New York
FCA: Negative values as close day	Verify that both the positive and negative values are acceptable for close day and the planned dates are calculated correctly.	New York
FCA: Close day checks for Daily close Checklist	Close day verifications in daily close workbook.	New York
FCA: Daily close planned end date checks	Verify planned end dates in a daily close workbook.	New York

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

FCA test suite (continued)

Test	Description	Release version
FCA: Kickstart errors	Verify that all the kickstart validations run.	New York
FCA: One final task verification	Verify that there is only one final task in each functional workbook.	New York
FCA: Confidential task	Verify that only the owner, reviewer, and approver of a confidential task can view the task.	New York
FCA: Unique checklist Verification	Verify that only one workbook is created for a specific day or period.	New York
FCA: Milestone and JE Validations	Verify that milestone and journal entry tasks don't have any child tasks.	New York

Financial Management

Financial Management quick start tests require activating the Financial Management Core - ATF Tests plugin (com.snc.financial_management.atf)

ITFM: Financial Modeling flow test suite

Test	Description	Release version
Verify FM Cost Allocation Flow	Verify the cost allocation flow in financial modeling.	Madrid

Granular Delegation

Granular Delegation quick start tests require activating the Granular Delegation (com.glide.granular_service_delegation) plugin.

Test	Description	Release version
[Delegation] Admin - Adding a delegation rule with delegator and delegate user criteria	Ensures a delegate or delegator honors the user criteria assigned to an HR task.	Brazil

GRC Audit Management

GRC: Audit Management quick start tests require activating the GRC: Audit Management plugin (com.sn_audit) and loading the demo data.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

GRC: Audit Management test suite

Test	Description	Release version
GRC: Create Audit Engagement and Generate Audit Task	Validates audit engagement creation and associates entities to generate controls and test plans. Generates audit task which is associated to a test plan.	Paris
GRC: Create and process a milestone	Create a milestone in an engagement. The due date cannot be in the past and the completion date cannot be in the future. For a milestone in open state, the percent complete is 0 and changes with the milestone state.	Paris
GRC: Cost and Resource plan rollup	Create an audit plan and associate an engagement to it. Add a cost plan and resource plan to this engagement. Notice that these costs are rolled up to the plan. Any edits to these costs in engagement reflects in the plan.	Paris
GRC: Create Engagement Project Manual and automatic	On an engagement in the validate state, perform the Enable advanced planning ui-action. An engagement project gets created. When the state of an engagement associated to an audit plan having "Advanced planning capabilities" is changed to validate, an engagement project gets created automatically.	Paris
GRC: Auditable Unit with Detailed Risk Assessment	Create an Auditable unit with method as "Detailed Risk Assessment" and request for Assessing the Risk Assessment by adding the Assessor. Once the assessor responds and marks Assessment as Complete after performing the control assessment and residual assessment, the	Paris

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

GRC: Audit Management test suite (continued)

Test	Description	Release version
	risk assessment fields are automatically updated.	

To learn more about Audit Management, see Audit Management.

GRC Continuous Authorization and Monitoring

GRC: Continuous Authorization and Monitoring quick start tests require activating the Continuous Authorization and Monitoring plugin (com.sn_compliance) and loading the demo data.

GRC: Continuous Authorization and Monitoring Quick Start Tests test suite

Test	Description	Release version
GRC: System Owner create and validate responsibilities and roles for an AB and AP	System Owner creates and validates responsibilities and roles for an Authorization Boundary and Authorization Package. Information Owners and System User are pre-populated when selecting the Authorization Boundary.	Quebec (compatible with Paris and Orlando)
GRC: System Owner validate App Modules visibility	Verifies that the system owner persona is able to view the Continuous Authorization & Monitoring application menu and the following modules under that menu: • All Authorization Boundaries • All Authorization Packages • Information Type Library • Control Overlays • Control Objectives • Controls • All Engagements	Quebec (compatible with Paris and Orlando)
GRC: System Owner Request First approval & My approvals module	System Owner requests an approval.	Quebec (compatible with Paris and Orlando)
SO: Create and validate responsibilities and roles for an AB and AP	Verifies if a system owner can create an Authorization Boundary by completing the	Quebec (compatible with Paris and Orlando)

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

GRC: Continuous Authorization and Monitoring Quick Start Tests test suite (continued)

Test	Description	Release version
	fields on the Authorization Boundary form. Also verify if the same SO can create an Authorization Package from the form view.	

To learn more about Continuous Authorization and Monitoring, see Continuous Authorization and Monitoring.

GRC Policy and Compliance Management

GRC: Policy and Compliance Management quick start tests require activating the Policy and Compliance Management plugin (com.sn_compliance) and loading the demo data.

GRC: Policy and Compliance Management Quick Start Tests test suite

Test	Description	Release version
GRC: Create Controls	Validates control creation.	Madrid
GRC: Create Policy Exception and Flow	Create a policy exception and navigate through its states.	Paris
GRC: Policy Lifecycle	Create a policy and navigate through its states.	Paris

To learn more about Policy and Compliance Management, see Policy and Compliance Management.

GRC Risk Management

GRC: Risk Management quick start tests require activating the Risk Management plugin (com.sn_risk) and loading demo data.

GRC: Risk Management Quick Start Tests test suite

Test	Description	Release version
GRC: Create Profile	Validate profile creation.	Madrid
GRC: Create Risk	Validates risk creation.	Madrid
GRC: Create Control	Validates control creation.	New York
GRC: Create Control Objective and Generate Controls	Validates control objective creation and associates profiles to generate controls.	New York
GRC: Create Issue	Validates issue creation	New York
GRC: Accept an Issue	Validates all the states of an issue till it is closed by accepting the issue.	New York

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

GRC: Risk Management Quick Start Tests test suite (continued)

Test	Description	Release version
GRC: Remediate an Issue	Validates issue cannot be closed be with an open remediation task.	New York
GRC: ATF Flow for Indicator (Manual):	Create an indicator template with type manual and associate a control objective to it. Execute one of the indicators thus formed. Mark the state of the indicator task created as closed and result as failed. The associated control will become non-compliant and an issue will be generated.	New York
GRC: ATF flow for Indicator (Basic)	Create an indicator template with type basic and result as failed. Associate a control objective to it and give the supporting data. Execute one of the indicators thus formed. The associated control will become non- compliant and an issue will be generated.	New York
GRC: ATF flow for indicator (Script)	Create an indicator template with type script and enter a script and set the value of result.passed and result.value. Associate a control objective to it. Execute one of the indicators. The associated control will become non-compliant and an issue will be generated.	New York

To learn more about Risk Management, see Risk Management.

Hardware Asset Management

Hardware Asset Management quick start tests are available when you install the Hardware Asset Management (HAM) application from the ServiceNow Store.

Hardware Asset Management test suite

Test	Description	Release version
HAM - Hardware Normalization	Validates the various normalization status values based on the normalized Manufacturer, Product, and Model.	Orlando

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Hardware Asset Management test suite (continued)

Test	Description	Release version
HAM - Hardware Asset Disposal work flow	Validates asset disposal work flow.	Paris
HAM - Automating Asset deploy workflow	Validates asset deployment work flow.	Paris
HAM - Automating Asset Swap workflow	Validates Asset Exchange/ Swap work flow. **Note:** Requires demo data.	Paris
HAM - Standard Hardware Asset Request Flow	Validates the Standard Hardware Asset Request flow, which is a part of the Hardware Asset Management application.	Paris
HAM - Hardware Asset Refresh flow	Validates asset refresh workflow.	Quebec
HAM - Loaner Asset Allocation Flow	Validates loaner asset workflow.	Quebec
HAM - Lease Contract Asset Expiration	Validates leased contract asset expiration end to end workflow, performing return or extend of the assets covered in the lease contract.	Quebec
HAM - Asset RMA Flow	Validates the asset Return Merchandise Authorisation (RMA) workflow with Inventory User.	Rome
HAM - Asset RMA Flow (SP)	Validates the asset Return Merchandise Authorisation (RMA) workflow from Service Portal.	Rome
HAM - Loaner Asset Request Flow	Validates the Loaner asset allocation workflow with Inventory Admin user.	Rome
HAM - Contract Renewal Flow	Validates the Contract Renewal flow with Contract Manager user.	Tokyo

To learn more about Hardware Asset Management, see Hardware Asset Management.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

HR Service Delivery

HR Service Delivery case tests

Test	Description	Release version
Case Creation Configuration	Validates an HR service has a case creation configuration and is populated into the HR case form when creating an HR case for that HR service.	Yokohama
Create Document template block and block content	Verifies a document template block and document template content are created for block content.	Yokohama
Create HR Employee Document	Creates an employee document and ensures the Attachment [sys_attachment] table points back to the record in the Employee Document [sn_hr_ef_employee_document] table. Verifies that the employee document is accessible from the Employee Center or Employee Center Pro via the employee HR profile.	
Create Employee Document with Active Document Type	Validates the creation of: • One active document type • One inactive document type • Employee file with active document type **Note:** Human Resources Scoped App: Core (com.sn_hr_core) and Employee Document Management (com.sn_employee_document_management) must be activated.	Yokohama
[Delegation] Admin - Adding a delegation rule with delegator and delegate user criteria	Verifies that user criteria for a delegation rule works correctly,	San Diego
E2E test for Bulk Case Creation	Verifies that bulk cases can be created for selected users in a specific user segment.	San Diego
ER: Case Restriction	Verifies that restricted cases are only viewable by users	San Diego

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

HR Service Delivery case tests (continued)

Test	Description	Release version
	with the sn_hr_er.confidential role.	
HR: Case updates visible to employees	Cases created and updated by an HR agent are visible to the Opened for or Subject person	Quebec
HR: Creation of 401k enrollment case from portal	Creates a 401(k) enrollment case from the Employee Center or Employee Center Pro. Verifies that the case creates successfully.	Madrid
HR: Creation of Employee Relations case from ESC portal	Creates a Disciplinary Issue case from the Employee Center or Employee Center Pro. Verifies that the case creates successfully.	New York
HR: Creation of Employment Verification Letter case from ESC portal	Creates an Employment Verification Letter case from the Employee Center or Employee Center Pro. Verifies that the case creates successfully.	Madrid
HR: Creation of Payroll case from ESC portal	Creates a direct deposit payroll setup case from the Employee Center or Employee Center Pro. Verifies that the case creates successfully.	New York
HR: Creation of Tuition Reimbursement request from portal	Creates a Tuition Reimbursement Request case from the Employee Center or Employee Center Pro. Verifies that the case creates successfully.	Madrid
HR: Fulfillment Instructions	Verifies that fulfillment instructions are created for a specific condition for an opened for user.	Orlando

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

HR Service Delivery case tests (continued)

Test	Description	Release version
	Also verifies the fulfillment instructions change when the details of the case changes.	
HR: Fulfillment Instructions conditions for COE specific fields	Verifies the conditions for COE specific fulfillment instruction. Fulfillment instructions update on an HR case based on the conditions defined in the fulfillment instructions.	Quebec
HR: General Benefits Inquiry Case Creation	Creates a General Benefits case using the native UI. It also verifies that the case opens and can be updated after case creation.	Madrid
HR: Payroll Discrepancy Case Creation	Creates a Payroll Discrepancy case using the native UI. It also verifies that the case opens and can be updated after case creation.	Madrid
HR: Reclassify Case Transfer	Tests when an HR case transfers from one COE (HR service) to a different COE using the reclassify (HR case number remains the same) method.	Orlando
HR: Response Template Configuration	Creates a response template for a payroll case and pastes the response in the worknotes.	Quebec
HR: Search catalog items & KBs in ESC	Verifies the search functionality in the Employee Center or Employee Center Pro.	Orlando
HR: Standard Case Transfer	Tests an HR case transfers from one COE (HR service) to a different COE using the standard (creates HR case number and deletes old number) case transfer.	Orlando
HR: Suspend Case Flow	Tests the ability to suspend an HR case and later resume the case.	Rome

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

HR Service Delivery case tests (continued)

Test	Description	Release version
HR: Tuition Reimbursement Case Creation	Creates a Tuition Reimbursement HR case.	Orlando
HR: Workforce Administration Case Creation	Creates a Workforce Administration case, signs an employee verification letter, and generates the letter.	Orlando
JA: Publish & edit plan details	Verifies the ability to delete an existing to-do in an action plan.	San Diego
Journey Designer: Create stage and edit stage	Verifies that the user can create a new stage, add a task template to the stage, edit the stage name, and delete the stage.	Vancouver
Journey Designer: Add task templates from scratch and edit	Verifies that the user can add a task template using the Build from scratch option, then edit the task template.	Vancouver
Journey Designer: Add existing task templates and remove task template	Verifies that a task template can be added to a Journey template with the Use template option, and then removed.	Vancouver
Learning Core Task Assignment	Verifies a learning task from a learning catalog can be assigned to a user.	San Diego

Note: Requires plugin activation of:

- Human Resources Scoped App: Core plugin (com.sn_hr_core)
- Employee Service Center [com.sn_hr_service_portal]

Lifecycle Events: Tests for verifying Lifecycle Events

Test	Description	Release version
Configure HR Service for Auto Case Closure	Verifies that Lifecycle event cases are automatically closed.	Orlando
Create Employee request and associate it to LE case	Creates and validates that an employee request appears in the Lifecycle Event case action list.	Rome
Create Employee request for old ticket page	Validates if employee requests can be created from the legacy ticket page.	Yokohama

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Lifecycle Events: Tests for verifying Lifecycle Events (continued)

Test	Description	Release version
[Delegation] Admin - Adding a delegation rule with delegator and delegate user criteria	Verifies user criteria is working for delegation rules for a delegator and delegate user.	Yokohama
HR Lifecycle: Access Ticket page for a NewHire Onboarding Case in Employee Service Center	Verifies creation of a new hire and can access the Ticket page of the Employee Service Center.	Orlando
HR Lifecycle: Access ESC ticket page and todos page and complete assigned todos	Creates a Tuition Reimbursement Request case from Employee Service Center and the Complete NDA to-do from the subject person.	Orlando
HR LifeCycle: Request Onboarding Case Creation	Creates a Request Onboarding case using the native UI and updates the case after creation.	Madrid
HR Lifecycle: Trigger Rescind Workflow	Triggers the Rescind workflow for a New Hire Onboarding Lifecycle Event case.	Quebec
HR Lifecycle: Verify Requests page for Open and Closed cases	Verifies open and closed cases on the Requests page.	Orlando
HR Lifecycle: Verify search in Requests Page	Verifies the search functionality on the Requests page for open and closed cases.	Orlando
[LE]: Add Ad hoc task	Verifies an ad hoc task can be added for a Lifecycle Event case.	Yokohama

Note: Requires plugin activation of:

- Human Resources Scoped App: Core plugin (com.sn_hr_core)
- Human Resources Scoped App: Lifecycle Events [com.sn_hr_lifecycle_events]
- Human Resources Scoped App: Lifecycle Events for Enterprise plugin [com.sn_hr_lifecycle_ent]
- Employee Service Center [com.sn_hr_service_portal]

KB: Knowledge block tests

Test	Description	Release version
Create knowledge block and attach block to article	Creates a knowledge block, publishes the block, and attaches the block to a knowledge article.	New York

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

KB: Knowledge block tests (continued)

Test	Description	Release version
Preview Knowledge Articles with Knowledge Blocks	Tests for preview of knowledge article that contains knowledge blocks.	New York

Note: Requires plugin activation of Human Resources Scoped App: Core plugin (com.sn_hr_core)

To learn more about HR Service Delivery, see HR Service Delivery.

Incident Management

Incident Management quick start tests require activating the Incident Management - ATF Tests plugin (com.snc.incident.atf).

Incident Management test suite

Test	Description	Release version
INCIDENT MGMT: Incident Resolution SLA	Test to verify the Incident Resolution SLA baseline functionality.	Madrid
INCIDENT MGMT: Incident Response SLA	Test to verify the Incident Response SLA baseline functionality.	Madrid
INCIDENT MGMT: Copy Incident	Test to verify whether the fields from the original Incident are copied correctly to the new Incident.	Madrid
INCIDENT MGMT: Copy Incident from a Closed Incident	Test to verify that the **Copy Incident** UI action is visible for closed Incidents.	Madrid
INCIDENT MGMT: Create Standard Change from Incident	Test to verify the creation of a Standard Change from an Incident.	Madrid
INCIDENT MGMT: Create of Emergency Change from an Incident	Test to verify the creation of an Emergency Change from an Incident.	Madrid
INCIDENT MGMT: Create Normal Change from an Incident	Test to verify the creation of a Normal Change from an Incident.	Madrid
INCIDENT MGMT: Create Problem from an Incident	Test to verify the creation of a Problem from an Incident.	Madrid
INCIDENT MGMT: Create Knowledge from an Incident	Test to verify the creation of a Knowledge from an Incident.	Madrid
INCIDENT MGMT: Incident State flow	Test to verify the state flow of an incident.	Madrid

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Incident Management test suite (continued)

Test	Description	Release version
INCIDENT MGMT: Reopening an Incident	Test to verify the reopen incident functionality.	Madrid
INCIDENT MGMT: Incident Assignment	Test to verify the incident assignment functionality.	Madrid
INCIDENT MGMT: Create child Incident using UI action and verify its fields	Test to verify the creation of a child Incident from an Incident through the **Create Child Incident** UI action. The test also verifies that the fields of the child Incident get copied correctly from the parent incident to the child incident.	Madrid
INCIDENT MGMT: Incident creation — Self Service	Test to verify the creation of an Incident using the **Create Incident** catalog item.	Madrid
INCIDENT MGMT: Parent and Child Incident state sync up	Test to verify that the state of a child Incident synchronizes with the parent Incident when the child Incident is created.	Madrid
INCIDENT MGMT: Parent and child Incident state sync up after reopening an Incident	Test to verify that the state of a child Incident synchronizes with the parent Incident when the parent Incident is reopened.	Madrid
INCIDENT MGMT: Verify creation of knowledge article from an Incident	Test to verify the creation of a knowledge article from an Incident using the **Create Knowledge** UI action on the Incident form. The UI action is visible when you activate the KCS Integration for Incident Management plugin (com.snc.incident.knowledge).	Orlando

To learn more about Incident Management, see Incident Management.

Incident Management in Service Operations Workspace

Incident Management in Service Operations Workspace quick start tests are available in the base system when you install or update to Service Operations Workspace ITSM Applications (sn-sow-itsm-cont) 6.0 version.

Incident Management in Service Operations Workspace test suite

Test	Description	Release version
SOW Incident: Create problem from incident	Test to verify the creation of a problem record from an Incident using the **Create**	Xanadu (August store release, 6.0 version)

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Incident Management in Service Operations Workspace test suite (continued)

Test	Description	Release version
	Problem UI action on the Incident form.	
SOW Incident: Verify Assign to me button functionality	Test to verify the assignment of the incident record to the logged-in user using the **Assign to me** UI option on the Record information side panel of the Incident form.	Xanadu (August store release, 6.0 version)

To learn more about Incident Management in Service Operations Workspace, see Incident Management in Service Operations Workspace.

Integration Commons for CMDB

CMDB INT: CMDB Integrations Validation test suite

Test suite to verify the integrity of an integration using multiple tests.

Test	Description	Release version
CMDB INT: Set Test Session Application	Modify the run server-side script to set an application name so that you can test only one integration. Otherwise, all integrations installed will be tested.	Paris
CMDB INT: Test Against Source Analysis	Test an integration against the values in the CMDB Integration Source Analysis [sn_cmdb_int_util_cmdb_integration_source_analysis] table.	Paris
CMDB INT: Validate Application Feed	Validate all application feeds in an integration.	Paris
CMDB INT: Validate Discovery Source	Validate that the discovery source exists.	Paris
CMDB INT: Validate Entity Mappings	Validate all entity mappings of an integration.	Paris
CMDB INT: Validate Fields	Validate fields for CMDB Integrations.	Paris
CMDB INT: Validate Lookups	Validate CMDB integration lookups.	Paris
CMDB INT: Validate Mandatory Operations	Validate that all integrations for mandatory operations exist for mapped fields.	Paris
CMDB INT: Validate Operations	Validate all operations for an integration.	Paris

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

CMDB INT: CMDB Integrations Validation test suite

Test suite to verify the integrity of an integration using multiple tests.

(continued)

Test	Description	Release version
CMDB INT: Validate References	Validate CMDB integration references.	Paris
CMDB INT: Validate Related Entries	Validate all related classes against the data dictionary for related entries.	Paris
CMDB INT: Validate Relationships	Validate CMDB integration relationships.	Paris

To learn more about Integration Commons for CMDB, see Integration Commons for CMDB.

Investment Funding

The Investment Funding quick start tests require activating the Investment Funding - ATF Tests plugin (com.snc.investment_planning.atf).

Investment Funding quick start tests

Test	Description	Release version
Validation of top-down funding and unfunding	• Validate that the correct amount is allocated from top investment to child investments while funding. • Validate that the correct amount is returned back to the top investment from the child investment while unfunding.	Orlando
Validation of bottom-up funding	• Validate that the correct amount is requested by the child investment from the parent investment. • Validate that the correct amount is allocated back to the child investment from the parent investment.	Paris

To learn more about Investment Funding, see Investment Funding.

Knowledge Management

Knowledge Management quick start tests require activating the Knowledge Management Core plugin (com.glideapp.knowledge), the Knowledge Management Advanced Installer plugin (com.snc.knowledge_advanced.installer), and the Knowledge Blocks plugin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

(com.snc.knowledge_blocks). The tests also require activating the Customer Service Management Demo Data plugin (com.snc.customerservice.demo).

KM: Knowledge Management test suite

Test	Description	Release version
KM : Create KCS Template Article with Approval Publish workflow	Verify the creation of a KCS template article with approval publish workflow. **Note:** Requires the Knowledge Management Advanced Installer plugin.	Madrid
KM: Create MultiVersioned standard Article with approval publish workflow	Verify the creation of a multi-versioned standard article with approval publish workflow. **Note:** Requires the Knowledge Management Advanced Installer plugin.	Madrid
KM: Article level subscription	Verify that users can subscribe to a knowledge article. **Note:** Requires the Knowledge Management Advanced Installer plugin.	New York
KM: User criteria covering canRead and canContribute for KnowledgeBase and canRead at Article level	Verify the creation of canRead and canContribute user criteria for knowledge base and canRead user criteria for article. **Note:** Requires the Knowledge Management Core plugin.	Madrid
KM: Create a KCS article from a case	Verify the creation of a KCS article from a case. **Note:** Requires the Knowledge Management Advanced Installer and Customer Service Management Demo Data plugins.	Madrid

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

KM: Knowledge Management test suite (continued)

Test	Description	Release version
KM: Knowledge Base Level Subscriptions	Verify that users can subscribe to a Knowledge Base. **Note:** Requires the Knowledge Management Advanced Installer plugin.	New York
KM: AQI	Verify the creation, assignment, and review of an AQI checklist. **Note:** Requires the Knowledge Management Advanced Installer plugin.	New York
KM: Requestor performs search, view the article, provide feedback (Helpful No with Feedback Task)	Verify the search request, review, provision of feedback as not helpful, and creation of a feedback task for an article on the Knowledge Management Service Portal. **Note:** Requires the Knowledge Management Advanced Installer plugin.	New York
KM: Create and edit an article from Feedback Task form	Confirm the availability of the **Create Article** and **Edit Article** buttons on a Knowledge Feedback Task form. **Note:** Requires the Knowledge Management Advanced Installer plugin.	New York
KM: Search for pinned articles and checking the highest click rank	Verify the search request and that the click rank value of a pinned article was added to the Knowledge Searches (ts_query_kb) table.	New York

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

KM: Knowledge Management test suite (continued)

Test	Description	Release version
	Note: Requires the Knowledge Management Advanced Installer plugin.	
KM: Create an Ownership Group and check edit permissions and subscriptions	Verify the assignment of an article to an ownership group, that all members of the ownership group are subscribed to the article, and have permission to edit. **Note:** Requires the Knowledge Management Advanced Installer plugin.	New York
KM: Create, search for, and view knowledge articles with knowledge blocks	Verify the creation of a knowledge block and its addition to a knowledge article. Also, verify that the knowledge block content is searchable. **Note:** Requires the Knowledge Blocks plugin.	Orlando
KM: Validate the Knowledge - Approval Retire workflow	Verify the success of Knowledge - Approval Retire workflow by retiring a published knowledge article. **Note:** Requires the Knowledge Management Advanced Installer plugin.	Orlando
KM: Validate feedback task assignment to the ownership group manager	Verify the assignment of a feedback task to the ownership group manager of a knowledge article. **Note:** Requires the Knowledge Management Advanced Installer plugin.	Orlando
KM: Validate that the article template selector includes	Verify that the article template selector includes	Orlando

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

KM: Knowledge Management test suite (continued)

Test	Description	Release version
knowledge bases and article templates	all knowledge bases with contribute access and article templates applicable to each knowledge base. **Note:** Requires Knowledge Management Advanced Installer.	
KM: Validate the Knowledge - Approval Publish workflow	Verify the success of a Knowledge - Approval Publish workflow by recalling, rejecting, and then approving a knowledge article through the approval process. **Note:** Requires the Knowledge Management Advanced Installer plugin.	Orlando
KM: Make an outdated version of an article the latest version	Verify that an outdated version of a knowledge article is made the latest version. **Note:** Requires the Knowledge Management Advanced Installer plugin.	Orlando
KM: Validate that the mapped related articles appear in the Related articles widget	Verify that the related articles mapped to a knowledge article appear in the Related articles widget on the knowledge article view page. **Note:** Requires the Knowledge Management Advanced Installer plugin.	Orlando
KM: Validate that the mapped related catalog items appear in the Related items widget	Verify that the related catalog items mapped to a knowledge article appear in the Related items widget on the knowledge article view page.	Orlando

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

KM: Knowledge Management test suite (continued)

Test	Description	Release version
	Note: Requires the Knowledge Management Advanced Installer plugin.	
KM: Validate that a versioned article is created, published, checked out, and retired	Verify the success of UI actions on a versioned knowledge article by creating, publishing, updating, and retiring the knowledge article in Agent Workspace. **Note:** Requires the Knowledge Management Advanced Installer plugin.	Quebec
KM: Verify the scheduled publish feature for knowledge articles	Verify the scheduled publish feature for knowledge articles by creating an article in approval publish workflow, putting it for scheduled for publish and observing the change in workflow state from the**Scheduled for publish** state to the **Published** state. **Note:** Requires the Knowledge Management Advanced Installer plugin.	Quebec
KM: Validate the Knowledge - Approval Publish workflow in Agent Workspace	Verify the success of a Knowledge - Approval Publish workflow by recalling and then approving a knowledge article through the approval process in Agent Workspace. **Note:** Requires the Knowledge Management Advanced Installer plugin.	Quebec

To learn more about Knowledge Management, see Knowledge Management.

Leader Hub

Leader Hub quick start tests require activating the Leader Hub plugin (sn_egd_lh) and loading demo data.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Leader Hub quick start tests

Test	Description	Release version
[Leader hub] - Validate LH flows	• Verifies that the following widgets are present on the At a glance page for user Maria Davies: ○ Org talent ○ Skill growth area ○ Growth engagement • Verifies the employee card for the leader who has accessed Leader Hub appears at the top of the organizational chart on the Org talent page. • Verifies that selecting the **View team details** button on a card for an employee navigates the leader to the Team details page. • Verifies that the Skills widget is present on the Org Skills page. **Note:** Requires demo data.	Xanadu

To learn more about Leader Hub, see Leader Hub.

Legal Request Management

Legal Request Management quick start tests require installing the following apps:

- Legal Request Management (sn_lg_ops)
- Legal Counsel Center (sn_lg_workspace)

Legal Request test suite

Test	Description	Release version
LRM: Privacy Request Test	Tests to validate submission and fulfillment of Privacy legal requests.	Utah

To learn more about Legal Request Management, see Legal Request Management.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Major Incident Major Incident Management

Major Incident Management quick start tests require activating the Incident Management - Major Incident Management plugin (com.snc.incident.mim).

MIM: Major Incident Management test suite

Test	Description	Release version
MIM: Create a Major Incident	Test to verify the creation of major incident from the application navigation module.	Orlando
MIM: Create a Major Incident Candidate	Test to verify the **Create a Major Incident Candidate** module from navigation menu.	Orlando
MIM: Create a Major Incident from a Major Incident Candidate	Test to verify the creation of a major incident from a major incident candidate.	Orlando
MIM: Major Incident Candidate creation when it matches Major Incident Management trigger rule	Test to verify the creation of a major incident candidate when conditions to create a major incident match the major incident management trigger rule.	Orlando
MIM: Propose a major Incident (Assignment group empty)	Test to verify auto assignment of major incident candidate when an incident is proposed as a major incident and the **Assignment group** and the **Assigned to** fields are empty.	Orlando
MIM: Propose a major Incident (Assignment group is not empty)	Test to verify the auto assignment of a major incident when an incident is proposed as major incident and the **Assignment Group** and the **Assigned to** fields are not empty.	Orlando
MIM: Promote a candidate to major Incident (Assignment group is empty)	Test to verify auto assignment of major incident when a candidate is promoted to a major incident and the **Assignment Group** is empty.	Orlando
MIM: Promote a candidate to major Incident (Assignment group is not empty)	Test to verify auto assignment of major incident when a candidate is promoted to a major incident and the **Assignment group** is not empty.	Orlando
MIM: Reject a Major Incident Candidate	Test to verify the rejection of a major incident candidate.	Orlando

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

MIM: Major Incident Management test suite (continued)

Test	Description	Release version
MIM: Demote a Major Incident	Test to verify whether a major incident gets demoted to an incident.	Orlando
MIM: Major Incident closure	Test to validate the major incident closure functionality.	Orlando
MIM: State sync up with ICP and ICT	Test is to verify the state sync up with Incident Communication Plan and Incident Communication Task.	Orlando
MIM: ICP attached to an Incident based on conditions and its state sync up with ICT	Test is to verify Incident Communication Plan attached to an Incident based on defined conditions.	Orlando
MIM: Resolving MI and PIR	Test is to verify resolving a Major Incident and Post Incident Report.	Orlando
MIM: Major Incident workbench layout verification	Test is to verify Major Incident workbench layout verification.	Orlando
MIM: Verify communication task from MI workbench	Test is to verify communication task from MI workbench.	Orlando

To learn more about Major Incident Management, see Major Incident Management.

On-Call Scheduling

Activate the On-Call Scheduling plugin (com.snc.on_call_rotation) to run the On-Call Scheduling quick start tests.

On-Call: On-Call scheduling ATF Suites

Test	Description	Release version
On-Call: Verify request time-off when PTO approval preferences is Not allowed	Verify whether you can request a time-off when PTO approval preferences is set to **Not allowed**	Orlando
On-Call: Create overlapping shifts without selecting a template	Verify whether you can create overlapping shifts without selecting a template.	Orlando
On-Call: Create overlapping shifts when allow shift overlap is set to No	Verify whether you can create overlapping shifts when **Allow overlap** is set to **No**.	Orlando
On-Call: Request time-off when PTO approval preferences is with approval	Verify whether you can request a time-off when PTO approval preference is **With approval**.	Orlando

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

On-Call: On-Call scheduling ATF Suites (continued)

Test	Description	Release version
On-Call: Assign shift managers for maintaining on-call schedules.	Verify whether shift managers can maintain on-call schedules.	Orlando
On-Call: Search on-call schedules by user name	Verify whether you can search for on-call schedules by user name.	Orlando
On-Call: Search on-call schedules by group name or schedule name	Verify whether you can search for on-call schedules by group name or schedule.	Orlando
On-Call: Roster details for overlapping shifts with the escalation rule 'Escalate to incoming shift'	Verify the roster details for overlapping shifts when the escalation rule is set to **Escalate to incoming shifts**.	Orlando
On-Call: Calendar preview with timezone options	Verify whether the calendar preview is available with the timezone options.	Orlando
On-Call: Create overlapping shifts by selecting a template	Verify whether you can create overlapping shifts by selecting a template.	Orlando
On-Call: Show active shifts		Orlando
On-Call: Make shift draft/ publish form shift form	Verify whether you can publish/unpublish a shift and alternate between making the rota **Draft** and **Publish** state from the form.	Orlando
On-Call: Roster Details for overlapping shifts with the escalation rule 'Escalate to outgoing shift'	Verify the roster details for overlapping shifts when the escalation rule is set to **Escalate to outgoing shifts**.	Orlando
On-Call: Roster Details for overlapping shifts with the escalation rule 'Escalate to all shifts'	Verify the roster details for overlapping shifts when the escalation rule is set to **Escalate to all shifts**.	Orlando
On-call: Verify On-Call workbench	Test to verify the On-Call workbench layout.	Orlando
On-call: Hide or show shifts	Verify whether you can hide or show shifts from On-Call calendar	Orlando

To learn more about On-Call Scheduling, see On-Call Scheduling.

Metric Intelligence

Metric Intelligence quick start tests require activating the Metric Intelligence plugin (com.snc.sa.metric).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Metric Intelligence quick start tests

Test	Description	Release version
OI: Health Metrics Collection	An individual test that verifies the following: • There is at least one Metric Intelligence Metrics extension that is running. • There are entries for the 'Health Metrics' in the Metric to CI table. • For each of the Metric Intelligence Metrics extensions that are currently running, that there is data stored in MetricBase. If the test initially fails, wait until the Metric Intelligence Metrics extension runs for at least 10-15 minutes. Then try again.	Madrid

To learn more about Metric Intelligence, see Metric Intelligence.

Predictive Intelligence

Predictive Intelligence quick start tests require activating the Predictive Intelligence [com.glide.platform_ml] plugin. To execute critical upgrade tests on existing machine learning solutions, create a basic authorization profile named ml_atf in the Basic Auth Configurations table (sys_auth_profile_basic.list). To run the tests successfully, the user attached to the ml_atf authorization profile must have the ml_admin role.

Predictive Intelligence: ATF Test Suite

Test	Description
PI: Presence of ML model artifacts persisted in glide	Verify all the trained ML model artifacts are persisted in glide (sys_attachments table) after model training/instance cloning so that ML prediction calls are successful.
PI: Valid setup of ML user (sharedservice.worker) in glide	Validate if the ML user in glide (sharedservice.worker) is active and not logged out so that model training is successful.
PI: Glide upgrade test for Classification solution	Validate that the classification model prediction on existing active models is producing the same class membership and confidence value results after a glide upgrade.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Predictive Intelligence: ATF Test Suite (continued)

Test	Description
PI: Glide upgrade test for Similarity solution	Validate that the similarity model prediction API calls on active models are successful after a glide upgrade.

Problem Management

Problem Management quick start tests require activating the Problem Management Best Practice — Madrid plugin (com.snc.best_practice.problem.madrid) and the Problem Management — ATF Tests plugin (com.snc.problem.atf). For all state related test, the Problem Management State Model (com.snc.best_practice.problem.madrid.state_model) plugin needs to be active.

PRB MGMT: Problem Management test suite

Test	Description	Release version
PRB MGMT: Cancel a Problem when the state of the Problem is Assess	Verify that when a Problem in the **Assess** state is canceled, the state of the Problem changes to **Closed** with **Resolution code** as **Canceled**.	Madrid
PRB MGMT: Cancel a Problem when the state of the Problem is Root Cause Analysis	Verify that when a Problem is in the **Root Cause Analysis** state and is canceled, the state of the Problem changes to **Closed** with **Resolution code** as **Canceled**.	Madrid
PRB MGMT: Mark a Problem as Duplicate when the state of the Problem is Assess	Verify that when a Problem is in the **Assess** state and is marked as duplicate, the state of the Problem changes to **Closed** with **Resolution code** as **Duplicate**.	Madrid
PRB MGMT: Mark a Problem as Duplicate when the state of the Problem is Root Cause analysis	Verify that when a Problem is in the **Root Cause Analysis** state and is marked as duplicate, the state of the Problem changes to **Closed** with **Resolution code** as **Duplicate**.	Madrid
PRB MGMT: Accept Risk of Problem (problem.acceptrisk.move_to_closed_risk_accepted in Progress)	Verify that when a Problem state is **Fix in Progress** and the risk is accepted, then the Problem state changes to **Resolved** with **Resolution code** as **Risk Accepted**.	Madrid

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

PRB MGMT: Problem Management test suite (continued)

Test	Description	Release version
	Note: The test is valid when Problem property **Accept Risk moves the Problem to Closed state instead of Resolved state** *(problem.acceptrisk.move_to_closed)* is false.	
PRB MGMT: Accept Risk of Problem (problem.acceptrisk.move_to_closed is true, state is Fix in Progress)	Verify that when a Problem state is **Fix in Progress** and the risk is accepted, then the Problem state changes to **Closed** with **Resolution code** as **Risk Accepted**. **Note:** The test is valid when Problem property **Accept Risk moves the Problem to Closed state instead of Resolved state** *(problem.acceptrisk.move_to_closed)* is true.	Madrid
PRB MGMT: Accept Risk of Problem (problem.acceptrisk.move_to_closed is false, state is Root Cause Analysis)	Verify that when a Problem state is **Root Cause Analysis** and the risk is accepted, then the Problem state changes to **Resolved** with **Resolution code** as **Risk Accepted**. **Note:** The test is valid when Problem property **Accept Risk moves the Problem to Closed state instead of Resolved state** *(problem.acceptrisk.move_to_closed)* is false.	Madrid
PRB MGMT: Accept Risk of Problem(problem.acceptrisk.move_to_closed is true, state is Root Cause Analysis)	Verify that when a Problem state is **Root Cause Analysis** and the risk is accepted, then the Problem state changes to **Closed** with **Resolution code** as **Risk Accepted**.	Madrid

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

PRB MGMT: Problem Management test suite (continued)

Test	Description	Release version
	Note: The test is valid when Problem property **Accept Risk moves the Problem to Closed state instead of Resolved state** *(problem.acceptrisk.move_to_closed)* is true.	
PRB MGMT: Reanalyze Problem which is closed-Risk Accepted from state Root Cause Analysis	Verify that when a Problem is reanalyzed after it is **Closed** with the **Resolution code** as **Risk Accepted**, Problem state changes to **Root Cause Analysis**.	Madrid
PRB MGMT: Create Emergency Change from Problem	Verify the creation of Emergency Change from a Problem.	Madrid
PRB MGMT: Create Normal Change from Problem	Verify the creation of Normal Change from a Problem.	Madrid
PRB MGMT: Problem State Management	Verify problem state management.	Madrid
PRB MGMT: Reanalyze a Problem from Complete	Verify that when a Problem is reanalyzed after it is **Closed** with the **Resolution code** as **Fix Applied**, Problem state changes to **Root Cause Analysis**.	Madrid
PRB MGMT: Reanalyze a Problem which is canceled from state Assess	Verify that when a Problem is reanalyzed after it is **Closed** with the **Resolution code** as **Canceled**, Problem state changes to **Root Cause Analysis**.	Madrid
PRB MGMT: Reanalyze a Problem which is canceled from state Root Cause Analysis	Verify that when a Problem is reanalyzed after it is **Closed** with the **Resolution code** as **Canceled**, Problem state changes to **Root Cause Analysis**.	Madrid
PRB MGMT: Reanalyze Problem which is closed-Risk Accepted from state Fix in progress	Verify that when a Problem is reanalyzed after it is **Closed** with the **Resolution code** as **Risk Accepted**, Problem state changes to **Root Cause Analysis**.	Madrid

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

PRB MGMT: Problem Management test suite (continued)

Test	Description	Release version
PRB MGMT: Create a Known Error article from Problem	Verify creation of Known Error article from a Problem.	Madrid
PRB MGMT: Risk Accept reason on Incident	Verify the **Risk Accepted reason** is copied to the Incidents which are awaiting resolution of a Problem.	Madrid
PRB MGMT: Communicate Fix	Verify the communicate fix functionality.	Madrid
PRB MGMT: Communicate Workaround	Verify the communicate workaround functionality.	Madrid
PRB MGMT: Fix information on Incident	Verify that when a Problem is resolved, the state of the Incidents that are awaiting resolution of the Problem changes to **Resolved**. The fix notes of the Problem are copied to the Incidents.	Madrid
PRB MGMT: Problem task (Type:General) state management	Verify Problem task state management of a general type Problem.	Madrid

To learn more about Problem Management, see Problem Management.

Project Portfolio Management

Project Portfolio Management quick start tests require activating the PPM Standard - ATF Tests plugin (com.snc.financial_planning_pmo.atf).

PMO: Financial Tests for verifying cost rollups and demand to project conversion test suite

Test	Description	Release version
PMO: Verify cost plan roll up to project/demand and program	Validate the total planned cost rollup from project and demand to program.	Madrid
PMO: Verify cost plan roll up to project/demand, program and portfolio	Validate the total planned cost rollup from project and demand to portfolio.	Madrid
PMO: Verify financials of Project created from Demand - Simple Financials	Validate the financial tab fields of a project created from a demand.	Madrid
PMO: Verify financials of Project created from Demand - With budget, cost plans, benefit plans	Validate the budget, cost plan, and benefit plan of a project created from a demand.	Madrid

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

PMO: Financial Tests for verifying cost rollups and demand to project conversion test suite (continued)

Test	Description	Release version
PMO: Verify ETC/EAC at cost plan breakdown and project level	Validate the EAC and ETC values at cost plan breakdown and project level.	Tokyo

PMO: Innovation Management Tests for verifying idea life cycle

Test	Description	Release version
Validate state changes of Idea	Validate that the state of idea is changed when an idea is accepted, rejected, or any other task is created from an idea.	San Diego

PMO: Project Management tests for validating basic life cycle and project rollups test suite

Test	Description	Release version
Validate PPM Cycle from Ideation to demand to project closure	Validate the flow of creating an idea, converting the idea to a demand, and then converting the demand to a project. **Note:** This test fails if the PPM Standard Multicurrency (com.snc.ppm_multicurrency) plugin is active.	Madrid
Validation of State and Date Rollup for Automatic Project	Validate the date and state rollup from tasks for a project of type Automatic.	Madrid
Validation of State and Date rollup for Manual Project	Validate the date and state rollup from tasks for a project of type Manual.	Madrid
Validation of project percent complete when all tasks are Closed Incomplete	Validate the project percent complete when all the tasks are closed as Closed Incomplete.	Orlando
Validate Project is not 100 percent complete if it has atleast one task as closed incomplete	Validate the project percent complete is not 100% when at least one task is Closed Incomplete or Closed Skipped with task % complete less than 100%.	Orlando
Validate waterfall project does not show Agile Planning Board	Validate that the waterfall projects cannot use the following:	Orlando

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

PMO: Project Management tests for validating basic life cycle and project rollups test suite (continued)

Test	Description	Release version
	• Add stories or epics • Add agile phase • Access Agile board from the project	
Validate dates are rolled up from existing projects to program	Validate that the start and end dates of the project are rolled up to the program to which the project belongs.	Paris
Validate dates are rolled up from new projects to program	Validate that the start and end dates of a new project are rolled up to the program to which the project belongs.	Paris
Add projects and demands with risks or issues to program	Validate that the projects and demands with risks or issue records associated with them are added to the program.	Paris
Add projects and demands with benefit plans to program	Validate that the projects and demands with benefit plans associated with them are added to the program.	Paris
Add projects and demands with cost plans to program	Validate that the projects and demands with cost plans associated with them are added to the program.	Paris
Validate program dates on addition of existing demands	Validate that the start and end date of the program are adjusted on addition of an existing demand.	Paris
Validate program dates on addition of new demands	Validate that the start and end date of the program are adjusted on addition of a new demand.	Paris
Validate program state rollup	Validate that the program state is rolled up from state of all the projects in the program.	Paris
Verify Demand task due date column field value empty	Validate that the **Due date** field is empty for a new demand task.	Rome
Verify baseline is created on Project & Demand when demand is converted to project	Validate that a project and demand baseline is created when a demand is converted to a project.	Rome
Verify timecard financial appears on Demand when time card submitted against demand task	Validate that the actual cost of the demand is updated when a time card is processed for a demand task.	Rome
Validate demand approved date	Validate that the approved dates for demand and the project created from the demand are the same.	Tokyo
Validate project preference "close project milestone tasks when they change to work state"	Validate that milestone tasks are closed when their state changes to Work in Progress. The project preference "close project milestone tasks when they change to work state" must be set.	Tokyo

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

PMO: Project Management tests for validating basic life cycle and project rollups test suite (continued)

Test	Description	Release version
Validate project preference "Enable move project for WIP projects"	Validate that the project start date changes appropriately for a WIP project after setting the "Enable move project for WIP projects" preference.	Tokyo
Validate project preference "Rollup project start date from tasks"	Validate that the project task start date does not roll up to the project after deselecting the "Rollup project start date from tasks" preference.	Tokyo

Child Test Suite: Validation of Move Project Action

Test	Description	Release version
Validate Planned start date of a project can be shifted by using the Move project action	Validate that the Planned start date can be updated to a later or earlier date using the Move project related link for projects in Planning or Open state.	Orlando
Validate Move Project is disabled when Project is selected for execution	Validate that the Move Project option is not available if the Project is in Execution phase.	Orlando
Validate Move Project functionality with sprint dates populated for an agile phase	Validate that the sprint start and end dates are cleared when the project is moved using the Move Project related link. **Note:** This test is available only when Agile Development 2.0 plugin (com.snc.sdlc.agile.2.0) is activated.	Orlando
Validate Move Project functionality with external dependencies and related entities	Validate that the external dependencies and related entities are also shifted and redrawn when the project is moved using the Move Project related link.	Orlando
Validate Move Project functionality with different project states	Validate that the Move Project does not work when the project is in Work In Progress or Closed Complete state.	Orlando

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Child Test Suite: Verify RIDAC flow of a Project

Test	Description	Release version
Verify RIDAC flow for Risk	Validate the RIDAC flow for a risk associated with a project.	Quebec
Verify RIDAC flow for Issue	Validate the RIDAC flow for an issue associated with a project.	Quebec
Verify RIDAC flow for Decision	Validate the RIDAC flow for a decision associated with a project.	Quebec
Verify RIDAC flow for Action	Validate the RIDAC flow for an action associated with a project.	Quebec
Verify changes in fields of risk form	Validate that any change is the Risk form fields such as Risk rank, Risk value, and Probability is updated successfully upon submission.	Quebec

Child Test Suite: Verify RIDAC flow of a Demand

Test	Description	Release version
Verify RIDAC flow for Risk	Validate the RIDAC flow for a risk associated with a demand.	Quebec
Verify RIDAC flow for Issue	Validate the RIDAC flow for an issue associated with a demand.	Quebec
Verify RIDAC flow for Decision	Validate the RIDAC flow for a decision associated with a demand.	Quebec
Verify RIDAC flow for Action	Validate the RIDAC flow for an action associated with a demand.	Quebec
Verify changes in fields of risk form	Validate that any change is the Risk form fields such as Risk rank, Risk value, and Probability is updated successfully upon submission.	Quebec

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

PMO: Resource Management tests for verifying the resource plan flows test suite

Test	Description	Release version
Verify user resource plan flow from Planned to Canceled state	Validate that the resource plan of a project can be moved to canceled state from planned state.	Madrid
Verify group resource plan flow from Planned to Complete state	Validate that the resource plan of a project can be moved to complete state from planned state.	Madrid
Verify role resource plan flow from Planned to Allocated state	Validate that the resource plan of a project can be moved to allocated state from planned state.	Madrid
Verify aggregated cost of all resource plans roll up to the corresponding project or demand fields	Validate that the aggregated cost of all resource plans on a project or demand roll up to the Planned Cost and Allocated Cost fields. Verify the costs also roll up to the Resource Cost section of respective projects and demands.	Orlando
Verify Copy Resource plan option	Validate that the Copy Resource plan option creates an exact copy of the source resource plan in the Planning state	Orlando
Verify resource plan aggregate roll up from project/ demand to program	Validate that the aggregated cost of all resource plans on a project or demand roll up to the total planned cost of the associated program.	Orlando
Verify records on completion of a resource plan	Validate the changes in a resource plan on completion: • The state of the resource plan is updated to Completed. • If the completion date is earlier than the resource plan end date, the end date of the resource plan is updated with the completion date. If the completion date entered is later than the resource plan end date, the resource plan end date is retained. • All the requested and allocation records for the resource plan for the period after the completion date are deleted. If there are any actual hours logged against an allocation, that allocation is not deleted. For those allocation records, the allocated hours become zero and the actual hours are retained. • The available and allocated hours for resources are updated in the aggregate tables.	Orlando
Verify records on completion of a resource plan with	Validate the following on completion of a resource plan with Planned Duration as allocation type:	Orlando

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

PMO: Resource Management tests for verifying the resource plan flows test suite (continued)

Test	Description	Release version
Planned Duration as allocation type	• The state of the resource plan is updated to Completed. • Allocations are not deleted. • End date of the allocation is updated to the completion date.	
Verify the RP replan Capability	Validate that when a cancelled resource plan is re-planned, the state of the resource plan changes to Planning.	Orlando
Verify whether change in resource plan is reflected in corresponding cost plan	Validate that when a resource plan is updated, the corresponding cost plan is updated accordingly. For example, the total planned cost is 500 USD and the planned hours is 10. Changing the planned hours to 20 updates the total planned cost to 1000 USD.	Paris
Resource-Test the default population of resource plan start & end date	Validate the following on creating a user or group resource plan from the related list of a demand: • If a demand is created without a start date and end date, the user or group resource plan has task as demand and no start and end date. • If a demand is created with a start date and end date, the user resource plan has the task as demand with those dates.	Paris
Validate that actual hours in operational resource plan and time card are equal	Validate that a time card category is mapped with an operational work type. On submitting the time card, verify that the actual hours in the resource plan and time card are equal.	Quebec
Verify Resource plan auto population for Operational plans	Validate that the operational resource plans associated with a time card are automatically retrieved on the time card when time is logged.	Rome
Verify resource plan auto population for non-operational resource plans	Validate that the resource plans associated with a project, project task, or demand for a time card are automatically retrieved on the time card when time is logged.	Rome

To learn more about Project Portfolio Management, see Project Portfolio Management.

Project Currency test suite

Project currency quick start tests require activating the PPM Standard Multicurrency – ATF Tests plugin (com.snc.ppm_multicurrency.atf).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Product currency test suite tests

Test	Description	Release version
Verify cost in project currency on cost plan	Validate the calculation of cost line breakdown with budget reference rate and verify roll up to cost plan and also for the project in project currency.	Orlando
Verify benefit in project currency on benefit plan	Validate the calculation of benefit line breakdown with budget reference rate and verify roll up to benefit plan and also for the project in project currency.	Orlando

To learn more about PPM Standard, see PPM Standard.

Reporting

The Reporting quick start test Automated Test Framework - Reporting plugin (com.glide.automated_testing_impl.report) is active by default or instance reboot.

Note: Reporting quick start tests do not test report access from dashboards. To test dashboards, see Quick start tests for Dashboards.

Reporting quick start tests

Test	Description	Release version
Report Visibility	Confirm whether reports are still visible to users whom they are shared with.	Madrid

Basic Software Asset Management Software Asset Management

Software Asset Management quick start tests require activating the Software Asset Management Professional plugin (com.snc.samp). Some quick start tests require activating the following additional plugins.

- Software Asset Management - Spend Detection (com.sn_sam_spend)
- Software Asset Management Professional for Microsoft (com.snc.samp.microsoft)
- Software Asset Management Professional for SAP (com.sn_samp_sap)
- Software Asset Management Professional for Oracle (com.snc.samp.oracle)
- Software Asset Management - SaaS License Management Integrations (sn_sam_saas_int)

Software Asset Management test suite

Test	Description	Release version
SAM - Mapping Custom PPN/ DMAP to newly published PPN and Content	Validates the replacement of a custom publisher part number (PPN) with a new Software	San Diego

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Software Asset Management test suite (continued)

Test	Description	Release version
	Asset Management Content Service PPN.	
SAM - Oracle PaaS BYOL	• Validates the addition of the new Serverless Hardware [cmdb_ci_serverless_hardware] table, which stores information about PaaS devices. • Validates the license compliance of Oracle Database servers in Amazon Web Services (AWS) PaaS environments. **Note:** Requires the Software Asset Management Professional for Oracle (com.snc.samp.oracle) plugin and the CMDB CI Class Models store application.	Rome
SAM - Software Product Lifecycles records on Software Model	Validates that the Software Lifecycle tab on the Software Model form is showing records. **Note:** Requires demo data and the Software Asset Management Professional for Microsoft (com.snc.samp.microsoft) plugin.	Quebec
SAM - BYOL	• Validates the purchase date on the Software Entitlement form • Validates the addition of newly added column legacy_license on the License Metric Results [samp_license_metric_result] and License Position Report [samp_license_position_report] tables	Quebec

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Software Asset Management test suite (continued)

Test	Description	Release version
	• Validates the addition of newly added columns, cloud_license_type and cloud_license _type_source in the Software Installations [cmdb_sam_sw_install] table **Note:** Requires the Software Asset Management Professional for Micorosft (com.snc.samp.microsoft) plugin.	
SAM - Validate CIS Suites	Validates reconciliation of Core Infrastructure Server (CIS) suites along with downgrade rights. **Note:** Requires demo data and the Software Asset Management Professional for Microsoft (com.snc.samp.microsoft) plugin.	Paris
SAM - PerCoreForMSCluster	Verifies the reconciliation functionality for Microsoft per core with cluster. **Note:** Requires the Software Asset Management Professional for Microsoft (com.snc.samp.microsoft) plugin.	Paris
SAM - Validate upgrade/ downgrade during Reconciliation for Microsoft publisher	Validates upgrade and downgrade rights during reconciliation for Microsoft products.	Paris

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Software Asset Management test suite (continued)

Test	Description	Release version
	Note: Requires demo data and the Software Asset Management Professional for Microsoft (com.snc.samp.microsoft) plugin.	
SAM - Generate demand to consolidate SaaS applications	Validates generation and submission of a demand on SaaS applications. **Note:** Requires the Software Asset Management - Spend Detection (com.sn_sam_spend) plugin and the PPM standard (com.snc.financial_planning_pmo) plugin.	Paris
SAM - Software Model and Software Entitlement checks for SAP Engines	Verifies that the Software Model and Software Entitlement forms change when the product is an SAP engine. **Note:** Requires the Software Asset Management Professional for SAP (com.sn_samp_sap) plugin.	Paris
SAM - Downgrade Rights on Software Model	Validates that the downgrade rights pushed from the content service are correctly populated on the Downgrade Rights related list on the software model form.	Orlando
SAM - Multi-core pack validation on Software Entitlement	Validates the functionality of new fields for a multi-core pack on software entitlements.	Orlando

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Software Asset Management test suite (continued)

Test	Description	Release version
	Note: Requires the Software Asset Management Professional for Microsoft (com.snc.samp.microsoft) plugin.	
SAM - Downgrade Rights on Software Entitlement	Validates that the downgrade rights pushed from the content service are correctly populated on the Downgrade Rights related list on the software entitlement form.	Orlando
SAM - Software Spend Transaction	Validates the creation of a Software Spend Transaction. **Note:** Requires the Software Asset Management - Spend Detection (com.sn_sam_spend) plugin.	Orlando
SAM - Software Model and Software Entitlement	Tests that a user can create a software model and software entitlement and validates those records.	New York
SAM - Software Installation and Discovery Model	Tests that a user can create a software installation and discovery model and validates those records.	New York
SAM - Software Entitlement Creation Using Custom PPN	Creates a custom software product, custom DMAP, custom Part Number, and software entitlement using the custom Part Number. Verifies that a software model is automatically created.	New York
SAM - Software Model Checks for SAP Named Users	Tests that the software model form changes when the publisher is SAP and the product is Named Users.	New York

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Software Asset Management test suite (continued)

Test	Description	Release version
	Note: Requires the Software Asset Management Professional for SAP (com.sn_samp_sap) plugin.	
SAM - Software Model Checks for SaaS	Tests that the Software Model form changes when a SaaS product is selected. **Note:** Requires the Software Asset Management - SaaS License Management (sn_sam_saas_int) plugin.	New York
SAM - Validate Fields on SaaS Software Products	Tests that the **Subscription software** and **Ignore installs** fields are present on the Software Product form. **Note:** Requires the Software Asset Management - SaaS License Management (sn_sam_saas_int) plugin.	New York

To learn more about Software Asset Management, see Software Asset Management.

Security Incident Response

Security Incident Response quick start tests require activating Security Incident Response plugin (com.snc.security_incident) and loading the demo data.

Security Incident Response tests

Test	Description	Release version
SIR: Create Security Incident	Determine whether a user can successfully create a security incident from the security incident form.	Madrid
SIR: Create Security Incident via Security Incident Catalog	Determine whether a user can successfully create a security incident from the catalog.	Madrid

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Security Incident Response tests (continued)

Test	Description	Release version
SIR: PIR Assessments OOTB configuration test	Use this test to validate PIR assessments and base system configurations.	Tokyo
SIR: PIR Assessments conditional Configuration tests	Verify that security incidents matching the mandatory conditional rule are not closed without completing the post incident assessment. Verify that the security incidents matching the optional conditional rule can be closed without completing the post incident assessment. Verify that assessments are not generated for the security incidents that do not match any rule.	Tokyo
SIR: PIR Run Time Experience	Verify that PIR reports are configured and attached to the security incidents as per the new design.	Tokyo
SIR: PIR Design Time Experience	Verify that the security incident is mapped with the report template based on the administrator configuration.	Tokyo
SIR: Link Security Incident to a existing Major Security Incident	Link a Security Incident to an existing Major Security Incident and validate data from Security Incident rolled up to Major Security Incident.	Tokyo
SIR: Promote Security Incident as Major Security Incident	Promote a Security Incident as Major Security Incident and validate data from Security Incident rolled up to Major Security Incident.	Tokyo
SIR: Propose Security Incident as Major Security Incident	Propose a Security Incident as Major Security Incident and validate data from Security Incident rolled up to Major Security Incident.	Tokyo
SIR: Security Incident life cycle	Validate a Security Incident life cycle with the policy violation response tasks workflow.	Yokohama

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Security Incident Response tests (continued)

Test	Description	Release version
SIR: Create Security Case	Create a Security Case from the Security Incident form.	Yokohama
Verify that only Allowed Members can access the security incident once Enforce Restriction is ON	Verify that only the allowed members can access the security incident once the Enforce Restriction is enabled.	Yokohama
Verify that security incident enabled with "Enforce Restriction" is not visible for any user	Verify that security incident enabled with "Enforce Restriction" is not visible for any user.	Yokohama
Validate Read Access	Validate the view access.	Yokohama
Validate Write Access	Validate the edit access.	Yokohama
SIR Workspace: Read Access	Verify that Read Access user can view the security incident without having security roles even on workspace.	Yokohama
SIR Workspace: Write Access	Verify that Write Access user can update the security incident without having security roles.	Yokohama
SIR Workspace: Create new Security Incident	Create new security incident from workspace.	Yokohama
SIR Workspace: Create Response Task	Create new response task from an existing security incident.	Yokohama
Now Assist for Security: Active Security Incident Summarization	Summarize an active security incident and validate the displayed sections.	Zurich
Now Assist for Security: Security Incident Summarization_Share to worknotes	Share the generated summary to worknotes.	Zurich
Now Assist for Security: Closed Security Incident Summarization	Summarize a closed security incident and validate the displayed sections.	Zurich

To learn more about Security Incident Response, see Security Incident Response.

Service Level Management

Service Level Management quick start tests require activating the Service Level Management - ATF Tests plugin (com.snc.service_level_management.atf).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Service Level Management quick start tests

Test	Description	Release version
SLM: Service Level Management Task SLA completed workflow	Tests that a user can create a task SLA and complete the workflow.	Orlando
SLM: Service Level Management Task SLA cancelled workflow	Tests that a user can create a task SLA and cancel the workflow.	Orlando
SLM: Service Level Management Task SLA timer REST API	Tests that the SLA Timer API response matches data expected by SLA Timer Seismic component.	Paris

To learn more about Service Level Management, see Service Level Management.

Service Mapping

Service Mapping quick start tests require activating the Service Mapping (com.snc.service-mapping) plugin and loading the demo data.

Service Mapping quick start tests

Test	Description	Release version
SM OOTB: Tests SM application visibility by roles	Validate the visibility of Service Mapping applications and modules for different roles. For example, the test verifies that a user logged in with the service_mapping_user role cannot access the Administration module under Service Mapping.	Madrid
SM OOTB: Service map verification	Use this test template to create custom tests for verifying that the topology of the most significant services is unchanged. Configure values under Test Run Data Sets, to identify the name of the service to test and the nodes that you expect to find in this service. For the node name values, enter the node attribute exactly as it appears on the map, for example, "Apache server." If the attribute name for a node is truncated or	Orlando, updated Paris

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Service Mapping quick start tests (continued)

Test	Description	Release version
	shows that a node is a CI cluster, configure this node name value to reflect the way it appears on the map, for example, "11x ExchangeF" for a group of 11 Exchange FrontEnd servers. **Note:** You cannot use this test to verify the service content beyond the top, unexpanded level.	
SM OOTB: Check UI accessibility after an upgrade	Run this test to check that the Service Mapping UI is fully functional after an upgrade.	Orlando

To learn more about Service Mapping, see Service Mapping.

Service Portfolio Management Premium

Service Portfolio Management Premium quick start tests require activating the Service Portfolio Management Premium plugin (com.snc.spm).

Service Portfolio Management Premium - ATF Tests test suite

Test	Description	Release version
Portfolio Editor: Not Portfolio Owner, Read Only Taxonomy Access	Ensure a portfolio editor can only read associated non-owned portfolio taxonomies.	Orlando
Portfolio Editor: Cannot create new Portfolios	Ensure a portfolio editor cannot create new portfolios.	Orlando
Portfolio Editor: Valid Portfolio Owner Taxonomy Access	Ensure a portfolio owner has access to taxonomies within owned portfolios.	Orlando
Service Editor: Access	Ensure a service editor can only edit owned services and offerings.	Orlando
Portfolio Editor: Can Modify Owned Portfolios	Ensure a portfolio editor can modify and update owned portfolios.	Orlando
Create a Portfolio	Ensure a portfolio admin can create a new portfolio.	Orlando
Portfolio Editor: Cannot Modify Non-Owned Portfolios	Ensure a portfolio editor cannot modify non-owned portfolios.	Orlando

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Service Portfolio Management Premium - ATF Tests test suite (continued)

Test	Description	Release version
Service Workflow	Verify that a service cannot move forward to Catalog phase without a service portfolio, taxonomy node, and service offering attached to it. Verify a service cannot move backward from the Catalog phase to the Pipeline phase.	Orlando
Service Portfolio (Normal)	Create a service portfolio, taxonomy layer, and taxonomy nodes.	Orlando
Create a Portfolio + 3 Taxonomy Layers	Create a service portfolio with three taxonomy layers.	Orlando
Service Portfolio (Exception) - Taxonomy Layer Definition Set 01	Create a service portfolio, taxonomy layer, and taxonomy nodes.	Orlando
SPM: Create a Portfolio	Create a service portfolio.	Paris
SPM: Create a Portfolio + 3 Taxonomy Layers	Create a service portfolio, with three taxonomy layers.	Paris
SPM: Service Portfolio (Normal)	Create a service portfolio, taxonomy layer, and taxonomy nodes.	Paris
SPM: Service Portfolio (Exception) - Taxonomy Layer Definition Set 01	Create a service portfolio, taxonomy layer, and taxonomy nodes.	Paris
SPM: Portfolio Editor: Can Modify Owned Portfolios	Ensure a portfolio editor can modify portfolios that they own.	Paris
SPM: Portfolio Editor: Cannot Modify Non-Owned Portfolios	Ensure a portfolio editor cannot modify portfolios that they do not own.	Paris
SPM: Portfolio Editor: Cannot create new Portfolios	Ensure a portfolio editor cannot create new Portfolios.	Paris
SPM: Portfolio Editor: Valid Portfolio Owner Taxonomy Access	Ensure portfolio owners have access to taxonomies that they own.	Paris
SPM: Portfolio Editor: Not Portfolio Owner, Read Only Taxonomy Access	Ensure a portfolio editor can only read the taxonomies of portfolios they do not own.	Paris
SPM: Service Editor: Access	Ensure a service editor can only edit services and	Paris

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Service Portfolio Management Premium - ATF Tests test suite (continued)

Test	Description	Release version
	offerings that they own or are a delegate of.	
SPM: Service Viewer: Access	Ensure a service viewer has access to view services.	Paris
SPM: Service Workflow	Verify a service cannot move forward to Catalog phase without a service portfolio, taxonomy node, and service offering attached to it. Verify a service cannot move backward from Catalog to Pipeline phase.	Paris

To learn more about Service Portfolio Management Premium, see Service Portfolio Management Premium.

Skills Management

Skills Management quick start tests require activating the Skills Management plugin (com.snc.skills_management).

Skills Management test suite

Test	Description	Release version
Skills MGMT: User skill level inheritance when user is part of multiple groups	Verify that the user is assigned the highest skill level when the user belongs to multiple groups that have been assigned the same skills with different skill levels.	New York
Skills MGMT: Add skills to lowest level category	Verify that skills can be added to the lowest level category.	New York
Skills MGMT: Create a child category under a parent category	Verify that a lower-level category can be created when the flag for **Add skills** is unchecked in the parent category.	New York
Skills MGMT: Create a skill category	Verify that a skill category can be created on the skill category form.	New York
Skills MGMT: Create skill level type and skill levels	Define the skill level type and different skill levels for the type.	New York
Skill MGMT: Skill level inheritance from group to user	Verify that skill levels can be inherited from a group to the users of the group and that the **Inherited** and **Skill level inherited** fields are set to true.	New York

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Skills Management test suite (continued)

Test	Description	Release version
Skills MGMT: Create a skill from Manage IT Skills user interface.	Verify that you can create a skill from the Manage IT Skills user interface.	Orlando
Skills MGMT: Add skills and skill levels to users in the Manage IT Skills user interface.	Verify that you can add skills and associated skill levels to users in the Manage IT Skills user interface.	Orlando
Skills MGMT: Select a skill and add users to the skill in the Manage IT Skills user interface.	Verify that you can select a skill and add the skill and associated skill levels to one or more users in the Manage IT Skills user interface.	Orlando
Skills MGMT: Select a skill that does not have a skill level and add users to that skill in the Manage IT Skills user interface.	Verify that you can select a skill without a skill level and add it to users in the Manage IT skills user interface.	Orlando

To learn more about Skills Management, see Skills Management.

Test Management 2.0

Test Management 2.0 quick start tests require activating the Test Management 2.0 plugin (com.snc.test_management.2.0). and the Test Management 2.0 - ATF Tests plugin (com.snc.test_management.2.0.atf).

Test Management 2.0: Test version test suite

Test	Description	Release version
Create test version should create test	Validate test creation and version.	Madrid
Should be able to mark test version as ready when it contains verification steps	Validate test state when test has verification steps.	Madrid
Should not able to mark test version as ready when it does not contain verification step	Validate test state when test does not have verification steps.	Madrid
Marking a test version as ready should retire other test version in ready state	Validate test state when marking test ready.	Madrid

Test Management 2.0: Test results rollup test suite

Test	Description	Release version
When test run closed, should update execution suite progress	Validate execution state progress.	Madrid

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Test Management 2.0: Test results rollup test suite (continued)

Test	Description	Release version
Should not be able to assign a test not in ready state	Validate test assignment.	Madrid
Test progress should roll up for test plan and test cycle	Validate test progress for test plan and test cycle.	Madrid

To learn more about Test Management 2.0, see Test Management 2.0.

Universal Request

Universal Request quick start tests require activating the Universal Request (com.snc.universal_request), Universal Request: Integration for Incident management (com.snc.incident.universal_request), and Human Resources Scoped App: Core (com.sn_hr_core) plugins.

Universal Request quick start tests

Test	Description	Release version
UR: Transferring an HR case to an Incident	Validate if an HR case is transferred to an incident and an incident is created.	Quebec
UR: Restrict or Unrestrict universal request	Validates the following: • The fields are restricted and hidden when a routing agent restricts a UR. • Only agents with *sensitiveinfo_agent* role are allowed to unrestrict the request. • And, after the request is unrestricted all fields are visible. • sys-audit log access is available for agents.	Yokohama
UR: Universal Request creation with RITM	Verifies a catalog request for any item from self-service can be created when Universal Request Certified is enabled.	San Diego

Vendor Manager Workspace and Third-party Risk Management

GRC: Vendor Manager Workspace and Third-party Risk Management quick start tests require activating the Vendor Manager Workspace and Third-party Risk Management plugin (com.sn_vdr_risk_asmt) and loading demo data.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

GRC: Vendor Risk Management and Third-party Risk Management Quick Start Tests

Test	Description
GRC: Create Engagement Risk Assessment	Creates and submits an engagement risk assessment to an engagement.
GRC: Vendor Risk Scoring - Cancel assessment for engagement	Verifies a request to recalculate a risk rating is automatically created on an Engagement and Third party after one assessment for the Engagement is cancelled.
GRC: Vendor Issue ATF	Creates and submits a third-party risk issue form and third-party risk task form.
GRC: Create Vendor Risk Assessment	Creates and submits a vendor risk assessment to a vendor.
GRC: Vendor Portal - Answer and Return Assessment	Vendor contact answers and submits assessment in the Service Vendor Portal.
GRC: Vendor Tiering Assessment	Selects and submits an assessment to respective assessors after changing the duration.

Verify that TPRM still works after you make configuration changes such as applying an upgrade or developing an application. Copy and customize the quick-start tests to pass when using your instance-specific data.

Vulnerability Response

Vulnerability Response quick start tests require activating the Vulnerability Response application (sn_vul) and loading the demo data.

Vulnerability Response tests

Test	Description	Release version
VR: Create Remediation Target Rule	Create a Remediation Target Rule.	Madrid
VR: Create Vulnerability Group Rule	Create a Vulnerability Group Rule.	Madrid
VR: Create Vulnerable Item via Form	Determine whether a user can successfully create a vulnerable item from the Vulnerable Item form.	Madrid
VR: Vulnerability Group Life Cycle	Determine whether a user can successfully resolve a vulnerability group.	Madrid
VR: Vulnerable Item life cycle	Determine whether a user can successfully move a vulnerable item through its life cycle, and also determine whether a closed vulnerable item can be reopened.	Madrid

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Vulnerability Response tests (continued)

Test	Description	Release version
VR: Rollup Calculator	Determine whether the rollup risk calculator can provide an overall risk score for an entire group of vulnerable items. The score uses the scores for all the vulnerable items in a vulnerability group.	New York
VR: Vulnerability Response Assignment Rules	Determine whether a sample set of assignment rules can successfully auto-assign vulnerable items to an assignment group for remediation.	New York
VR: Vulnerability Calculators	Test the vulnerability calculators.	New York
VR: CI Lookup - Qualys	Create a new lookup rule with method "field_matching" called "Lookup By Network Adapter" for Qualys. Determine whether a configuration item is successfully matched in the Discovered Item table by network adapter and IP address with the new lookup rule.	Orlando
VR: Create Normal and Emergency Change Request	Determine whether the user can successfully create normal and emergency change requests from a vulnerability group.	Orlando
VR: Split Vulnerability Group	Determine whether the user can successfully split a vulnerability group.	Orlando
VR: Update VG state when a CHG is cancelled.	Determine whether the State field on a vulnerability group successfully transitions when a change request that is associated with the vulnerability group is cancelled.	Orlando
VR: Update VG state when a CHG transitions to Review.	Determine whether the State field on a vulnerability group successfully transitions when a change request that is associated with it moves to the Review state.	Orlando

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Vulnerability Response tests (continued)

Test	Description	Release version
VR: CI Lookup - Rapid7	Test CI lookup using the existing Rapid7 Vulnerability Integration lookup rule, **IP Address**.	Orlando
VR: CI lookup - Qualys	Test CI lookup by creating a new lookup rule for the Qualys Vulnerability Integration	Orlando
VR: Exception Approval Workflow for VI	Create an exception request and verify that the approval process is working.	Orlando
VR: False Positive Approval Workflow for VI	Create a false positive exception request and verify that the approval process is working.	Orlando
VR: Application Vulnerability Response (AVR)	Determine whether your rules and calculators are working correctly. Verify that updates are working.	Orlando
Remediation target rules: VI import test	Tests VR remediation target rules during import.	Paris
VR: Classification Rule	A sample classification rule that automatically classifies a vulnerability.	Tokyo

To learn more about Vulnerability Response, see Vulnerability Response.

Walk-up Experience

Walk-up Experience quick start tests require activating the Walk-up Experience plugin (com.snc.walkup) and loading demo data.

Walk-up Experience - Tests test suite

Test	Description	Release version
Create a location	Ensure users with the Walk-up administrator [sn_walkup.walkup_admin] role can access all existing walk-up queues, create new queue locations, and configure queues appropriately.	Orlando
Onsite checkin (ITIL User)	Verify that users with the ITIL role can check into an onsite queue location.	Orlando

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Walk-up Experience - Tests test suite (continued)

Test	Description	Release version
Onsite checkin (ESS User)	Verify that users with the ESS role can check into an onsite queue location.	Orlando
Onsite checkin (Guest)	Verify that guest users can check into an onsite queue location.	Orlando

To learn more about Walk-up Experience, see Walk-up Experience.

Parallel testing

Reduce test design time by running multiple tests and test suites in parallel. Design tests to run in parallel by avoiding resource conflicts and data dependencies.

Note: If two or more users are developing tests simultaneously, parallel testing reduces test design time. After test design completes, it is recommended to organize tests into a single hierarchical suite structure, and run the tests as a single base suite.

Parallel testing limit

Parallel testing enables users to run multiple automated tests simultaneously. This process continues until the number of parallel running tests is as per the following formula.

```
Number of parallel tests = max(1, number of worker threads - 2)
```

Note: If your instance has 2 or less worker threads, configuration improvements review is recommended.

The actual number of parallel tests that a non-production instance can support depends on the system resources that the instance has when it is provisioned. The parallel testing limit ensures that an instance always has system resources available for other non-testing tasks.

Test waiting queue

When the system reaches the parallel testing limit, it reschedules tests to run later. It automatically places the tests back in sys_trigger until a worker thread is available to pick them up. Each test in the waiting queue has a schedule the next time the test runs.

Design considerations

Run multiple tests and test suites in parallel to reduce test design time. Avoid resource conflicts and data dependencies by designing parallel running tests. Avoid resource conflicts and data dependencies by designing tests that rely only on newly generated or self-created data, or have mutual exclusion rules defined between tests that share resources.

Prevent resource conflicts between parallel tests

Prevent resource conflicts by running tests that create their own data. Tests that run with existing data prevent other tests that need the same data from running in parallel.

Note: If you have two or more tests with resource conflicts, see Mark tests as mutually exclusive to create a mutual exclusion rule that prevents the tests from running in parallel.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Accelerate ATF tests failure resolution

Resolve ATF test failures faster using the actionable support provided by the new ATF failure insights feature. You can achieve it by accessing the Find changes since last successful run related link.

Before you begin

Role required: admin

Procedure

1. Navigate to **All** > **Automated Test Framework (ATF)** > **Tests**.
   A list of ATF tests shows up.
2. Select the test you want to debug.
   The selected test form shows up.
3. Select the Test Results related list.
   The list of last few execution results of the selected test shows up. If the test has never been executed, you don't see any test results. You can then execute the test to generate results.
4. Select one of the failed results to see what has changed since the last successful run of the test.
   The Test Results form of the selected result shows up.
5. Select the Find changes since last successful run related link.

   Note: This related link is visible only when you open a failed test result.

   The What changed related list shows up.

   Note: The What changed related list shows up only if you have metadata records that have changed in between the last successful run and the recent failure. It gives a better insight into what specific records could have caused the failure.

   You might experience a delayed update in the records in the What changed related list. If you still don't see the desired results, it is recommended to try selecting the Find changes since last successful run related link again.

6. Implement one of the following ways to compare the results.
   - Right-click on one of the changed files listed under the What changed related list. Select **Comparison** to see the changes since the last successful run.
   - Select and open the changed file itself and select **Comparison** on the What changed form.

     Note: A screen showing the comparison between the 2 versions of the selected file shows up. It shows the exact changes made to the file. The left hand side is the most recent version of the record.

   - Go to **All** > **Automated Test Framework (ATF)** > **Test Results**. Select any 2 tests and compare results by selecting the action.
7. **Optional:** Select **Create a task** on the Test Results form if you want to create a task for a changed file.
   You will have to select at least 1 changed file to create a task for. Certain information about the changed record are auto-filled.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: By default, the task created is an incident task type. This is determined by the sn_atf.test.triage.task.type property. It creates the task type depending on the task table you have assigned the task to.

If you select more than 1 changed file while creating a task, it still creates just 1 task for the multiple files.

Metadata exception list

The following list of tables are not supported by the ATF tests failure resolution feature.

- sys_ui_hp_cascading_filter
- sys_ui_hp_cascading_filter_target_references
- sys_ui_hp_date
- sys_ui_hp_group
- sys_ui_hp_publisher
- sys_ui_hp_reference
- sys_ui_annotation_type
- sys_ui_application
- sys_ui_extension_instance
- sys_ui_extension_point
- sys_ui_help
- sys_ui_module
- sys_ui_ng_action
- sys_ui_overview_help
- sys_ui_overview_help_panel
- sys_ui_reference_style
- sys_declarative_action_assignment
- sys_declarative_action_definition
- sys_declarative_action_model_definition
- sys_declarative_action_model_field
- sys_declarative_action_payload_definition
- sys_declarative_action_payload_mapping
- sys_declarative_action_render_mode
- sp_page
- sp_agent_chat_config
- sp_column
- sp_container
- sp_css
- sp_css_include
- sp_dependency
- sp_header_footer

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- sp_js_include
- sp_ng_template
- sp_page_title_variable
- sp_portal
- sp_row
- sc_catalog
- sc_catagory
- sc_cat_item
- sc_cat_item_delivery_task
- sc_cat_item_content
- catalog_script_client

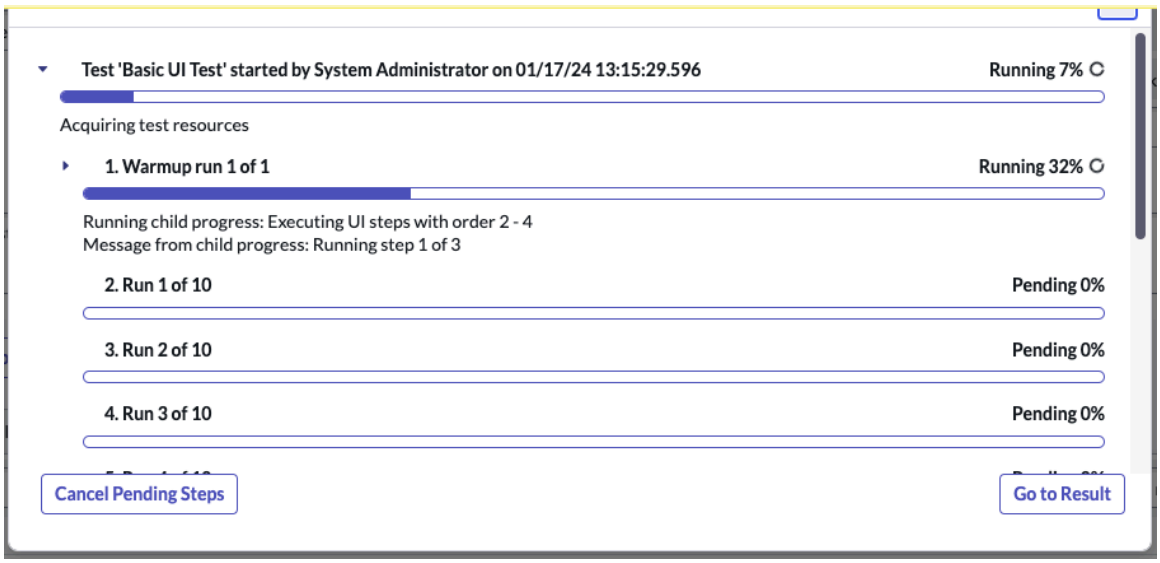
Performance profiling

Performance profiling allows you to do performance testing on your instances.

Apart from being a functional testing medium that ensures nothing breaks when any changes are made to an instance, ATF can also detect performance degradation during upgrades. You can know the cause, investigate, and fix the performance issue.

You can execute performance profiling on any of your ATF tests or suites. For each test or suite, the default test run is 10. In a suite, each test runs 10 times sequentially. The first run is a warmup that helps in warming up the cache values and is not counted to the 10 test runs of a test or suite.

Note: Performance profiling can't be done on more than 1 test simultaneously. It doesn't support parallel performance assessment.



Note: When you execute a performance test run, the system pauses and waits for any ongoing jobs to finish before starting the test run. This helps in avoiding any slowdown of your instance.

Execute performance profiling

Execute performance profiling on a test or a suite for performance testing on your instance. You can also detect performance degradation when you upgrade your instance and then investigate and fix the issues.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin
Role required: admin

Procedure
1. Navigate to **All** > **Automated Test Framework (ATF)** > **Test**.

   Note: You can also select the Suites module if you want to execute performance profiling on a suite.

   A list of tests or suites shows up.
2. Select the test you want to execute performance profiling on.
3. Select **Run Performance Test** on the Test form.
   A **Run Test** modal shows up. The modal shows the warmup run and all the 10 test runs of the selected test or suite. You can track all the test runs as they start executing.

   Note: You can select **Run Performance Suite** if you have selected a suite for performance profiling.

4. Scroll down and select the **Performance Test Results** related list.
5. Select the required performance number to view the results of all the test runs within the selected performance test.
   The Performance Run form shows up. You can also directly navigate to **All** > **Automated Test Framework (ATF)** > **Test** > **Performance Profiling** > **Performance Runs** to see the Performance Run form. Add a screenshot

   Note: You can view the test name, status of the test runs, and the duration of the test run. If the **Is warmup** value is true, its the first test run that is done to warmup the cache values. The rest of the test runs will have **Is warmup** value set to false. The warmup test is not counted as one of the 10 test runs for each test or suite.

6. Navigate to **All** > **Automated Test Framework (ATF)** > **Test** > **Performance Profiling** > **Performance Comparisons**.
   The Performance Comparisons form shows up. <add a screenshot of the form>
7. Select the first and second run to view a comparison result.
   Add the complete screenshot. You can view information like performance means and the delta mean values.

   Note: You can compare only 2 test runs at one point of time. In the next comparison, you can consider one of the previous test run as the baseline to compare with the next test run.

ATF Health Check

ATF Health Check detects ATF misconfiguration that may lead to support cases by scanning your instance for 14 known issues and providing findings you can resolve proactively.

ATF Health Check is a set of Instance Scan checks for the Automated Test Framework (ATF). It inspects your instance's ATF configuration, tests, and step definitions for a known list of configurations and deviations from general guidelines. Each issue is reported as a finding you can act on.

The checks were built directly from the issues seen most often in ATF-related support cases. Examples include `gs.sleep()` calls left in test scripts, tests that modify large numbers of existing records, ATF Debug mode left enabled, and an invalid ATF Cloud User. Running the scan

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

lets you confirm whether any of those issues are present on your instance, so you can resolve them proactively.

Prerequisites

The following prerequisites must be met to use ATF Health Check:

- The base Automated Test Framework plugin (`com.glide.automated_testing_framework`) and the Instance Scan plugin (`com.glide.instance_scan`) must be active
- You need the `admin` role to execute the scan suites
- The `admin` role is also required to change the configurable modified-records threshold
- The four checks in the ATF Test Generator and Cloud Runner suite run only if the ATF Test Generator and Cloud Runner Store app (`sn_atf_tg`) is active. Otherwise, they are skipped automatically

Terminology

Instance Scan

The ServiceNow platform capability that runs configurable checks against an instance and reports the results as findings

Suite

A named, organized group of related checks that can be run together

Check

A single, specific test performed during a scan—for example, "is ATF Debug mode enabled?"

Finding

A flagged result produced when a check's condition is met—that is, an issue was detected

What's Included

ATF Health Check adds one parent suite with two child suites underneath it:

ATF Health Check Suites (parent): Root container for the following child suites. Run this suite to execute every ATF Health Check scan in one pass

- ATF Core (10 checks): Covers core ATF configuration, test design, and performance. Available on any instance with the base ATF plugin active. Organized into Performance checks (7) and Upgradability checks (3)
- ATF Test Generator and Cloud Runner (4 checks): Covers the ATF Test Generator and Cloud Runner (TG/CR) store app. These checks only execute if `sn_atf_tg` is active. Organized into Manageability checks (3) and Upgradability checks (1)

Each finding links directly to the affected record, so you can navigate and resolve issues immediately.

Check categories

Checks are organized by category to help you prioritize and understand the scope of each issue:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Performance: Identifies configurations that slow down test execution, such as debug mode, fixed waits, excessive screenshots, or tests that modify too many records
- Upgradability: Verifies that ATF data, configurations, and tests follow best practices that support clean instance cloning and upgrades
- Manageability: Validates that infrastructure and integrations (mutual authentication, Cloud Runner app, ADCV2 configuration) are correctly configured for Test Generator and Cloud Runner

Related topics

ATF administration overview

Optimizing ATF performance

ATF test suites overview

Run an ATF Health Check scan

Execute an ATF Health Check scan to detect configuration issues on your instance. You can run a full suite of all 14 checks, run only core checks, run checks for ATF Test Generator and Cloud Runner, or run a single check.

Before you begin

Role required: admin

About this task

Before running an ATF Health Check scan, verify the following:

- The base Automated Test Framework plugin (`com.glide.automated_testing_framework`) is active
- The Instance Scan plugin (`com.glide.instance_scan`) is active
- The ATF Health Check plugin (`com.glide.automated_testing_impl.atf_health_check`) is installed

For checks in the ATF Test Generator and Cloud Runner suite to execute, the ATF Test Generator and Cloud Runner store app (`sn_atf_tg`) must be active. If the app is not active, those checks are skipped automatically.

Procedure

1. Navigate to **Instance Scans** > **Suites**.
2. Open one of the following suites:
   - ATF Health Check Suites — Run all 14 checks (both ATF Core and ATF Test Generator and Cloud Runner checks)
   - ATF Core — Run only the 10 core ATF checks
   - ATF Test Generator and Cloud Runner — Run only the 4 checks for the Test Generator and Cloud Runner store app
3. Select **Execute Suite Scan**.
4. Wait for the scan to complete.

   Note: Scan time depends on the size of your instance and the number of checks being run.

5. Open the scan results to view the status of each check.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The results show whether each check passed and how many findings it produced.

6. For each check with findings, open the check record to review its individual findings. Each finding links directly to the affected record, such as the specific ATF test, step, step config, or property. You can navigate to the record and resolve the issue.

> **Note:** To run a single check, open the check record and select **Test Check** or **Run Point Scan**. You can also schedule the suite using Instance Scan scheduling options.

Result
The scan completes and displays findings for any issues detected on your instance. Each finding includes:

- The check that detected the issue
- The specific record affected (test, step, property, etc.)
- Details about the issue

Navigate to each affected record using the links in the findings to review and resolve the configuration issue.

ATF Health Check checks and configuration

Look up all 14 ATF Health Check scans, understand what each check detects, and configure the modified-records threshold to fit your environment.

ATF Health Check checks

ATF Health Check scans your instance for 14 common configuration issues organized across two suites:

- ATF Core — 10 checks for core ATF features (performance and upgradability)
- ATF Test Generator and Cloud Runner — 4 checks for the ATF Test Generator and Cloud Runner store app (manageability and upgradability)

Each check scans your instance and flags findings when issues are detected. Each finding links directly to the affected record so you can navigate and resolve the issue.

ATF Core (10 checks)

These checks detect common performance and upgradability issues in your core ATF configuration. All checks are either Table Check or Script Only Check types and flag findings when conditions aren't met.

ATF Core Suite Checks

Check Name	Type	What It Scans	Category	Finding Details
ATF Debug mode is not enabled	Table Check	System property: `sn_atf.debug`	Performance	Flags if property value is `true` (debug mode is enabled, reducing performance)
Avoid use of gs.sleep() in Run Server Side Script steps	Table Check	Table: `sys_atf_step` filtered by step	Performance	Flags each step containing non-commented

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ATF Core Suite Checks (continued)

Check Name	Type	What It Scans	Category	Finding Details
		config = Run Server Side Script		`gs.sleep(` calls in the script
Avoid use of gs.sleep() in Custom Step Configs	Table Check	Table: `sys_atf_step_config`, field: `step_execution_generator`	Performance	Flags each custom step config containing non-commented `gs.sleep(` calls
ATF Tests with too many modified records	Table Check	Table: `sys_atf_test` (active=true), aggregates `sys_atf_modified_record` count per test	Performance	Flags tests where modified record count exceeds the configurable threshold (default: 80)
Single record modified by too many ATF Tests	Script Only Check	Aggregates `sys_atf_modified_record` grouped by record reference	Performance	Flags any single record modified by more than the threshold number of tests (default: 80)
ATF Glide Screenshots are not disabled	Table Check	System property: `sn_atf.screenshots.use_glide_screenshot`	Performance	Flags if property value is not `true` (Glide screenshots are disabled, impacting test capability)
ATF Full page screenshots are not enabled	Table Check	System property: `sn_atf.screenshots.capture_full_page`	Performance	Flags if property value is `true` (full page screenshots are enabled, reducing performance)
Do not exclude necessary ATF Tables during cloning	Table Check	Table: `clone_data_exclude`	Upgradability	Flags each of the 21 required ATF tables found in the clone exclusion list (e.g., `sys_atf_test`, `sys_atf_step`, `sys_atf_step_config`, etc.)
Reopen submitted form before UI validations	Table Check	Table: `sys_atf_test` (active=true), sequences test steps by order	Upgradability	Flags if a UI validation step runs when no form is open (i.e., after a form

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ATF Core Suite Checks (continued)

Check Name	Type	What It Scans	Category	Finding Details
				submit without re-opening the form)
ATF Tests start by creating or impersonating user	Table Check	Table: `sys_atf_test` (active=true), checks the first step (step.order limit 1)	Upgradability	Flags tests where the first step is neither an Impersonate step nor a Create User step with impersonate=true

ATF Test Generator and Cloud Runner (4 checks)

These checks validate the configuration and health of the ATF Test Generator and Cloud Runner store app. All checks only execute if plugin `sn_atf_tg` is active; if not active, the checks are automatically skipped.

ATF Test Generator and Cloud Runner Suite Checks

Check Name	Type	What It Scans	Category	Finding Details
Mutual Auth plugin is active	Script Only Check	Plugin: `com.glide.auth.mutual`, Property: `glide.authenticate.mutual.enabled`	Manageability	Flags if the Mutual Auth plugin is not active OR if the property is not `true`
Latest ATF Test Generator and Cloud Runner app is installed	Script Only Check	Table: `sys_store_app`, scope: `sn_atf_tg`	Manageability	Flags if the installed version of the Test Generator and Cloud Runner Store app is not the latest available version
ATF Instance has correct ADCV2 configuration	Script Only Check	REST request to: `{instanceURL}/adcv2/supports_tls`	Manageability	Flags if response status ≠ 200. Also flags if response body does not contain `true` (inbound TLS not enabled). Also flags if response header `server` does not contain `snow_adc` (ADCV2 load balancer not in use).
ATF Cloud User is set to valid user	Script Only Check	System property: `sn_atf_tg.username`	Upgradability	Flags if the cloud user does

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ATF Test Generator and Cloud Runner Suite Checks (continued)

Check Name	Type	What It Scans	Category	Finding Details
		to get configured cloud user; validates 9 user conditions		not exist, is inactive, is locked out, is web-services-only, is an internal integration user, has concurrent session limits, or needs password reset. Also flags if the user has `snc_read_only` role, lacks `admin` role, or has recent syslog errors.

Configuring the modified-records threshold

Two checks in the ATF Core suite measure how many records are modified by ATF tests:

- ATF Tests with too many modified records: flags individual tests that modify too many records
- Single record modified by too many ATF Tests: flags individual records modified by too many tests

Both checks use the same `sys_properties_list.do` configurable threshold system property.

Property	Value
Property Name	`sn_atf.scan.max_records_modified_by_test`
Type	Integer
Default Value	80
Read/Write Role	`admin`
Purpose	Sets the maximum threshold for the number of records modified by a single ATF test, or the maximum number of tests that can modify a single record. When either limit is exceeded, a finding is flagged by the respective check. For example, raise the threshold if 80 is too strict for your test suite design; lower it if you want scans to flag shared-data risk earlier.

Resolving ATF Health Check issues

Review the following information to resolve the issues for ATF Health Check:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Issue	What it means	Resolution options
No findings appear when I run a scan	The scan ran successfully but detected no issues. Your ATF configuration meets the health check standards.	Check that all checks have a Pass status in the scan results
I ran the scan and got zero findings for the Test Generator and Cloud Runner suite	Checks for ATF Test Generator and Cloud Runner only run when the store app is active. If the app is not installed, an empty result for that suite is expected.	You can install and activate the ATF Test Generator and Cloud Runner store app to see the findings
Can a check's script or condition be edited	No, all checks are delivered as read-only (sys_policy = read).	If a check doesn't fit your environment, use Instance Scan's standard exclusion options to skip it in your scan runs.
Do these checks change anything on my instance?	No, every ATF Health Check only reads configuration and data to produce findings; none of them modify records.	You can manually resolve a finding.
Some checks are skipped or show "Not Applicable	The checks for ATF Test Generator and Cloud Runner suite only run if `sn_atf_tg` plugin is active. If the plugin is not active, those 4 checks are automatically skipped.	You can install and activate the ATF Test Generator and Cloud Runner store app plugin to execute its checks
Findings for "too many modified records"	Either a single test is modifying more than the threshold number of records, OR a single record is being modified by more than the threshold number of tests.	**1.** Open the check finding to see which test or record is affected **2.** Review the specific test(s) or record(s) mentioned in the finding **3.** Consider refactoring the test(s) to modify fewer records, or decreasing the number of tests that modify that specific record **4.** Alternatively, adjust the threshold property `sn_atf.scan.max_records_modifie` if your use case requires higher limits
Findings for excluded ATF tables during cloning	One or more essential ATF tables have been excluded from your clone configuration, which could cause issues when upgrading or cloning your instance.	**1.** Open the check finding to identify which ATF tables are excluded **2.** Navigate to Clone Configuration Exclusions

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Issue	What it means	Resolution options
		3. Remove the excluded ATF tables from the exclusion list 4. Test your clone process after removing the exclusions
Findings for "Reopen submitted form before UI validations"	One of your ATF tests has a UI validation step that runs after a form has been submitted without being re-opened. This can cause the step to fail because the form is no longer open.	1. Open the flagged test from the finding link 2. Review the test steps in sequence to identify where the form is submitted 3. After the Submit Form or Submit Record Producer step, add a step to reopen the form (for example, Open Existing Record, Open New Form, Navigate Module, etc.) 4. Then add your UI validation step
Findings for "ATF Cloud User is set to valid user"	The cloud user configured in property `sn_atf_tg.username` does not meet one or more of the 9 required conditions for a valid ATF Cloud User.	1. Open the finding to see which conditions aren't met 2. Verify the user account exists and is active in your instance 3. Confirm the user has the `admin` role and does NOT have the `snc_read_only` role 4. Check that the user is not locked, does not need password reset, and has no concurrent session limits 5. Review syslog for any errors associated with this user 6. If the user can't be fixed, configure a different valid user in property `sn_atf_tg.username`
Findings for "ATF Instance has correct ADCV2 configuration"	Your instance does not have the correct ADCV2 configuration for the ATF Test Generator and Cloud Runner to function properly.	1. Open the finding to see which specific condition(s) aren't met 2. Verify that your instance is behind an ADCV2 load balancer (ServiceNow

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Issue	What it means	Resolution options
		ADCV2 service must be active) 3. Verify inbound TLS is enabled for your instance 4. Contact ServiceNow support if ADCV2 configuration issues persist

Administering the Automated Test Framework (ATF)

Enable or disable the Automated Test Framework, modify retention policies, move tests between instances, control user access to the Automated Test Framework, and create custom test step configurations and step environments.

For details about individual properties that control how the Automated Test Framework works, see Properties.

Creating custom test step configurations

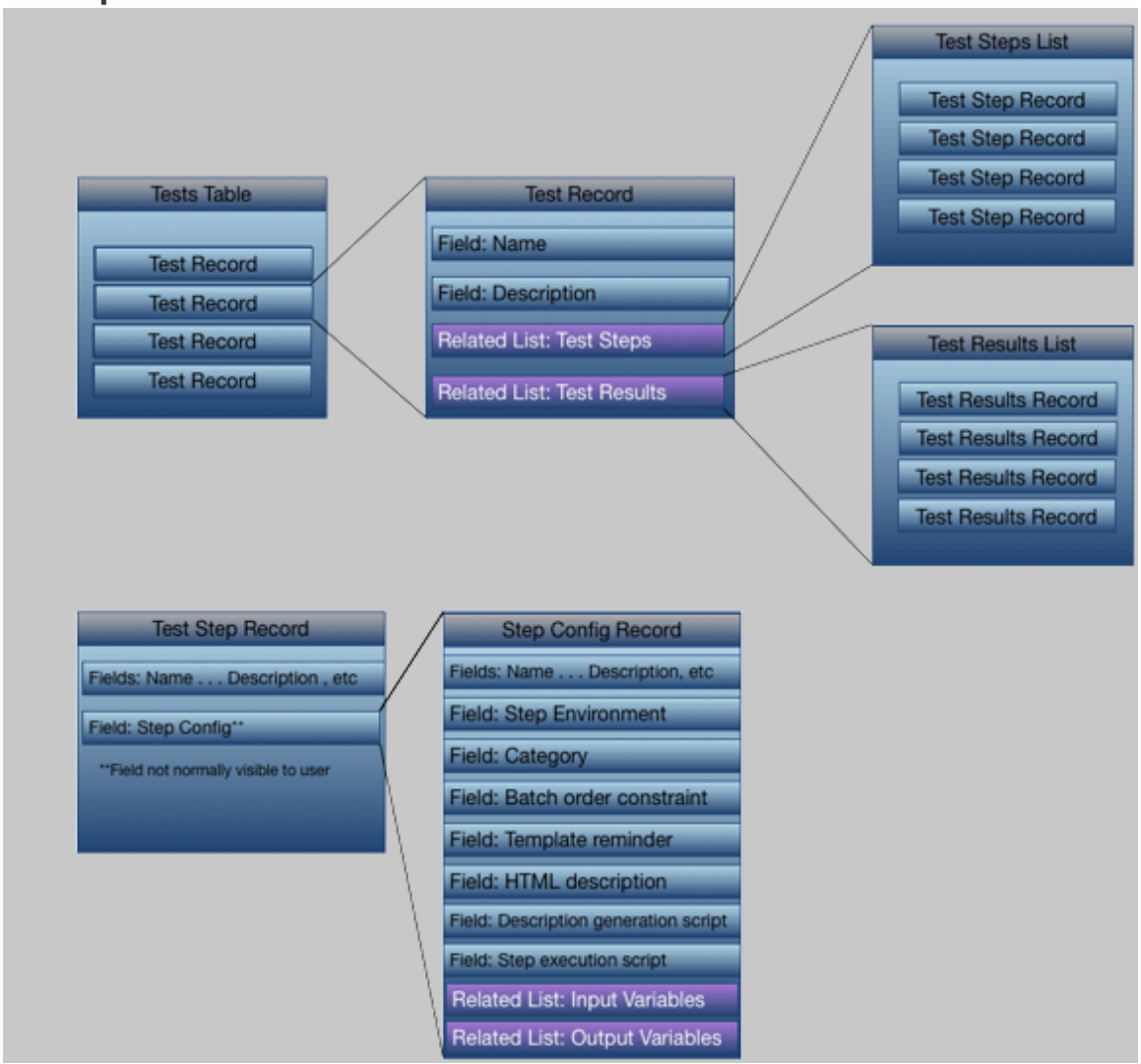
Step configuration records (or step configs) define how each step type behaves. You can create new step configurations that define custom steps that run on the server.

When you add a step to an automated test, that step has a defined type such as Set Field Values, or Record Insert. Each step configuration has a number of characteristics that affect how steps of that type behave, including the inputs required, the actions performed, and so on.

These characteristics are defined in the step's step configuration record.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Test step architecture



The Step execution script field determines the actions the system executes when a step with this config runs.

Related topics

- Step configurations
- Step execution scripts

Create custom step configuration

Create a custom step configuration that can form the basis of new steps that run on the server.

Before you begin

Role required: atf_test_admin

Note: Creating configurations are only supported in the server steps, the UI steps (running in the browser) are not supported. Also, only scripted steps are supported, custom Java-based steps are not supported.

About this task

The Automated Test Framework includes specific types of steps such as Open Form, Set Value, Assert Value and so forth. With the **Step Configurations** module, you can create steps that perform actions you specify. You can only create configurations for steps that run on the Server. You cannot create configurations for steps that run on the browser.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Administration** > **Step Configurations**.
2. Click **New**.
   The system displays the **Test Step Config** form.
3. In the **Name** field, enter a name for your step type.
4. Leave **Active** checked.
5. Leave **Step environment** set to **Server-Independent**.
   You can define only step configurations that run on the server and not step configurations that run in the browser.
6. **Optional:** In the **Category** field, select the category to which you want to assign this step.
   Categories are used for filtering the step list in the Add Step dialog. For more information, see Category field example.
7. **Optional:** In the **Batch Order Constraint** field, choose one of the following values.
   - **None**: A step based on this configuration can appear at any point in a test.
   - **Start Batch Execution**: If this test includes a batch with this step, this step must be the first in the batch.
   - **Run in the middle of an execution**: If a test includes a batch with this step, this step must appear after the first and before the last step of the batch.
   - **Stop Execution**: If a test includes a batch with this step, this step must be the last step in the batch.
8. In the **Order** field, enter an integer specifying where steps with this configuration appear in the step list on the Add Test Step dialog.
   For more information, see the example using the **Order** field in Order field example.
9. In the **Template reminder** field, enter the instructions you want to appear when this step is included in a test as part of a template.
   For more information, see the example of using the Template reminder field.
10. In the **HTML description** field, enter the text you want to appear when the cursor highlights this step on the Create New Step dialog.

    For more information, see the example using the HTML description field in .

    Note: The next two steps involve writing scripts, which you should wait to complete until you have added input and output variables.

11. In the **Description generation script** field, add code to the provided template to generate the description assigned to a Test Step record when a step of this type is included in a test.
    For more information about writing this script, see Step description generation script. To see an example of where the system displays this description, see Description generation script example.
12. In the **Step Execution Script** field, add code to the provided template to define the script that executes when a step of this type runs.
    The script template provides instructions and examples for working with step inputs, outputs, and step results. For more details on the step execution script, see Step execution scripts.
13. Click **Submit**.
    The system creates a new test step configuration and returns to the list of test configurations.
14. **Optional:** Add input variables to this step config.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a. Access the step config record.

b. Scroll to the Input Variables tab, then click **New**.

c. Fill in the required fields for the new variable.

d. Click **Submit**.

e. Repeat these steps until you have added all the input variables needed.

f. **Optional:** Control the order in which input variables appear on the New Step form by editing the values in the **Order** column for the Input Variables related list.

15. **Optional:** Add output variables to this step config.

a. Scroll to the Output Variables tab, then click **New**.

b. Fill in the required fields for the new variable.

c. Click **Submit**

d. Repeat these steps until you have added all the output variables needed.

Related topics

Step execution scripts

Step configurations

Automated Test Framework Step Config record

Add a predefined list of steps (template) to an automated test

Create a custom step configuration category

Create a custom step config category.

Before you begin

Role required: atf_test_admin

About this task

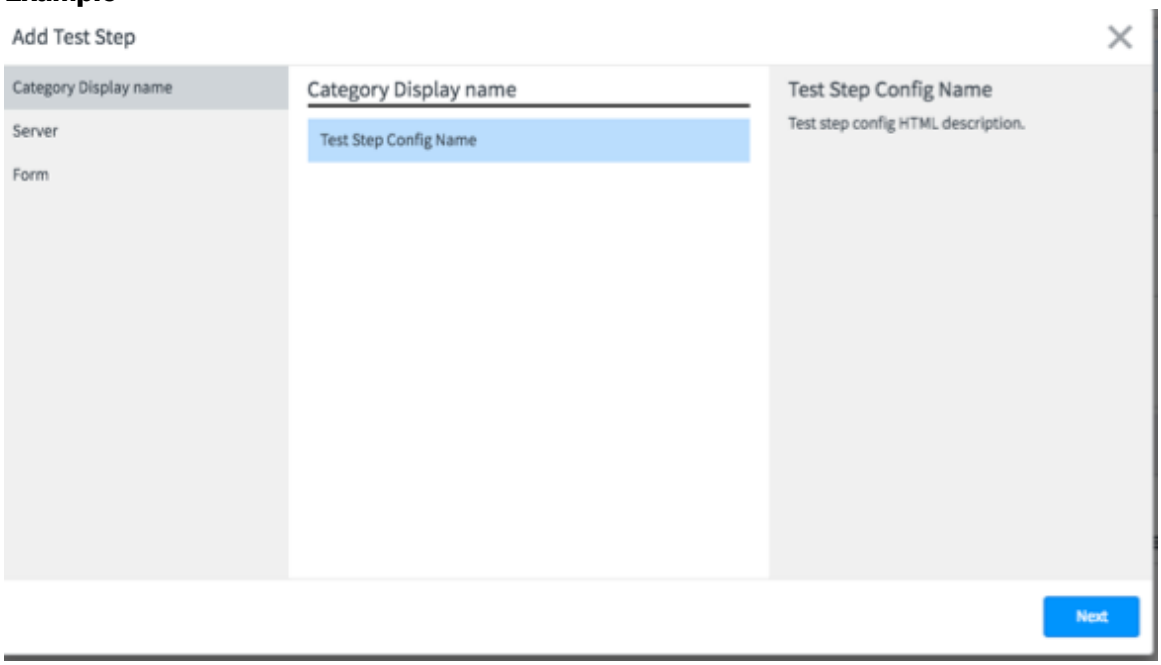
Categories are used for filtering the step list in the Add Step dialog. For more information, see Category field example.

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Administration** > **Step Configuration Categories**.
2. Click **New**.
   The system shows the **Test Step Config Category** form.
3. In the **Name** field, enter a name for your step category.
4. In the **Step Environment** field, enter the step environment in which steps under this category execute:
   - Server - Independent, if you want this category to contain steps that execute on the Server.
   - UI, if you want this category to contain steps that execute on browser.
   - Server - REST, if you want this category to contain steps that send Inbound REST messages to the instance.
5. In the **Display name** field, enter the category name you want to appear in the middle column of the Add Test Step dialog when this category is selected.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example



6. Click **Submit**.
   The system creates a test step category and returns to the list of test step environments.

Working with test step templates

Test step templates contain a list of steps to be added all at once to an automated test.

Related topics

Add a predefined list of steps (template) to an automated test

Create an automated test steps template

Reduce testing time by creating a template containing a list of steps to add all at once to an automated test.

Before you begin

Role required: atf_test_admin or atf_test_designer

About this task

Many tests follow similar patterns. One common pattern, for example, is to open a form, set some field values, validate some field values, click a UI action, open a record producer, open a catalog item, and submit the current form. If a template exists containing these steps, you can add them to a test all at once. The Automated Test Framework comes with default templates in the base system. With this procedure, you create your own templates.

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Administration** > **Test Templates**.
   The system displays the Test Templates list.
2. Click **New**.
   The system displays the Test Template form.
3. In the **Name** field, enter a name for your template.
4. In the **Test Template** field, click the lock icon ( ).
   The **Test Template** field unlocks and expands to allow editing.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

5. In the **Test Template** field, enter the name of the first test step to add to this template.
   The system adds the test step to the Test template list.
6. Continue adding test steps – in the order that you want them to appear – until you've added all the steps that you want to include in the list.

   Note: The test template supports tables, catalog items, and the record producer.

7. In the **Description** field, enter a description for this template.
8. Click **Submit**.

Related topics

Add a predefined list of steps (template) to an automated test

Edit automated test steps template

Edit an existing test template.

Before you begin
Role required: atf_test_admin and atf_test_designer

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Administration** > **Test Templates**.
   The system displays the **Test Templates** list.
2. Click the row for the test template you want to edit.
   The system displays the Test Template form.
3. In the **Test Template** field, click the lock icon ( ).
   The Test Template field unlocks and expands to allow editing.
4. **Note:** You can delete steps anywhere in the list, but you can add steps only to the end of the list. You can re-order steps in a test after you add them using the template, but you cannot re-order steps in the template itself.

   Add or delete steps to the template.
   - To delete a step, select that step, then click the **X** icon.
   - To add a step, in the **Test Template** field, enter the name of the step to add.
5. Continue adding and deleting test steps until it contains the steps you want.
6. Click **Submit**.

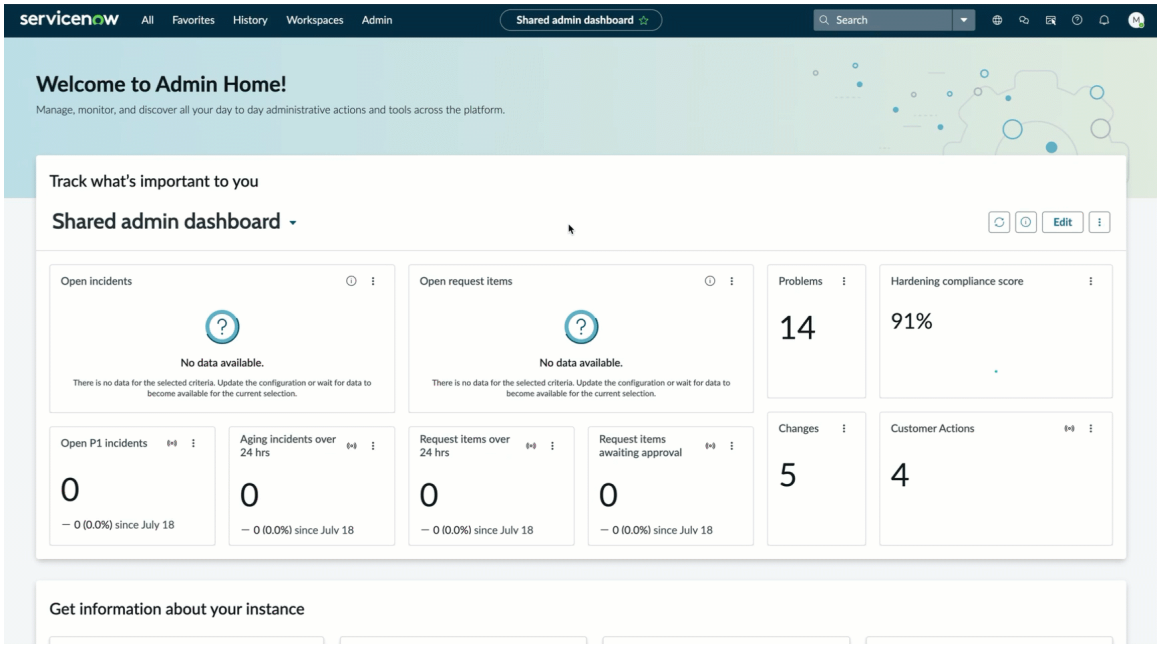
Related topics

Add a predefined list of steps (template) to an automated test

Testing Configurable Workspace components

Simplify test creation by directly interacting with components on most Configurable Workspace pages via the Page Inspector.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Create a test for Configurable Workspace interaction via Page Inspector

Leverage the Page Inspector to create tests by directly interacting with components on most Configurable Workspace pages.

Before you begin

Role required: admin

Procedure

1. Navigate to **All** > **Automated Test Framework (ATF)** > **Tests**.
   A list of ATF tests shows up.
2. Select **New** to create a new test.
3. Fill up the fields and submit it.
4. Select **Add Test Step** on the created test form.
5. Select the Configurable Workspace test step category.
6. **Optional:** Select Open Workspace Page test step.

   Note: This step is applicable if you want to navigate to a workspace page using a URL.

   a. Fill up the details on the Open Workspace Page form.
   b. Add the workspace URL you want to use in the Workspace URL field.
   c. Select **Submit** to create the test step.
7. **Optional:** Select **Add Test Step** on the created test form again and select the Configurable Workspace test step category.

   Note: This step is applicable only if you want to create another Configurable Workspace test step.

8. **Optional:** Select Test Page test step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: This step is applicable only if you want to interact with a seismic component on a workspace page. See Testable Configurable Workspace components for more information.

The Configurable Workspace Test Authoring modal shows up.

a. Enter the URL of the workspace page you want to inspect.

Note: By default, the URL selected in the Open Workspace Page test step shows up. If you haven't created an Open Workspace Page test step, you must enter the workspace URL you want to inspect.

b. Select **Open** to open the workspace page. The workspace page opens up in a new tab. You will notice a right panel that displays the components of the page you want to inspect.

c. Select REFRESH COMPONENT LIST to make sure it shows all the components on the page.

d. Drag and drop the inspector on a component on the page.

Note: The name of the selected component is displayed on the right panel. Along with that, actions related to the selected component are also displayed. Select the required action for the component. These actions are actions that are understood by the components.

e. Select the required action for the component.

f. Enter the value for any inputs and optionally enter an expected return value to assert for the selected action.

g. Select **Submit** to create the test step.

9. Continue adding Test Page steps until all required navigation and component interaction is accomplished.
Test steps can be viewed, deleted, or reordered in the dropdown under the test name or by reloading the Test Steps related list on the test form.

Note: The Test Steps related list information on the test form get updated with the previous steps.

10. **Optional:** Open a Test Page test step from the test form related list and select **Edit page interaction** to modify existing Test Page steps or resume a test authoring session.
The Page Inspector loads the workspace page from the most recent Open Workspace Page step and replay all Test Page steps in the batch. Deleting previous steps results in the reload and replay of all Test Page steps in the batch to validate a successful workspace navigation path.

Enable or disable executing Automated Test Framework tests

Allow or prevent tests and test suites from executing on this instance.

Before you begin
Role required: atf_test_admin

About this task
By default, the system property that is used to run automated tests is disabled to prevent you from accidentally running these tests on a production system. To avoid data corruption or an outage, run tests only on development, test, and other non-production instances.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Administration** > **Properties**.
2. Set the test execution property.
   - To enable test and test suite execution, select **Enable test/test suite execution**.
   - To disable test and test suite execution, clear **Enable test/test suite execution**.
3. Select **Save**.

Modify data retention policy for ATF test results

Modify the Auto Flush data retention policy, which designates how long the system retains data, and referencing data, for test and test suite results. You can change the frequency of flushing for the sys_atf_test_result or sys_atf_test_suite_result base tables. This setting controls how far back in time test result data is available.

Before you begin

Role required: atf_test_admin

About this task

The system regularly flushes data in the sys_atf_test_result and sys_atf_test_suite_result base tables, (and optionally, referencing data). By default, the system deletes test and test suite results data 30 days after creation. This task enables you to modify the Auto Flush retention policy for data stored in a specific base table (sys_atf_test_result or sys_atf_test_suite_result).

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Administration** > **Table Cleanup**.
   The system displays a list of the retention policies (Auto Flushes) it maintains for automated testing results tables.
2. Select the retention policy (sys_atf_test_result or sys_atf_test_suite_result) to modify.
   The system displays the record for this retention policy.

   Note: The **Tablename** field displays the name of the table to which the selected Auto Flush retention policy applies. Selecting another tablename in this field compromises the integrity of the Auto Flush record. Leave the tablename on existing ATF policies at the base system (default) value so it does not adversely affect ATF data retention behavior.

3. Specify how the system should determine the length of time for retention of data, and referencing data.

   a. In the **Matchfield** field, enter the field you want the system to use to monitor duration. For example, to specify that you want to delete data x amount of time after the system created it, leave **Matchfield** set to its default value of **sys_created_on**.

   b. In the **Age in seconds** field, enter the amount of time (in seconds) the system must wait before deleting the associated data and referencing data.

4. If you want to apply the policy to the specified data (for example, sys_atf_test_result), and to any data that references it, select **Cascade delete** (default value).
   Affected referencing data is stored the following tables: sys_atf_test_result_item, sys_atf_test_result_step, and sys_attachment (when table_name = sys_atf_test_result). If you want the policy to simply flush data in the selected table (for example, sys_atf_test_result), and skip flushing of the referencing data, then clear **Cascade delete**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

5. In the **Conditions** field, specify the filter conditions to use for selection of data (and optionally, referencing data) for this Auto Flush retention policy.
   The default is **Retain indefinitely is false**, because the Test results record also contains a **Retain indefinitely** check box that allows opting out of the auto flushes for specific test results.
6. Click **Update**.

Manage status and retention policies for automated test client runners

Modify how often active client test runners report in to the system and how long the system retains records for inactive client test runners.

Before you begin
Role required: atf_test_admin

About this task
When you start a client test runner, the system registers that runner as active, meaning that it is either running a test or is available to run a test. While the runner is active, it reports in to the system at a specified interval. If the runner does not report in at the expected time, the system marks the runner as inactive. After a period of time the system deletes the runner. This task enables you to modify the Automated Test Framework properties that control these intervals.

Procedure
1. Navigate to **All** > **Automated Test Framework** > **Administration** > **Properties**.
2. Navigate to the Test Runner Properties section.
3. To set the interval at which active client test runners report in to the system, enter the reporting interval in seconds in the **Test runner heartbeat interval** field.
4. To set the period of time a test runner can remain inactive before the system deletes it, enter the period of time in seconds in the **Test runner timeout** field.
5. Click **Save**.

Related topics
- Working with client test runners
- Active manual test runners
- Active scheduled test runners

Moving automated tests from one instance to another

Move automated tests from one instance to another using the normal process for update sets.

Role required: atf_test_admin

You can move automated tests, automated test suites, and related data using update sets. For more information, see System update sets.

Compare results and execution times for different automated test and suite results

You can compare execution times for different runs of an automated test or automated test suite. You can also compare results over time for a single automated test suite.

Before you begin
Role required: atf_test_admin or atf_test_designer

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **Automated Test Framework** > **Tests**.
2. Select the row for the test whose results you want to compare.
   The system displays the Test form.
3. Navigate to the **Test Results** related list.
4. Check the rows for the test results you want to compare.
5. From the **Actions on selected rows** control, click **Compare test step results**.
   The system displays the Compare test result execution times bar graph.

Compare execution times for different runs of the same automated test suite

Compare how long it took the system to execute each test across different runs of the same test suite.

Before you begin
Role required: none.

Procedure

1. Navigate to **Automated Test Framework** > **Suites**.
2. Select the row for the test suite whose results you want to compare.
   The system displays the Test Suite form.
3. Navigate to the **Test Suite Results** related list.
4. Check the rows for the test suite results you want to compare.
5. From the **Actions on selected rows** control, click **Compare test results**.
   The system displays the Compare test result execution times bar graph.

Compare results for automated test suite runs (aging report)

Compare how many tests passed versus failed across different runs of the same test suite.

Before you begin
Role required: none.

Procedure

1. Navigate to **Automated Test Framework** > **Suites**.
2. Select the row for the test suite whose results you want to compare.
   The system displays the Test Suite form.
3. Under Related Links, click **Display aging report**.
   The system displays the Test suite aging report.

Administering REST test step configurations

Set request and response payload sizes, filter request and response headers, and create basic auth profiles.

Create a basic auth profile using the Automated Test Framework

Create basic auth profiles to specify basic authentication credentials for Send Request - Inbound test steps.

Before you begin
Role required: web_service_admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

About this task

The user name and password must be valid credentials on the instance where the tests using the profile are run.

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Tests**.
2. Select a test that uses a Send Request - Inbound step.
3. Select a Send Request - Inbound step.
4. In the **Basic authentication** field, select the hour glass to look up the available profiles.
5. On the **Basic Auth Configurations** form, select **New**.
6. In the **Name** field, enter a name for the profile.
7. In the **Username** field, enter a user name.
8. In the **Password** field, enter a password.
9. Click **Submit**.

Filter REST request and response headers

You can add a list of REST request and response headers that are not to be saved in step-result records. You can filter headers that might contain authentication credentials or other sensitive information. The phrase "Header redacted for security" is saved instead.

To specify headers to be filtered, create a system property `glide.atf.rest.log.header_blacklist` with a comma-separated list of header names to be filtered. For information on adding properties, see Add a system property.

Automated Test Framework REST properties

These properties are installed with ATF REST.

Note: To open the System Property [sys_properties] table, enter `sys_properties.list` in the navigation filter.

Property	Usage
glide.atf.rest.log.header_blacklist	A list of headers whose content is not to be added to the log, or shown on a form. The phrase 'Header redacted for security' is saved instead. • **Type**: String • **Default value**: empty • **Other possible values**: A comma-separated list of header names. • **Location**: System Property [sys_properties] table
glide.atf.rest.request_payload_max_size	The maximum size of the request payload. • **Type**: String • **Default value**: 100 Kb

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Property	Usage
	• **Maximum value**: 1024 Kb • **Location**: System Property [sys_properties] table
glide.atf.rest.response_payload_max_size	The maximum size of the response payload. • **Type**: String • **Default value**: 100 Kb • **Maximum value**: 5120 Kb • **Maximum value**: System Property [sys_properties] table

Optimizing automatic test performance

You can troubleshoot automatic test performance by inspecting system transaction log records and potentially shorten execution time by adjusting how often automatic tests capture screenshots.

Managing automatic test screenshot settings

Capturing many screenshots can impair test performance. You can control which types of screenshots the system captures to minimize this effect.

By default, the system captures a screenshot every time it executes a form test step. This information can be useful for understanding test results, but may slow down how fast the system executes the test.

You can change automatic test framework settings so that the system captures all screenshots (as it does by default), no screenshots, or just screenshots for failed test steps.

You can change these settings to affect all tests run on this instance, or to affect just the current client test runner session. To affect all tests run on this instance, set the automatic test framework property from the automatic test framework properties page. To affect just the current client test runner session, set the screenshot mode from client test runner browser window.

Set the system property to control when the Automated Test Framework captures screenshots

To control how often this instance captures screenshots for form test steps, set the screenshot capture mode on the automatic test framework properties page.

Before you begin

Role required: atf_test_admin or admin

About this task

Setting the screenshot mode from the automatic test framework properties page affects any new client test runners started on this instance. This setting does not affect currently-running test-runners.

Procedure

1. Navigate to **All** > **Automated Test Framework** > **Administration** > **Properties**.
2. Set the **Enable or disable screenshot capture** property from the drop-down menu.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- To capture screenshots for all steps, select **Enable for all steps**.
- To capture screenshots only for failed steps, select **Enable for all failed steps**.
- To capture no screenshots, select **Disable for all steps**.

3. Set the **Screenshot timeout** time interval, in seconds.
   The Client Test Run does not take a screenshot capture if the length of the attempt exceeds this value. Users should review performance settings and browser caches on affected client systems before increasing this value.
4. Click **Save**.

Set the client test-runner property to control when the Automated Test Framework captures screenshots

To control how often the current client test runner captures screenshots for form test-steps, set the screenshot capture mode on the client test runner browser window.

Before you begin
Role required: atf_test_admin or admin

About this task
Setting the screenshot mode from the client test-runner browser window affects only this client test-runner and persists until this test-runner session is closed. This setting does not affect any other running test-runners or any future test runners.

Procedure

1. From the client test-runner browser window, click the form preferences icon ().
2. Click the **Screenshot mode** option.
   - To capture screenshots for all steps, select **Enable for all steps**.
   - To capture screenshots only for failed steps, select **Enable for all failed steps**.
   - To capture no screenshots, select **Disable for all steps**.
3. Click **Save**.

View transaction data for automated test results

To help troubleshoot performance issues with automatic tests, you can inspect related records from the transactions log entry [syslog_transaction] table.

Before you begin
Role required: test_admin or admin

About this task
Note: The system may not be able to log some transactions with short durations.

Procedure

Navigate to related transaction record in the system transaction log.

- From the Step results record, view transactions in the Step transactions related list.
- From the Test Results record, view transactions in the Test Transactions related list.

Working with scheduled test suites

You can schedule a test suite to run at a specified date and time.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

To schedule a test suite, you need three components:

- a test suite record
- a schedule record specifying when you want the system to run the test suite
- a scheduled suite run record that associates the test suite to run with the schedule for running it

With this model, you can associate a schedule with many different test suites and vice-versa.

Note: You can schedule only test suites, not individual tests. Scheduled tests will only run if there is an open Scheduled Client Test Runner page matching the scheduled suite's browser conditions. Scheduled tests cannot run on a machine that is locked, powered down, or does not already have the browser open.

The watchlist on the test suite run record also allows you to specify users to receive an email when the system finishes executing the test suite run.

If the test suite contains one or more form steps (steps involving a user interface), you must ensure that a scheduled client test runner is actively running in a browser when the schedule triggers the suite run.

Note: See Browser recommendations and requirements for recommendations and requirements for running the client test runner.

For step-by-step instructions on how to schedule a test suite, see Schedule an automated test suite.

Designate users to receive email when system finishes running a scheduled test suite

You can designate users to be notified when a scheduled test suite finishes executing.

Before you begin

Role required: atf_test_admin

Procedure

1. Navigate to the scheduled test suite run record.
2. Add to the record's **watchlist** all users you want the system to notify when this scheduled test suite run completes.

Automated Test Framework scheduled test suite completed email

When the system completes executing a scheduled test suite, it sends an email to users on the Scheduled Suite Run record watchlist. This email contains information and links to further information about the Scheduled Suite Run and its results.

Note: Email functionality must be enabled in order to send an email.

Suite Stats

The Suite Stats section of the email reports the number of suites and individual tests broken down by result status: Failed (F), Error (E), Skipped (S), Canceled (C), or Passed (P). For a description of what each status means, see Test suite results record.

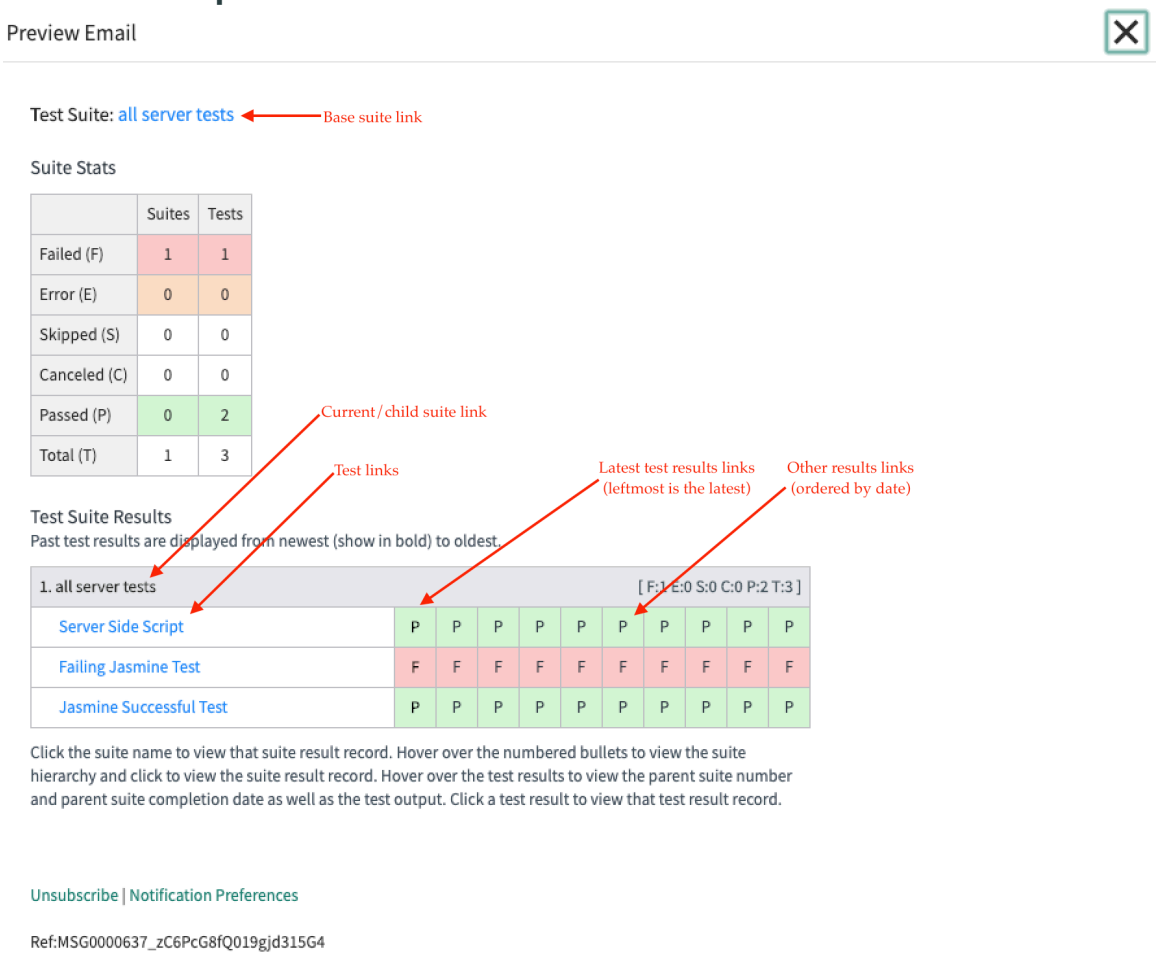
Test Suite Results

The Test Suite Results section of the email reports test suite results over time.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: By default, this report includes only suites with failed tests, but you can change this setting with the **Email properties** field on the Properties page. If set to default, only the failed tests within a test suite are reported.

Preview email report



Note: ATF reports only recent tests results on the email report. All the previous test results still exist if it's within the set retention time limit. See Table cleanup for more information on retention policy for ATF test results.

Test Results Key

Letter	Color	Description
F	red	Test failed
P	green	Test passed
S	white	Test skipped
C	white	Test canceled
E	orange	Test has error

Each entry acts as a link to the result record for that run. If you point the mouse to any of these entries, the system displays the parent suite number, the parent suite end time, and the test result output.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties affecting email content

On the Automated Test Framework Properties form, you can set options affecting the format and content of the email.

ATF Code Coverage

Metadata code coverage tracks which lines of code in your custom scripts are executed during ATF test runs. It helps identify untested code, validate test quality, and assess deployment risk in ReleaseOps.

Metadata code coverage is a testing and quality assurance feature that automatically collects and analyzes code coverage data when running ATF tests. It provides visibility into:

- Which lines of code in your scripts execute during test runs
- Code coverage percentages at the script level and deployment level
- Gaps in test coverage across your custom code
- Integration with test triage and test result analysis

Supported Code Scopes

Metadata code coverage can track code execution in the following scopes:

- Server-side scripts: Business rules, script includes, workflow activities, and other server-side code written in Rhino JavaScript
- Client-side scripts: Client scripts, UI policies, and other client-side code executed in the browser
- Global scope code: Custom global scope scripts and libraries
- Application-scoped code: Code within specific application scopes can be filtered and analyzed separately

Key Concepts

Line coverage tracks whether individual lines of code in a script were executed during a test run. Line-level granularity helps identify which specific code paths are exercised by your tests.

Coverage aggregation combines coverage data from multiple test runs to show cumulative coverage at the test, test suite, script, and deployment levels.

Traced records are captured during test execution with code coverage enabled. The system records which table records and metadata attributes are accessed by your scripts. This tracing data is linked to coverage records for detailed analysis.

Test result integration captures code coverage metrics at the test, test suite, and test result levels, allowing you to correlate test outcomes with the code paths they exercise.

Use Cases

Test quality assessment uses code coverage data to identify critical code paths that lack test coverage and prioritize writing additional tests for those areas.

Deployment risk analysis reviews code coverage data before deploying custom code changes to verify the modified code is adequately tested. Low coverage on recently changed code indicates higher deployment risk.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Test triage and troubleshooting uses code coverage data to support the Test Triage feature. This feature helps identify why tests fail by showing which code paths were exercised and which metadata was modified.

Deployment request assessment in ReleaseOps automatically evaluates code coverage as part of the deployment request assessment process. By default, if custom code coverage is below 70%, the deployment request transitions to the Reconciling state and a test failure task is created.

Deployment Coverage Threshold

Metadata code coverage integrates with ReleaseOps deployment assessment. By default, if custom code in a deployment request has less than 70% ATF test coverage, the following occurs:

- The deployment request transitions to the `Reconciling` state
- A test failure task is automatically created
- Deployment is blocked pending review and potential remediation

You can adjust the coverage threshold in the deployment request assessment playbook. See Set Automated Test Framework (ATF) code coverage for configuration details.

Limitations and Considerations

When working with metadata code coverage, be aware of the following:

- Code coverage collection may impact test performance. The system includes thresholds to prevent excessive data accumulation.
- Code coverage can be disabled globally or at the script level if performance becomes a concern.
- Data truncation: Large API request payloads may be truncated when persisted. By default, fields are limited to 4,000 characters. When truncation occurs, a `was_truncated` flag is set to `true` in the API response, with details provided in the `truncation_message` field. You can adjust the default limit using the `glide.db.truncate_utf8` system property.
- Coverage data is subject to database field length constraints. Fields affected include test_suite_results, test_results, and metadata_info JSON arrays.
- Code coverage only includes code executed by the server-side Rhino engine or browser; static metadata, configurations, and non-executable records are not covered.
- When tracing is interrupted due to threshold constraints, an indicator is recorded on the test result to show that coverage data is incomplete.

Availability

Metadata code coverage is available starting with the Australia release.

Programmatic Access

You can retrieve and analyze code coverage data programmatically using the ATF Code Coverage REST API. The API provides three main endpoints for computing code coverage:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- `POST /api/now/atf/code_coverage/all` — Aggregates coverage across all scripts covered by specified test suite or test runs
- `POST /api/now/atf/code_coverage/by_script_id` — Computes coverage for specific script records
- `POST /api/now/atf/code_coverage/by_line_number` — Computes coverage using filtered line numbers per metadata record

Each endpoint returns coverage percentages at the script and deployment levels, along with metadata about the request and any data truncation warnings. For detailed API specifications, parameter documentation, and code examples, see ATF Code Coverage API.

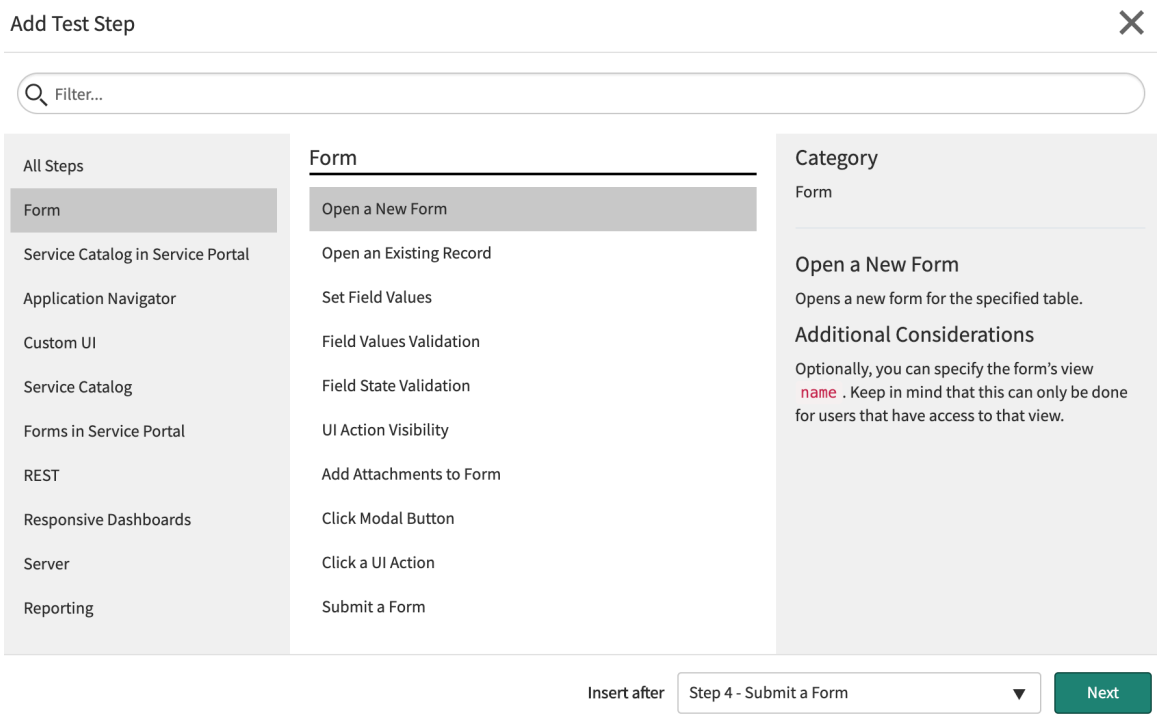
Related Features

- Automated Test Framework (ATF): The testing framework that generates code coverage data during test execution
- Test Triage: Analyzes test failures using code coverage and metadata tracing to identify root causes
- ReleaseOps: Integrates code coverage assessment into deployment request workflows
- Deployment Analyzer: Uses code coverage data to assess deployment readiness and risk

Automated Test Framework (ATF) test step categories

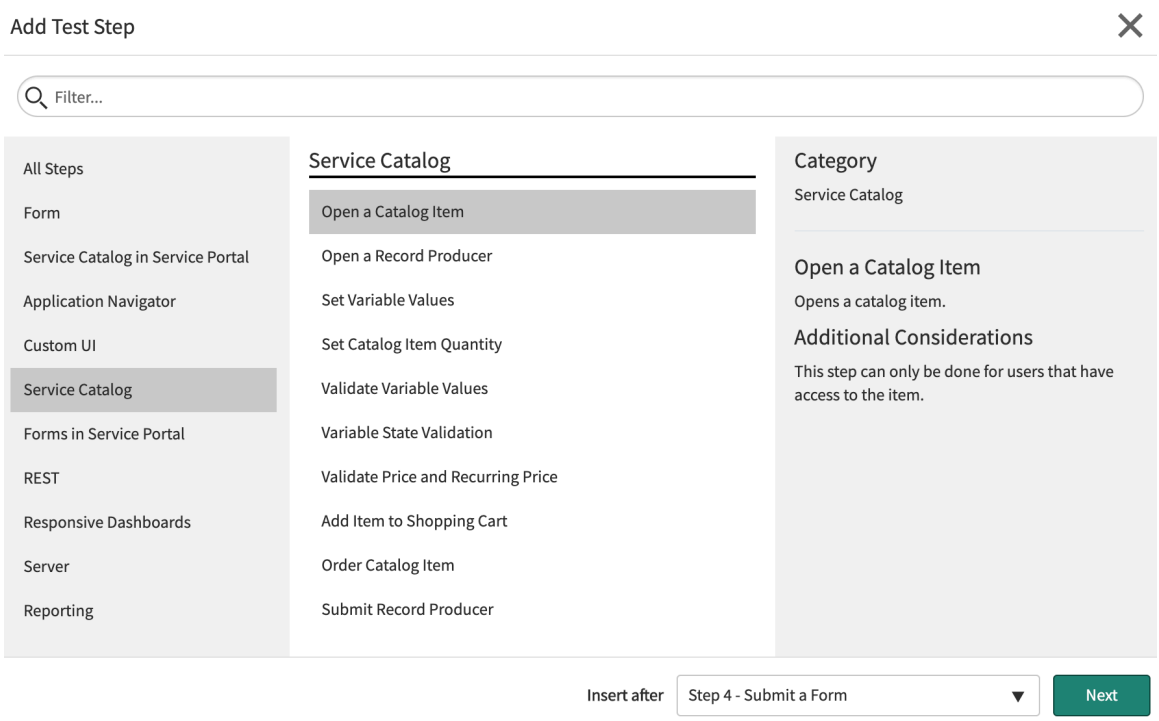
Find test steps for a particular user interface or ServiceNow AI Platform feature.

The Add Test Step panel lists test categories in the left pane. You can directly search for a specific test step by its name in the search field at the top of the panel. You can also select **All Steps** in the left pane to list all available test steps in the middle pane. A description of the selected step appears in the right pane.



You can select a specific test category to list only the test steps available for that test category in the middle panel. A description of the selected step appears in the right panel.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Reusable Tests category

Create reusable test components that can be incorporated into various other tests (regular or reusable test), minimizing redundant test code and enhancing test maintainability.

The Reusable Tests test step category contains all the reusable tests created on the instance by default. When you add a test step, select **Add Test Step**. You can now invoke the reusable tests you have created through the Reusable Tests test step category.

Note: By default, the reusable tests show up in this category. If you have customized a reusable test for another category, it shows up under the selected test step category.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Read-only application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to. The test can be a regular or reusable test.
Timeout	Amount of time to wait for the step to pass.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
Reusable Test	Read-only name of the reusable test that you're adding.
Notes	Description of the test step.

See Reusable tests for more details.

Service Catalog in Service Portal category

Validate catalog item transactions and requester flows from Service Portal.

Activation of the Automated Test Framework for Service Catalog in Service Portal

These ATF test steps require activation of the Automated Test Framework Service Catalog Service Portal (com.glide.automated_testing_impl.service_catalog_portal) plugin, which is active by default on new instances. Administrators may need to activate the plugin on instances upgraded from earlier versions.

Support for parametrized tests

Service Catalog in Service Portal step configurations support parametrized tests. For more information on parametrized tests, refer to Parameterized tests.

Open a Record Producer (SP)

Open a record producer in the Service Portal.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Portal	Portal for which you want to test this step.
Page	Service Catalog page associated with the test step.
Record Producer	Record producer that you want to open.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
	Note: You should have access to the record producer.
Query Parameters	URL query parameters for the page, such as *sys_id*.

Open a Catalog Item (SP)

Open a catalog item in the Service Portal.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Portal	Portal for which you want to test this step.
Page	Service Catalog page associated with the test step.
Catalog Item	Catalog item that you want to open. **Note:** You should have access to this catalog item.
Query Parameters	URL query parameters for the page, such as *sys_id*.

Open an Order Guide (SP)

Open an order guide in the Service Portal.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
	can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Portal	Portal for which you want to test this step.
Page	Service Catalog page associated with the test step.
Order Guide	Order guide that you want to open. **Note:** You should have access to this order guide.
Query Parameters	URL query parameters for the page, such as *sys_id*.

Add row to multi-row variable set (SP)

Add a row to a multi-row variable set included in the current catalog item in Service Portal. You can use this step configuration only when the current catalog item contains a multi-row variable set.

Use this step after the Open a Catalog Item(SP) step, Open a Record Producer(SP) step, or Open an Order Guide (SP) step. If a contextual value has been used for the Open a Catalog Item(SP) step, Open a Record Producer(SP) step, or Open an Order Guide (SP) step, set the catalog item in this step.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Notes	Notes about the test step.
Catalog Item	Catalog item whose multi-row variable set requires an addition of a row. If an item is not already selected, you can either search for the item, or insert a reference to the contextual value of the item from a list of available parameters.
Multi-Row Variable Set	Multi-row variable set for which a row should be added.

Save current row of multi-row variable set (SP)

Save the current row of a multi-row variable set included in the current catalog item in Service Portal. You can use this step configuration only when the current catalog item contains a multi-row variable set. Use this step after the Add row to multi-row variable set (SP) step.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Notes	Notes about the test step.
Assert Type	Criteria for the test to pass. **Successfully saved row to a multi-row variable set** Test passes only if the current row of the multi-row variable is saved. **Cannot save a row to multi-row variable set** Test passes only if the current row of the multi-row variable is not saved.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Set Variable Values (SP)

Set variable values for a catalog item or record producer in the Service Portal. For a catalog item, use this step after using the Open a Catalog Item (SP) step, and before using the Order Catalog Item (SP) step.

For a record producer, use this step after using the Open a Record Producer (SP) step, and before using the Submit Record Producer (SP) step.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step. **Note:** Use the condition builder to set the field value. The condition builder displays an appropriate control for the field data type. For example, a reference field displays a **Lookup record** control.
Item	Catalog item or record producer for which you want to set variable values.
Multi-Row Variable Set	Multi-row variable set for which variable values should be set. **Note:** Use the Add row to multi-row variable set (SP) step configuration prior to the current step configuration.
Variable Values	List of variables and the values that you want to set for them. **Note:** You can set the value for multiple variables.

Note: Custom variables and custom variable with labels are not supported for Set Variable Values step configuration.

Validate Variable Values (SP)

Validates variable values of a catalog item or record producer in Service Portal. For a catalog item, use this step after using the Open a Catalog Item (SP) step, and before using the Order Catalog Item (SP) step.

For a record producer, use this step after using the Open a Record Producer (SP) step, and before using the Submit Record Producer (SP) step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step. **Note:** Use the condition builder to set the field value. The condition builder displays an appropriate control for the field data type. For example, a reference field displays a **Lookup record** control.
Item	Catalog item or record producer whose variables should be validated.
Multi-Row Variable Set	Multi-row variable set for which variable values should be validated. **Note:** Use the Add row to multi-row variable set (SP) step configuration prior to the current step configuration.
Catalog Conditions	Conditions for variable validation. If the conditions are met, the test passes. **Note:** The label of a variable associated with a variable set reflects the variable set name. The format is *variable_set_name » variable_name*.

Note: Custom variables and custom variable with labels are not supported for Validate Variable Values step configuration.

Variable State Validation (SP)

Validates the state of variables in Service Portal. Possible variable states are mandatory, not mandatory, read only, not read only, visible, and not visible.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
	As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Item	Catalog item or record producer whose variables should be validated.
Multi-Row Variable Set	Multi-row variable set for which variable states should be validated. **Note:** Use the Add row to multi-row variable set (SP) step configuration prior to the current step configuration.
Visible	List of the catalog item variables that must be visible for the step to pass.
Not visible	List of the catalog item variables that must be hidden for the step to pass.
Read only	List of the catalog item variables that must be read-only for the step to pass.
Not read only	List of the catalog item variables that must not be read-only for the step to pass.
Mandatory	List of the catalog item variables that must be mandatory for the step to pass.
Not mandatory	List of the catalog item variables that must not be mandatory for the step to pass.

Note: Custom variables and custom variable with labels are not supported for Variable State Validation step configuration.

Validate Price and Recurring Price (SP)

Validate the price and recurring price of a catalog item in Service Portal. Use this step after using the Open a Catalog Item (SP) step, and before using the Order Catalog Item (SP) step. This step is not applicable for a record producer.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
	As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Price	Price of the catalog item for the step to pass.
Recurring price	Recurring price of the catalog item for the step to pass.
Recurring price frequency	Recurring price frequency of the catalog item for the step to pass.

Navigate within Order Guide (SP)

Navigate within an order guide.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
Notes	Notes about the test step.
Guide Step	Order guide step to which you want to navigate: • **Describe Needs** • **Choose Options** • **Summary**
Assert Type	Criteria for the test to pass. **Navigation Successful** Test passes only if the navigation is successful. **Navigation Failed** Test passes only if the navigation fails.

Set Catalog Item Quantity (SP)

Set the quantity for a catalog item in Service Portal. This step is not applicable for a record producer. Use this step after using the Open a Catalog Item (SP) step, and before using the Order Catalog Item (SP) step.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Quantity	Quantity of the catalog item that you want to order.

Validate Order Guide Items (SP)

Validate items included in the order guide.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Catalog Items	Catalog items that you want to validate. **Note:** You should have access to these catalog items.

Review Order Guide Summary (SP)

Review the order guide summary in the Service Portal.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Catalog Items	Catalog items that you want to review. **Note:** You should have access to these catalog items.
Price	Price of the catalog item for the step to pass.

Review Item in Order Guide (SP)

Review individual items in the order guide and choose whether or not to include the item.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
Notes	Notes about the test step.
Catalog Item	Catalog item that you want to review. **Note:** You should have access to this catalog item.
Included	Selected if the catalog item should be included in the order guide, otherwise unselected.

Add Item to Shopping Cart (SP)

Add the current catalog item to the shopping cart in Service Portal. Use this step after using the Open a Catalog Item (SP) step.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Assert Type	Criteria for the test to pass. **Successfully added item to Shopping Cart** Test passes only if the catalog item is successfully added to the shopping cart. **Cannot add item to Shopping Cart** Test passes only if the catalog item cannot be added to the shopping cart.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Outputs

Field	Description
cart_item_id	The sys_id of the added catalog item.

Add Order Guide to Shopping Cart (SP)

Add an order guide to the shopping cart.

Input

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Assert Type	Criteria for the test to pass. **Cannot add order guide to shopping cart** Test passes only if the order guide cannot be added to the shopping cart. **Successfully added order guide to shopping cart** Test passes only if the order guide is successfully added to the shopping cart.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Order a Catalog Item (SP)

Click **Order Now** for the current catalog item in the Service Portal. Use this step after using the Open a Catalog Item (SP) step.

After this step, you cannot use any other steps on the catalog item. If the two-step checkout is false, a request is generated for the catalog item. If the two-step checkout is true, you are redirected to the cart preview page.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Assert type	Criteria for the test to pass. **Successfully ordered Catalog Item** Test passes only if the catalog item is successfully ordered. **Cannot order Catalog Item** Test passes only if the catalog item cannot be ordered.

Outputs

Field	Description
table	The table to which the submitted request belongs.
record_id	The sys_id of the submitted request.

Submit an Order Guide (SP)

Click **Order Now** to order an order guide. Do not add more than one record producer to the order guide.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Assert Type	Criteria for the test to pass. **Cannot order the Order Guide** Test passes only if the order guide cannot be ordered. **Successfully ordered the Order Guide** Test passes only if the order guide is successfully ordered.

Outputs

Field	Description
table	The table containing the submitted order guide.
record_id	The sys_id of the submitted order guide.

Submit Record Producer (SP)

Submit the current record producer in the Service Portal. Use this step after using the Open a Record producer (SP) step. After this step, you cannot use any other steps on the record producer.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Assert Type	Criteria for the test to pass. **Successfully submitted Record Producer** Test passes only if the record producer is submitted successfully. **Cannot submit Record Producer** Test passes only if the record producer cannot be submitted.

Outputs

Field	Description
table	The table containing target record of the record producer.

Application Navigator category

Verify the functionality of menus and modules in the application navigator.

Application Menu Visibility

Verifies the visibility, or lack thereof, of selected application menus in the application navigator (left navigation bar). For example, you create a test that first impersonates a user, then verifies that specified application menus (such as Self-Service and Reports) are visible, or are not visible, to that user.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
Active	Option to activate this test step for use.
Application	The application scope in which the system runs this test or test suite.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Navigator	Navigator version to assert against • Core UI (UI16): If you are creating new steps, you will have Core UI by default. • Next Experience: If you have Next Experience enabled, Next Experience is the default navigator. If it is disabled, Core UI is the default navigator.
Visible assert type	Specifies how application menus selected in the **Visible application menus** field should be tested for visibility in the application navigator. • **At least these application menus are visible**: At minimum, all the selected application menus are visible in the application navigator. • **Only these application menus are visible**: Only the selected application menus are visible in the application navigator.
Visible application menus	Application menus whose visibility in the application navigator is being verified.
Not visible assert type	Specifies how application menus selected in the **Not visible application menus** field should be tested for lack of visibility in the application navigator. • **At least these application menus are not visible**: At minimum, all the selected application menus are not visible in the application navigator. • **Only these application menus are not visible**: Only the selected application menus are not visible in the application navigator.
Not visible application menus	Application menus whose lack of visibility in the application navigator is being verified.

Related topics

Create an application menu

Create an application menu

Application menus define the main content that users can access in the application navigator. You can configure which applications appear in the application navigator.

Before you begin

Role required: admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

About this task

When creating an application menu, consider grouping like modules into a consolidated application menu. When editing an existing menu, add more useful modules to the application menu, and remove unneeded ones.

Procedure

1. Navigate to **All** > **System Definition** > **Application Menus**.
2. Click **New**.
3. Complete the form.

Field	Description
Title	Defines the display name of the application menu.
Roles	Restricts access to the specified roles. All users can view the application menu when it is active.
Category	Specifies the menu category that defines the navigation menu style (default value is **Custom Applications**).
Hint	Defines the text that appears in a tooltip when a user points to this application menu.
Active	Select the check box to activate the application menu. Only active application menus appear in the application navigator.
Description	Provide a more detailed explanation of what this application does.
Other fields	
Order	Defines the relative position of the application menu in the application navigator. If you do not specify an order, the default order of the menu category is used.
Default device type	This field is not used. You can define application menus for mobile devices in a separate table.

Note: You may need to configure the form to see all fields.

4. Click **Submit**.
5. Create modules to appear in the application menu.
   Only application menus that contain modules appear in the application navigator.

Create a module

Modules are the children, or the second tier navigation options to the applications in the application navigator. Modules often link to other pages or records in the platform. You can configure which modules appear in the application navigator using Application Menus module.

Watch this five-minute video to learn about adding application menus and modules to the application navigator.

Before you begin

Role required: admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Open the application menu record using one of the following methods.
   - Navigate to **System Definition** > **Application Menus** and select the application menu from the list.
   - Point to the application menu and click the edit application (pencil) icon.

   Two methods for creating a module

2. Scroll down to the **Modules** related list and click **New**.
3. Define the module by completing the fields on the Module form.

Module form

Field	Description
Title	Defines the module name. Choose a title that clearly identifies the module.
Application menu	Specifies the name of the application menu under which the module appears.
Hint	Defines the tooltip that appears when a user points to the module name. **Note:** Module Hints are deprecated in Core UI
Order	The order in which the module displays relative to other modules.

4. On the **Visibility** tab, complete the fields.

Visibility tab

Field	Description
Roles	Restricts module access to the specified roles. If this field is left blank, the module is visible to all users who have access to the application menu.
Active	Defines whether the module appears in the application navigator.
Override application menu roles	Allows users to access this module even if they do not have permission to view the containing application menu. Users must still meet the role requirements for this module.

5. On the **List Type** tab, complete the fields.

   The fields displayed depend on the **Link type** you choose. All module URIs must be encoded. If you supply arguments to the module URI, either you or ServiceNow. For more information about encoding module URIs, see Encoding module URIs.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

List Type tab

Field	Description
Link type	Specifies what type of link this module opens. You must specify additional information based on the link type. See Module link types.
Table	Specifies the table used by the module. **Note:** The list shows only tables and database views that are in the same scope as the module.
Filter	Conditions for the items presented in the module, for example, **Active is true**.
Argument	String appended to the URI to create the module URI. Can be a sysparm_query. These values must be encoded either by you or ServiceNow. For more information about encoding module URIs, see Encoding module URIs.
Order	Specifies the order in which the modules appear under the application.

Module link types

The **Link type** field on the Module form specifies what type of link the module opens.

Module link types

Link Type	Description
Assessment	Links to the assessment-based survey you select in the **Assessment** reference field. See Create a survey module.
Content Page	Displays the content page you select in the **Content page** reference field. See Create a content page.
Documentation Link	Links to a documentation page and opens in a new tab or window. This link type is used with embedded metadata in documentation topics. To open an internal document from a module, use the **URL (from Arguments)** module link type.
Homepage	Displays the homepage you select in the **Homepage** reference field.
HTML (from Arguments)	Places HTML in the application navigator. This link type is used for more complicated links, where a flat URL is not customizable enough. **Note:** • The **HTML (from Arguments)** link type is supported in the legacy UI15 and UI11 interfaces only. In Core UI, use the **URL (from Arguments)** link type instead. • Enter a value for the **Arguments** field.
List Filter	Displays an unpopulated list view for the table you select in the **Table** field. Allows users to specify a filter without loading the list first. Use the **Filter** field to define the default filter for the list. Use the **View name** field to specify a View management.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Module link types (continued)

Link Type	Description
List of Records	Displays the list view for the table you select in the **Table** field. Use the **Filter** field to define the default filter for the list. Use the **View name** field to specify a view.
Map Page	Displays the map page you select in the **Map page** reference field.
New Record	Displays a form for creating a record in the table you select in the **Table** field. • Use the **View name** field to specify a view. • Use the **Arguments** field to apply a template. See Create a module for a template .
Run a Report	Runs the saved report you select in the **Report** field.
Script (from Arguments)	Runs a script, as defined in the **Arguments** field. **Note:** Enter a value for the **Arguments** field.
Search Screen	Link that displays a blank form for searching records in the table. Use the **View name** field to specify a view. **Note:** Use the parameter *&sysparm_result_view=view_name* to define the view the results are rendered in. All searches use a **[starts with]** query to search for matching text. Other query types are not supported in search screens.
Separator	Creates a division between modules. Enter a name in the **Title** field to add a section name that users can collapse or expand.
Single Record	Displays a form for a single record on the table. Use the **View name** field to specify a view.
Survey	Links to the legacy survey you select in the **Survey** reference field. Use the **Survey overwrite** check box to determine whether the survey can be taken multiple times. **Note:** The **Survey** link type is for use with legacy surveys only, which assessment-based surveys replace. Select the **Assessment** link type to link the module to an assessment-based survey.
Timeline Page	Displays the timeline page you select in the **Timeline Page** reference field. See Timeline pages .
URL (from Arguments)	Opens any URL, as defined in the **Arguments** field. [Optional] Use the **Window name** field to define a link that opens in a new window.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Module link types (continued)

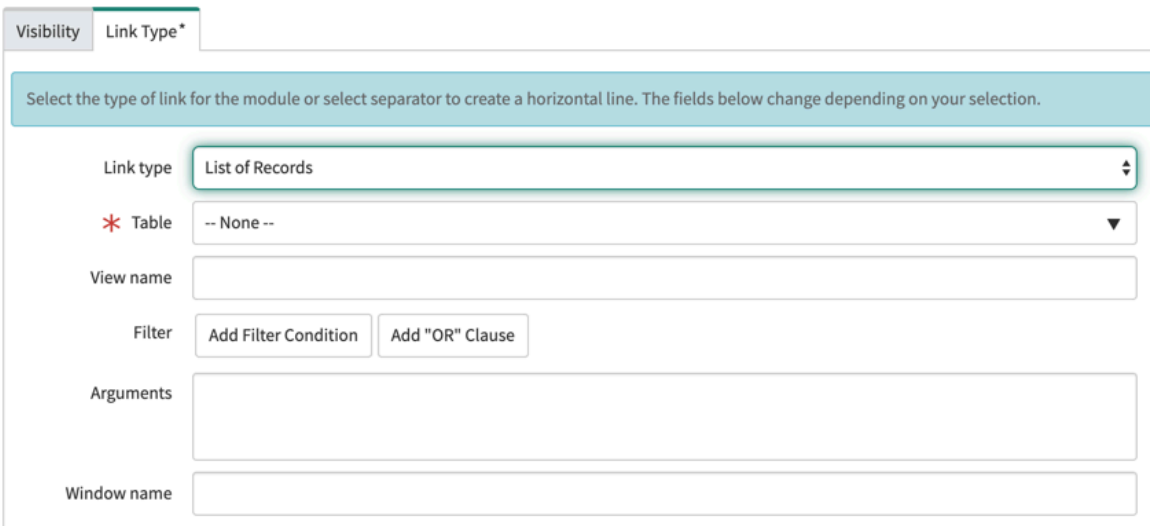
Link Type	Description
	Note: • For internal links, always use a relative link such as `./catalog_home.do?sysparm_view=catalog_default` or `catalog_home.do?sysparm_view=catalog_default`. Do not use an absolute link to a ServiceNow instance. It creates problems when you move an update set from a development instance to a production instance because the URL still references the development instance. • Enter a value for the **Arguments** field.

Encoding module URIs

Clicking a module name in the navigation pane executes a URI that opens the module's page in the content pane. All the characters in module URIs must be URL-encoded or the link breaks.

Note: If you're upgrading to the New York release or later from a pre-New-York release, your module UIs may break if they do not follow the conventions presented in this topic.

When you create modules, you have the option of adding arguments and filter conditions that sort and/or reduce the number of results displayed in the content pane. When you click a module name in **System Definitions** > **Application Definitions**, you can see those conditions and arguments on the **Link Type** tab.



The argument definition in **Arguments** and filter conditions defined in **Filter** become part of the module's URI and must be URL-encoded. ServiceNow automatically URL-encodes filter conditions and appends them to the module URI using sysparm_query. For example, adding the filter condition, **Active is true** appends `sysparm_query=active%3Dtrue` to the module's URI; %3D is the URL-encoding for the equals sign (=).

The following table shows when you must URL-encode the argument in the **Arguments** field and when ServiceNow URL-encodes the argument.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Rules for encoding arguments

Has a filter condition?	Argument definition starts with	Who encodes the argument?	How argument is handled
No	^	ServiceNow	Removes the caret (^) from the argument, encodes it and uses sysparm_query to append it to the module URI.
No	&	You	Removes the ampersand (&) from the argument and appends it to the module URI.
No	Anything else	ServiceNow	Encodes the argument and uses sysparm_query to append it to the module URI.
Yes	^	ServiceNow	URL-encodes the filter definition and the argument and uses sysparm_query to append the combination to the module URI.
Yes	Anything else	You	URL-encodes the filter definition and uses sysparm_query to append it and the (unaltered) arguments to the module URI.

You can turn on (the default) and off the URL-encoding requirement for module UIs using the *`glide.ui.encode_module_uri`* property.

Examples

The following examples demonstrate when you have to URL-encode the argument definition in **Arguments**:

- There are no filter conditions and the argument definition in **Arguments** starts with an ampersand, for example,`&sysparm_fixed_query=assigned_to=javascript:gs.user_id()`.

  This argument breaks the module URI because the equals sign and the colon are not URL-encoded, and the ampersand prevents ServiceNow from URL-encoding the argument. URL-encode the argument: `&sysparm_fixed_query=assigned_to%3Djavascript%3Ags.user_id()`.

- There are filter conditions and the argument definition in **Arguments** does not start with a caret (^), for example, `sysparm_name=Barnes & Noble's`.

  This argument breaks the module URI because the ampersand and spaces are not URL-encoded. URL-encode the argument: `sysparm_name=Barnes%20%26%20Nobel's`

Module Visibility

Verify the visibility, or lack thereof, of selected modules in the application navigator (left navigation bar). For example, create a test that first impersonates a user, then verifies that specified modules (such as Homepage and My Requests) are visible, or are not visible, to that user.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	The application scope in which the system runs this test or test suite.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Navigator	Navigator version to assert against • Core UI (UI16): If you are creating new steps, you will have Core UI by default. • Next Experience: If you have Next Experience enabled, Next Experience is the default navigator. If it is disabled, Core UI is the default navigator.
Visible assert type	Specifies how modules selected in the **Visible modules** field should be tested for visibility in the application navigator. • **At least these modules are visible**: At minimum, the modules selected in the **Visible modules** field are visible in the application navigator. • **Only these modules are visible**: Only the specific modules selected in the **Visible modules** field are visible in the application navigator.
Visible modules	Modules whose visibility in the application navigator is being verified.
Not visible assert type	Specifies how modules selected in the **Not visible modules** field should be tested for lack of visibility in the application navigator. • **At least these modules are not visible**: At minimum, the modules selected in the **Not visible modules** field are not visible in the application navigator. • **Only these modules are not visible**: Only the specific modules selected in the **Not visible modules** field are not visible in the application navigator.
Not visible modules	Modules whose lack of visibility in the application navigator is being verified.

Related topics

Create a module

Navigate to Module

Open a module from the application navigator, as if a user had clicked it. The module must be visible to the currently executing user to navigate to it.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: Not all pages are currently testable. Wherever the module takes you is your responsibility.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Module	Module that should be opened. To navigate to the selected module, the module must be visible to the currently executing user in the application navigator. The following modules are not supported and cannot be tested: • Modules that are separators • Modules that do not link to a specific page, but instead execute client-side JavaScript (such as Studio and the Script Debugger) • Modules that link to external websites, such as the ServiceNow documentation site (servicenow.com/docs) • Modules that reload or redirect the entire page

Related topics

Create a module

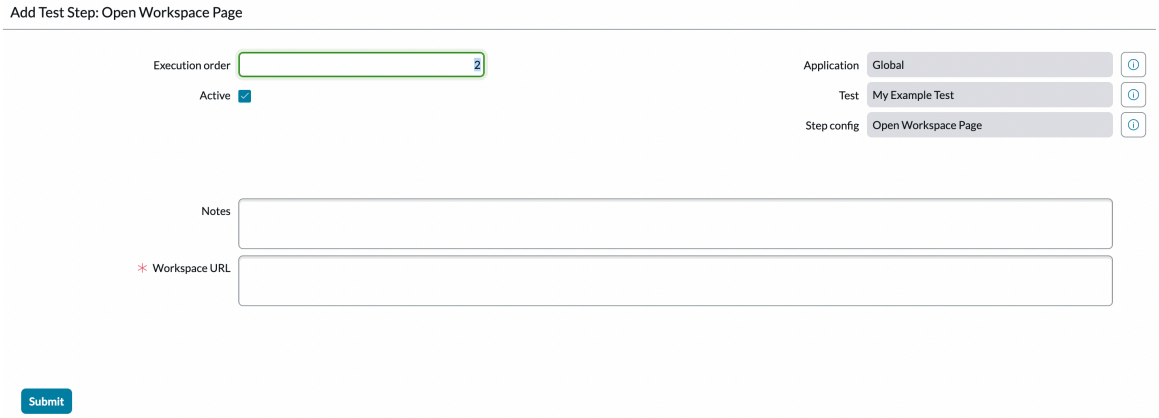
Configurable Workspace category

Interact directly with the configurable workspace components for simpler test creation.

Open Workspace Page

Navigate to a workspace page using a URL. The feature allows for URLs to be entered with or without the preceding domain name.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Inputs

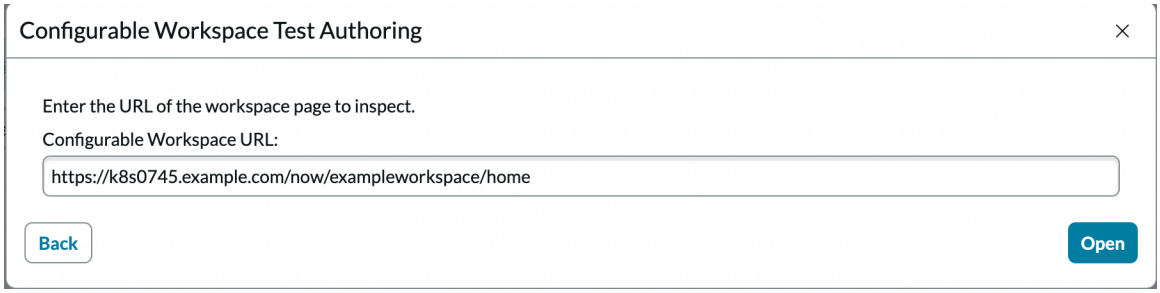
Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Notes	Notes about the test step.
Workspace URL	URL of the workspace page you want to use. **Note:** The URL can be entered with or without the domain, but it will be trimmed upon saving for compatibility between test instances.

Test Page

Interact with a seismic component on a workspace page using the Test Page test step.

Enter the URL of the workspace page you want to inspect for the Configurable Workspace URL field on the Configurable Workspace Test Authoring modal.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Note: By default, the URL from the preceding Open Workspace Page test step is displayed. If this step is not immediately preceded by an Open Workspace Page step, you must enter the target workspace page URL manually for inspection.

Custom UI category

Validate the behavior of page components on custom user interfaces.

Set Component Values (Custom UI)

Set component values on a custom UI page.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Component values	Page components to be tested and the value each page component should have for the step to succeed. **Note:** The **Value** field defaults to the last retrieved value of the selected component.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Assert Text on Page (Custom UI)

Assert that the specified text is or is not on a custom UI page.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Text	Text to be compared with the text on the page.

Note:

- The text to be searched is case-sensitive.
- Starting with the Rome release, this test step also works for Workspace for any new tests.
- The Assert Text on Page test step also accepts text containing white spaces.

Component Value Validation (Custom UI)

Validate a component value on a custom UI page.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
	As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Component values	Page components to be tested and the value each page component should have for the step to succeed. **Note:** The **Value** field defaults to the last retrieved value of the selected component.

Click Component (Custom UI)

Click a component on a custom UI page.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Component	Component to be tested in this step.

Component State Validation (Custom UI)

Validate the state of a specified component on a custom UI page.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Component	Component to be tested in this step.
Assert type	Type of assert: • **Disabled** • **Enabled**

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Open Service Portal Page

Open a portal page. Test designers must first open a page in a portal before testing UI components on the page.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Portal	Select a portal.
Page	Select the page to open.
Query parameters	Add any required query parameters for the page. For example, to open a record in the form page, enter the table and sys_id parameters. For more information about Service Portal query parameters, see Page navigation by URL.

Form category

Validate the functionality of fields and UI actions on a form.

Note: Test steps that include the **Form UI** field give you the option to select an available UI. For any available workspace, navigation between tabs is not supported. Use the **Open a New Form** or **Open an Existing Record** step to reopen a form.

It is to be noted that Agent Workspace has been removed from the Platform. If you have any Agent Workspace tests or test suites, they won't work anymore.

Open a New Form

Open a form to a new record in the specified table and **Form UI**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step
Form UI	Option to select an available UI. Apart from the Standard UI, you have other workspace selections available. Depending on the selected workspace, the UI might vary. **Note:** Workspaces are not supported in Internet Explorer (IE). See KB0683275 for more details.
Table	Name of the table for the new form.
View	Name of the view in which you want this form to open. The testing user must have access to that view. If the name is not a valid form view, the form opens in its default view. **Note:** This field appears when **Standard UI** is chosen from **Form UI**.

Open an Existing Record

Open a form to an existing record in the specified table and **Form UI**.

Note: Using an existing record may cause unexpected behavior for this test. See Automated Test Framework design considerations for more information.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step
Form UI	Option to select an available UI. Apart from the Standard UI, you have other workspace selections available. Depending on the selected workspace, the UI might vary. **Note:** Workspaces are not supported in Internet Explorer (IE). See KB0683275 for more details.
Table	Name of the table for the new form.
Record	Record ID of the record that you want to open and name of the document that you want to open.
View	Name of the view in which you want this form to open. The testing user must have access to that view. If the name is not a valid form view, the form opens in its default view. **Note:** This field appears when **Standard UI** is chosen from **Form UI**.

Set Field Values

Set the fields on the current form to the specified values.

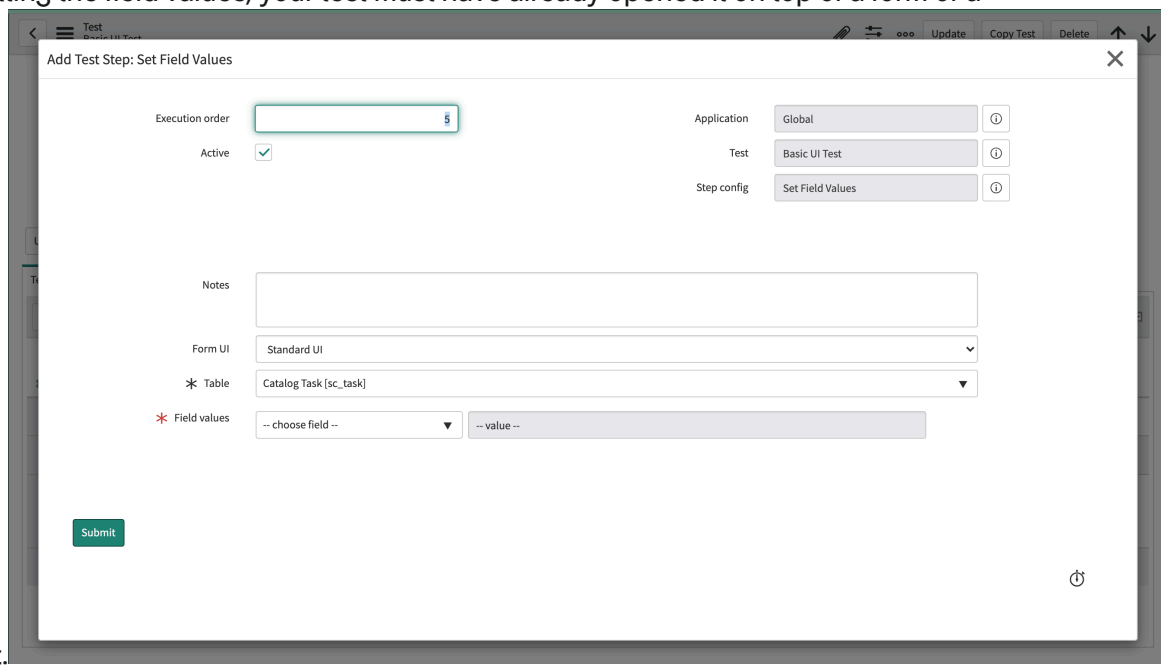
To run this step, your test must have already opened a form using either the **Open a New Form** or **Open an Existing Record** step. It is recommended to not run this step directly after a **Submit**

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

a Form or **Click a UI Action** step. This is because they can redirect your test to a different page based on the navigation stack configuration on your instance or the script defined in the clicked UI action. Unless you are certain that the UI action will take you to a specific page, you should explicitly use an **Open a New Form** step after **Submit a Form** or **Click a UI Action** to ensure that the test is on the form as expected. Ensure that the test keeps passing consistently when added to a suite.

The **Field Values Validation**, **Set Field Values**, **Field State Validation**, and **UI Action Visibility** steps can appear in any order.

Note: This step waits for the form to load before setting field values. This step doesn't support reference qualifiers, neither at test design time nor at test runtime. A modal form appears either on top of another form or a list. To submit a modal form after setting the field values, your test must have already opened it on top of a form or a list.



Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step
Form UI	Option to select an available UI. Apart from the Standard UI, you have other workspace selections available. Depending on the selected workspace, the UI might vary. **Note:** Workspaces are not supported in Internet Explorer (IE). See KB0683275 for more details.
Table	Name of the table for the new form.
Set field value	Assigns a value to a field on an open form.

Field Values Validation

Validate field values on the current form.

To run this step, your test must have already opened a form using either the **Open a New Form** or **Open an Existing Record** step. It is recommended to not run this step directly after a **Submit a Form** or **Click a UI Action** step. This is because they can redirect your test to a different page based on the navigation stack configuration on your instance or the script defined in the clicked UI action. Unless you are certain that the UI action will take you to a specific page, you should explicitly use an **Open a New Form** step after **Submit a Form** or **Click a UI Action** to ensure that the test is on the form as expected. Ensure that the test keeps passing consistently when added to a suite.

The **Field Values Validation**, **Set Field Values**, **Field State Validation**, and **UI Action Visibility** steps can appear in any order.

For the **Field Values Validation** step, specify the values that you want to test using the standard conditions builder. You can test several conditions against the same field. This step passes if the overall condition is satisfied and fails if the condition is not satisfied. The **Conditions** field is case-sensitive and requires to have the exact value as on the form. To test the values of individual fields independently of each other, include a separate **Field Values Validation** step for each value that you test.

Note: The **Field Values Validation** step works only with fields that belong to the record for the open form. For example, with the incident table, this step is not able to validate the **Additional comments**, **Approval history**, **Comments**, or **Work notes** fields because these UI controls are not actual fields on the incident record. These UI controls make it convenient to work with related tables. To validate these cases, use the Server test step, **Record Validation**, instead.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step **Note:** Use the condition builder to set the field value. The condition builder displays an appropriate control for the field data type. For example, a reference field displays a **Lookup record** control.
Form UI	Option to select an available UI. Apart from the Standard UI, you have other workspace selections available. Depending on the selected workspace, the UI might vary. **Note:** Workspaces are not supported in Internet Explorer (IE). See KB0683275 for more details.
Table	Name of the table for the new form.
Conditions	Fields and values to be validated. Includes only fields that are visible on the open form. This field is case-sensitive.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field State Validation

Validate the state of specified fields. States validated can include mandatory, non-mandatory, read-only, non-read-only, visible, and non-visible.

You can specify a maximum time to wait for the states of the fields to match the conditions in this step.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step
Form UI	Option to select an available UI. Apart from the Standard UI, you have other workspace selections available. Depending on the selected workspace, the UI might vary. **Note:** Workspaces are not supported in Internet Explorer (IE). See KB0683275 for more details.
Table	Name of the table for the new form.
Visible	Validates whether the fields on this form are visible. The test fails if the fields are not visible.
Not visible	Validates whether the fields on this form are visible. The test fails if the fields are visible.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
Read only	Validates whether the fields on this form are read only. The test fails if the fields are not read only.
Not read only	Validates whether the fields on this form are read only. The test fails if the fields are read only.
Mandatory	Validates whether the fields on this form are mandatory. The test fails if the fields are not mandatory.
Not mandatory	Validates whether the fields on this form are mandatory. The test fails if the fields are mandatory.

UI Action Visibility

Verify if a UI action is visible on the current form. To run this step, your test must have already opened a form using either the **Open a New Form** or **Open an Existing Record** step.

The default visible UI actions vary depending on the user that you're currently impersonating.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
Notes	Notes about the test step
Form UI	Option to select an available UI. Apart from the Standard UI, you have other workspace selections available. Depending on the selected workspace, the UI might vary. **Note:** Workspaces are not supported in Internet Explorer (IE). See KB0683275 for more details.
Table	Name of the table for the new form.
Visible	Validates whether UI actions on this form are visible. The test fails if the UI actions are not visible.
Not visible	Validates whether UI actions on this form are visible. The test fails if the UI actions are visible.

Declarative Action Visibility

Verify if a declarative action is visible on the current form.

The default visible declarative actions vary depending on the user that you're currently impersonating.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Notes	Notes about the test step
Form UI	Option to select an available UI. Apart from the Standard UI, you have other workspace

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
	selections available. Depending on the selected workspace, the UI might vary. **Note:** Workspaces are not supported in Internet Explorer (IE). See KB0683275 for more details.
Table	Name of the table for the new form.
Visible	Validates whether declarative actions on this form are visible. The test fails if the declarative actions are not visible.
Not visible	Validates whether declarative actions on this form are visible. The test fails if the declarative actions are visible.

Add Attachments to Form

Add one or more mandatory attachments to the current form. Select the attachments that the test step adds to the form from the Upload Attachments list.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Application	Application scope in which the system runs this step.
Active	Option to activate this test step for use.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Notes	Notes about the test step
Upload Attachments	Button to add one or more mandatory attachments to the form.

Note: The Add Attachments to Form test step can't access the newly added UI workspace options.

Click Modal Button

Click a button within a modal in the specified **Form UI**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Specify your testing by selecting either the standard platform UI or workspace UI from the **Form UI** field. If you select the standard platform UI, this test step selects the button by ID on the specified modal and validates the following conditions:

- UI page was opened in a modal.
- Button is visible and enabled.

If you select an available workspace UI, this test step selects either the Confirm or Cancel action and optionally sets the field values within the modal. This step succeeds only if a modal dialog is open on the form, and if the specified button exists on that modal dialog.

Only modals opened with the following g_modal functions are supported:

- alert
- confirm
- confirmDestroy
- showFields

Note: Click Modal Button now supports global and scoped application modals.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Timeout	Number of seconds for the modal and the button to appear.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step
Form UI	Option to select an available UI.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
	Note: Workspaces are not supported in Internet Explorer (IE). See KB0683275 for more details.
UI page	The UI modal dialog to be tested. **Note:** This field appears only if **Standard UI** is selected from **Form UI**.
Button	The ID attribute of the button element. For example, the **OK**button has an ID of `OK_button` in the UI modal dialog. **Note:** This field appears only if **Standard UI** is selected from **Form UI**.
Modal values	The field values to be set in the modal. **Note:** This field appears only if a workspace is chosen as **Form UI**.
Modal action	The action to be selected in the modal. **Note:** This field appears only if a workspace is chosen as **Form UI**.
Assert type	The effect on the tests on clicking the modal button. • **Modal closed and page is reloaded or redirected**: The test passes only if the modal was closed and the page was reloaded or redirected to another page. • **Modal closed and page is not reloaded or redirected**: The test passes only if the modal was closed and the page was not reloaded or redirected to another page. • **Modal not closed**: The test passes only if the modal still exists and was not closed.
Assertion timeout	Number of seconds allowed before the assert fails. If the assert fails, the system repeats the assert until the duration of the Timeout is reached. If the validation fails after the Timeout duration has passed, the step fails.

Note: The Click Modal Button test step can't access the newly added UI workspace options.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Click a UI Action

Click a UI action on the current form.

When this step runs, the system performs the action normally activated by that control. The test step also validates that the current form contains the control and that the control is visible and enabled. To run this step, your test must have already opened a form using either the **Open a New Form** or **Open an Existing Record** step. It is recommended to not run this step directly after a **Submit a Form** or **Click a UI Action** step. This is because they can redirect your test to a different page based on the navigation stack configuration on your instance or the script defined in the clicked UI action. Unless you are certain that the UI action will take you to a specific page, you should explicitly use an **Open a New Form** step after **Submit a Form** or **Click a UI Action** to ensure that the test is on the form as expected. Ensure that the test keeps passing consistently when added to a suite.

In the Brazil release, this step supports UI actions of type Form context menu.

Note: Don't write tests that depend on the system displaying a specific page after executing a **Submit a Form** or **Click a UI Action** step. After these test steps, the system returns to the page that was open before the form was opened. The test cannot determine what that page was, so writing a test that expects a particular page can lead to unpredictable results.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step
Form UI	Option to select an available UI. Apart from the Standard UI, you have other workspace selections available. Depending on the selected workspace, the UI might vary.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
	Note: Workspaces are not supported in Internet Explorer (IE). See KB0683275 for more details.
Table	Name of the table for the new form.
UI action	The UI action to click. **Note:** The declarative action and UI action buttons look exactly the same. Depending on the selected test step (either Click a UI Action or Click a Declarative Action), the type of button can be identified.
Assert type	The effect on the tests with clicking of a UI action. • **Form submission canceled in browser**: The step passes only if form submission is canceled. • **Form submitted to server**: The step passes only if the form is submitted.

Outputs

Field	Description
table	The table for the form that contains this UI action.
record	The sys_id of the record on which the action was clicked.

Click a Declarative Action

Click a declarative action on the current form.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Notes	Notes about the test step
Form UI	Option to select an available UI. Apart from the Standard UI, you have other workspace selections available. Depending on the selected workspace, the UI might vary. **Note:** Workspaces are not supported in Internet Explorer (IE). See KB0683275 for more details.
Table	Name of the table for the new form.
Declarative action	Option to select a declarative action from the list. **Note:** The declarative action and UI action buttons look exactly the same. Depending on the selected test step (either Click a UI Action or Click a Declarative Action), the type of button can be identified.
Assert type	The effect on the tests with clicking of a UI action. • **Form submission canceled in browser**: The step passes only if form submission is canceled. • **Form submitted to server**: The step passes only if the form is submitted.

Submit a Form

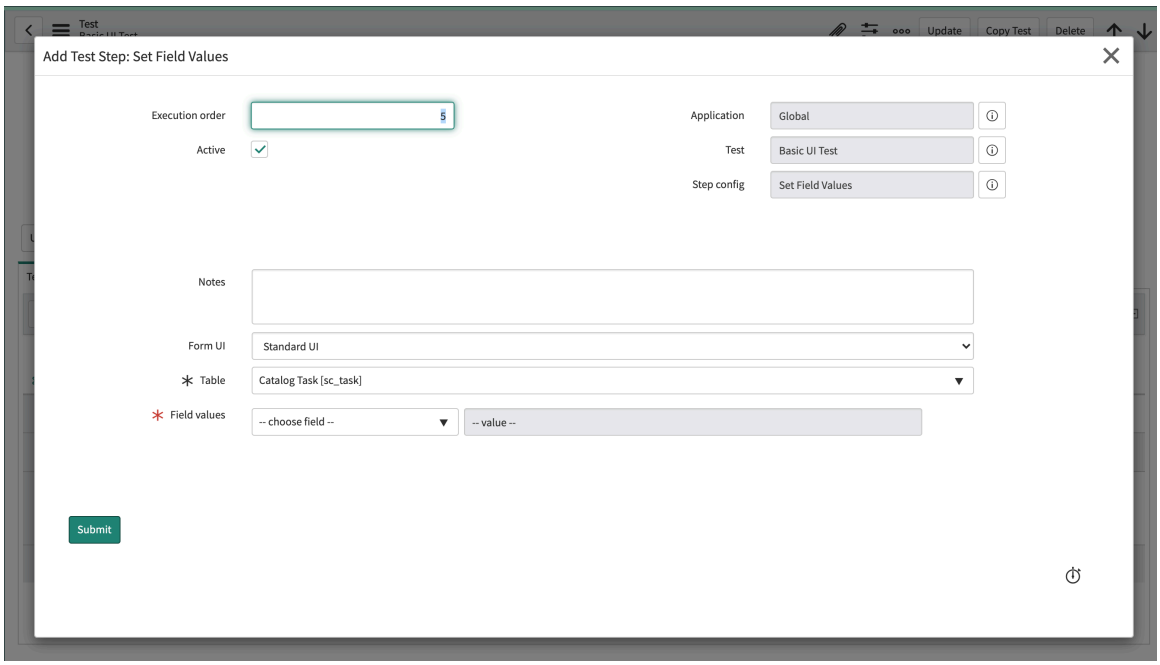
Submit the current form.

To run this step, your test must have already opened a form using either the **Open a New Form** or **Open an Existing Record** step. It is recommended to not run this step directly after a **Submit a Form** or **Click a UI Action** step. This is because they can redirect your test to a different page based on the navigation stack configuration on your instance or the script defined in the clicked UI action. Unless you are certain that the UI action will take you to a specific page, you should explicitly use an **Open a New Form** step after **Submit a Form** or **Click a UI Action** to ensure that the test is on the form as expected. Ensure that the test keeps passing consistently when added to a suite.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: Don't write tests that depend on the system displaying a specific page after executing a **Submit a Form** or **Click a UI Action** step. After these test steps, the system returns to the page that was open before the form was opened. The test cannot determine what that page was, so writing a test that expects a particular page can lead to unpredictable results.

A modal form appears either on top of another form or a list. To submit a modal form, your test must have already opened it on top of a form or a list.



Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
	Note: This field appears when an available workspace is chosen from **Form UI**.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step
Form UI	Option to select an available UI. Apart from the Standard UI, you have other workspace selections available. Depending on the selected workspace, the UI might vary. **Note:** Workspaces are not supported in Internet Explorer (IE). See KB0683275 for more details.
Assert type	The effect on the tests on submitting a form. • **Form submitted to server**: The step passes only if the form is submitted. • **Form submission canceled in browser**: The step passes only if form submission is canceled. • **None**: Make no assertion on submitting the form. The step always passes when using this assert type regardless of the result of submitting the form.

Outputs

Field	Description
table	The table for the submitted record.
record	The sys_id of the submitted record.

Service Catalog category

Validate single catalog item transactions as well as requester and fulfiller flows in Service Catalog.

Activation of the Automated Test Framework for Service Catalog

These test steps require activation of the The Automated Test Framework Service Catalog (com.glide.automated_testing_impl.service_catalog) plugin, which is active by default on new

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

instances. Administrators may need to activate the plugin on instances upgraded from earlier versions.

Support for parametrized tests

Service Catalog step configurations support parametrized tests. For more information on parametrized tests, refer to Parameterized tests.

Variable editor support

After opening a record that supports variable editor (requested item, catalog task, or incident), you can add step configurations to set variable values, or validate variable states or values. Use the step configurations in the following order to support variable editor.

1. Step configurations to order a catalog item or record producer in the **Service Catalog** category.
2. **Open an Existing Record** step configuration in the **Form** category.
3. **Set Variable Values**, **Validate Variable Values**, or **Variable State Validation** step configuration in the **Service Catalog** category.

Note: Custom variables and custom variable with labels are not supported for Set Variable Values, Validate Variable Values, and Variable State Validation step configurations.

Open a Catalog Item

Open a catalog item.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Catalog Item	Catalog item that you want to open. **Note:** You should have access to this catalog item.

Open a Record Producer

Open a record producer.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Record Producer	Record producer that you want to open. **Note:** You should have access to the record producer.

Set Variable Values

Set variable values for the current catalog item or the record producer.

For a catalog item, use this step after opening a catalog item page using the **Open a Catalog Item** step, and before using the **Order Catalog Item** step. For a record producer, use this step after opening a record producer page using the **Open a Record Producer** step, and before using the **Submit Record Producer** step.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
	Note: Use the condition builder to set the field value. The condition builder displays an appropriate control for the field data type. For example, a reference field displays a **Lookup record** control.
Item	Catalog item or record producer for which you want to set variable values.
Variable Values	List of variables and the values that you want to set for them. **Note:** You can set the value for multiple variables.

Set Catalog Item Quantity

Set the quantity for the current catalog item.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Item	Catalog item whose quantity you want to set.
Quantity	Quantity you want to set for the catalog item.

Validate Variable Values

Validate variable values on the current catalog item or record producer.

For a catalog item, use this step after opening a catalog item page using the **Open a Catalog Item** step, and before using the **Order Catalog Item** step. For a record producer, use this step after opening a record producer page using the **Open a Record Producer** step, and before using the **Submit Record Producer** step.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
	As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step. **Note:** Use the condition builder to set the field value. The condition builder displays an appropriate control for the field data type. For example, a reference field displays a **Lookup record** control.
Item	Catalog item or record producer whose variables should be validated.
Catalog Conditions	Conditions for variable validation. If the conditions are met, the test passes. **Note:** The label of a variable associated with a variable set reflects the variable set name. The format is *`variable_set_name » variable_name`*.

Variable State Validation

Validate the state of variables. Possible variable states are mandatory, not mandatory, read only, not read only, visible, and not visible.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Catalog item	Catalog item for which you want to validate the state. **Note:** You should have access to this catalog item.
Visible	List of the catalog item variables that must be visible for the step to pass.
Not visible	List of the catalog item variables that must be hidden for the step to pass.
Read only	List of the catalog item variables that must be read-only for the step to pass.
Not read only	List of the catalog item variables that must not be read-only for the step to pass.
Mandatory	List of the catalog item variables that must be mandatory for the step to pass.
Not mandatory	List of the catalog item variables that must not be mandatory for the step to pass.

Validate Price and Recurring Price

Validate price and recurring price of a catalog item. Use this step after opening a catalog item page using the **Open a Catalog Item** step, and before using the **Order Catalog Item** step.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
	As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Price	Price of the catalog item for the step to pass.
Recurring price	Recurring price of the catalog item for the step to pass.
Recurring price frequency	Recurring price frequency of the catalog item for the step to pass.

Add Item to Shopping Cart

Add a catalog item to the shopping cart. Use this step after opening a catalog item page using the Open a Catalog Item step. After this step, you cannot use any other steps on the catalog item.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
Assert Type	Criteria for the test to pass. **Successfully added item to Shopping Cart** Test passes only if the catalog item is successfully added to the shopping cart. **Cannot add item to Shopping Cart** Test passes only if the catalog item cannot be added to the shopping cart.

Outputs

Field	Description
cart_item_id	The sys_id of the added catalog item.

Order Catalog Item

Clicks **Order Now** for a catalog item. Use this step after opening a catalog item page using the Open a Catalog Item step.

After this step, you cannot use any other steps on the catalog item. If the two-step checkout is false, a request is generated for the catalog item. If the two-step checkout is true, you are redirected to the cart preview page.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Assert type	Criteria for the test to pass. **Successfully ordered Catalog Item** Test passes only if the catalog item is successfully ordered. **Cannot order Catalog Item**

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
	Test passes only if the catalog item cannot be ordered.

Outputs

Field	Description
request_id	The sys_id of the created catalog request.
cart	the sys_id of the cart holding the catalog item.

Submit Record Producer

Submits the current record producer. Use this step after opening the record producer page using the Open a Record producer step. After this step, you cannot use any other steps on the catalog item.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Assert Type	Criteria for the test to pass. **Successfully submitted Record Producer** Test passes only if the record producer is submitted successfully. **Cannot submit Record Producer** Test passes only if the record producer cannot be submitted.

Outputs

Field	Description
record_id	The sys_id of target record of the record producer.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Forms in Service Portal category

Validate the functionality of fields and UI actions in Service Portal form widgets.

Service Portal dependency

Creating automated Service Portal steps requires knowledge of the ServiceNow data model and Service Portal form and data structures.

Open a Form (SP)

Opens a form in a portal.

Use this step for the base system Form page. For custom form pages, use the Open Service Portal Page step from the Custom UI category.

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Portal	Portal in which the defined form opens. Service Portal is the default.
Page	Page to open in the defined portal. The form page is the default.
Table	Table containing the form to open.
sys_id	Sys_id of the record to open. Default is `-1`, which opens a new record.
View	The form view to open. If blank, the system uses the default view.
Query parameters	Additional URL query parameters and values for the form.

Set Field Values (SP)

Sets the values of fields in a form. To use this step, you must have already opened a form using the **Open a Form (SP)** test step.

Field	Description
Execution order	Integer specifying the order in which the test executes this step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
	As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step. **Note:** Use the condition builder to set the field value. The condition builder displays an appropriate control for the field data type. For example, a reference field displays a **Lookup record** control.
Table	The table for the form on which to set field values. The value should be the table in the **Open a Form (SP)** step.
Field values	Fields for which to set values and the values to be set for those fields.

Field Values Validation (SP)

Validates field values on the current form based on defined conditions. To use this step, you must have already opened a form using the **Open a Form (SP)** test step.

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
	Note: Use the condition builder to set the field value. The condition builder displays an appropriate control for the field data type. For example, a reference field displays a **Lookup record** control.
Table	The table that contains the form on which to validate fields. The value should be the table in the **Open a Form (SP)** step.
Conditions	Conditions used to validate one or more fields on the form. If the condition evaluates to true, the step passes.

Field State Validation (SP)

Validates field states on a form in Service Portal.

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Table	The table for the form on which to validate field states. The value should be the table in the **Open a Form (SP)** step.
Visible	Validates whether the fields on this form are visible. The test fails if the fields are not visible.
Not visible	Validates whether the fields on this form are visible. The test fails if the fields are visible.
Read only	Validates whether the fields on this form are read only. The test fails if the fields are not read only.
Not read only	Validates whether the fields on this form are read only. The test fails if the fields are read only.
Mandatory	Validates whether the fields on this form are mandatory. The test fails if the fields are not mandatory.
Not Mandatory	Validates whether the fields on this form are mandatory. The test fails if the fields are mandatory.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Add Attachments to Form (SP)

Test the functionality of attaching a file to a Service Portal form widget.

To use this step, you must have already opened a form using the **Open a Form (SP)** test step or **Open Service Portal Page steps**. This step can't be used after a **Submit Form** step or **Click a UI Action** step has been used.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Notes	Notes about the test step.
Upload Attachments	Button to attach one or more files to the form.

UI Action Visibility Validation (SP)

Determines whether a UI action on the current Service Portal form is visible. To use this step, you must have already opened a form using the **Open a Form (SP)** test step.

Service Portal only supports server UI actions. The setRedirectURL() method and client UI actions are not supported. UI action visibility can vary depending on the currently logged in or impersonated user.

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Table	The table for the form on which to check UI action visibility. The value should be the table in the **Open a Form (SP)** step.
Visible	Fields from the UI Actions table to be checked for visibility. Includes only form-based UI actions. The test fails if the UI action is not visible on the form for the currently logged in user. **Note:** If the list contains UI actions with the same name, check the form to determine the sys_id of the element. Then filter by sys_id in the UI Action table to select the correct element in the step.
Not visible	Fields from the UI Actions table to be checked for visibility. Includes only form-based UI actions. The test fails if the UI action is visible on the form for the currently logged in user. **Note:** If the list contains UI actions with the same name, check the form to determine the sys_id of the element. Then filter by sys_id in the UI Action table to select the correct element in the step.

Click UI Action (SP)

Selects a UI action on the current Service Portal form and outputs the table and sys_id of the record on which the action was selected.

To use this step, you must have already opened a form using the **Open a Form (SP)** test step. After using this step, you cannot use any other form steps.

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Table	The table for the form on which to click a UI action. The value should be the table in the **Open a Form (SP)** step.
UI action	The UI action to click, selected from the UI Actions table. Includes only form-based UI actions.
Assert type	Specifies where to check for form submission after clicking the UI action: • **--None--**: Selects the UI action without validating mandatory or other fields. • **Form submission canceled in browser**: Checks whether the form was canceled in the browser and did not reach the server due to validation or other issues. • **Form submitted to server**: Checks whether the form was submitted to the server.

Outputs

Field	Description
record_id	Sys_id of the record on which the action was clicked.
table	Table with the clicked UI action.

Submit a Form (SP)

Submits the current form in a Service Portal page and outputs the table and sys_id of the submitted record.

To use this step, you must have already opened a form using the **Open a Form (SP)** test step. After using this step, the page closes. You can't use any other steps on the current page.

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Notes	Notes about the test step.
Assert type	Specifies where to check for form submission: • **--None--**: Submits the form without validating mandatory or other fields. • **Form submitted to server**: Checks if the form was submitted to the server. • **Form submission canceled in browser**: Checks whether the form was canceled in the browser and did not reach the server due to validation or other issues.

Outputs

Field	Description
record_id	Sys_id of the submitted record.
table	Table for the submitted record.

List and Related List

Validate the functionality and visibility of records and UI actions in lists and related lists.

Test Agent can now generate ATF tests that use List and Related List test steps, extending test coverage beyond form interactions to include list view operations such as validating visibility and applying filters. See Test Agent for more information.

Consider the following while creating the tests:

- Related lists has three sources: default platform relationships (Attachments, Approvals, Journal Entries), reference fields on the table, and custom relationships defined in `sys_relationship`. Only test related lists that have custom logic (client scripts, UI policies, or custom relationships) affecting their behavior. Default platform related lists without custom logic don't require testing.
- Before deciding a related list has no custom behavior, search the target table for matching client scripts (`sys_script_client`), UI policies (`sys_ui_policy`), and custom relationships (`sys_relationship`) that reference related list operations.
- Related list parameters use the format `<child_table>.<field>` (for example, `task.parent`) or `REL:<sys_id>` for custom relationships. These values must be looked up for each table.
- When a related list is confirmed to have custom logic, test all applicable methods: `relatedListVisibility` for visibility, `applyFilterToList` for filtering, `recordPresentInList` for membership, `openRecordInList` for navigation, and `listUIActionVisibility` for UI Actions. `clickListUIAction` should be used only when the specific action behavior needs verification, not as a default check.
- ATF list steps are part of a 13-category ATF namespace. Select `atf.list` for list/related list interactions; use `atf.form` for form field interactions, and `atf.uiTestScript` for custom UI not reachable through standard form or list steps.

Validate Related List Visibility

Validate the visibility of the selected related lists on a form.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them. You can change this default order by editing the **Execution order** values.
Application	Application scope in which the system runs this step.
Active	Option to activate this test step for use.
Test	Name of the test that you're adding the step to.
Step config	Name of the step.
Notes	Notes about the test step.
Table	Name of the table of the parent form where related list visibility is validated.
Visible	List of related lists to assert as visible.
Not visible	List of related lists to assert as not visible.

Apply Filter to List

Apply a filter to a list to find the required record.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Name of the test that you're adding the step to.
Step config	Name of the step.
Notes	Notes about the test step.
List type	Option to select the type of list. • **List** • **Related list**
Assert Type	Specifies the assertion on executing the test step:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
	• **At least one matching record found**: The step succeeds if there is at least one matching record. • **No matching record found**: The step fails if there is any matching record. • **One matching record found**: The step succeeds if there is exactly one matching record.
Table	If the **List type** is **List**, this field is the table of the opened list. If the **List type** is **Related list**, this field states the name of the table of the parent form where the list belongs.
Related List	The related list to apply a filter to. **Note:** This field is available when **Related list** is selected from **List type**.
List filter	Filter conditions to apply to the list.

Outputs

Field	Description
first_record	The first record found in the list after applying the indicated filter.

Validate Record Present in List

Validate the presence of a record in a list. A valid form must be open and the list containing the record must be visible to proceed.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them. You can change this default order by editing the **Execution order** values.
Application	Application scope in which the system runs this step.
Active	Option to activate this test step for use.
Test	Name of the test that you're adding the step to.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Step config	Name of the step.
Notes	Notes about the test step.
List type	Option to select the type of list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
	• **List** • **Related list**
Assert type	Specifies the assertion on executing the test step: • **Record is present in the list** • **Record is not present in the list**
Table	Name of the parent table where the list belongs.
Related list	The related list which has the record to validate. **Note:** This field is available when **Related list** is selected from **List type**.
Record	The record whose presence is validated.

Open a Record in List

Open a specific record in a list.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them. You can change this default order by editing the **Execution order** values.
Application	Application scope in which the system runs this step.
Active	Option to activate this test step for use.
Test	Name of the test that you're adding the step to.
Step config	Name of the step.
Notes	Notes about the test step.
List type	Option to select the type of list. • **List** • **Related list**
Table	Name of the parent table where the list belongs.
Related List	The related list which has the record to open. **Note:** This field is available when **Related list** is selected from **List type**.
Record	The record to be opened.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Validate List UI Action Visibility

Validate that a UI action is visible in a list. If you're impersonating a user, the visibility of a UI action can change depending on the user being impersonated.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them. You can change this default order by editing the **Execution order** values.
Application	Application scope in which the system runs this step.
Active	Option to activate this test step for use.
Test	Name of the test that you're adding the step to.
Step config	Name of the step.
Notes	Notes about the test step.
List type	Option to select the type of list. • **List** • **Related list**
Table	Name of the table of the parent form.
Related List	The related list which has the UI actions. **Note:** This field is available when **Related list** is selected from **List type**.
Visible	List of UI actions to assert as visible.
Not visible	List of UI actions to assert as not visible.

Click a List UI Action

Select a list UI action in a list on a form.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them. You can change this default order by editing the **Execution order** values.
Application	Application scope in which the system runs this step.
Active	Option to activate this test step for use.
Test	Name of the test that you're adding the step to.
Step config	Name of the step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
Notes	Notes about the test step.
List type	Option to select the type of list. • **List** • **Related list**
Table	Name of the table where the UI action belongs.
Related list	The related list which has the UI actions. **Note:** This field is available when **Related list** is selected from **List type**.
List action	The list UI action to be clicked.
Action type	The type of UI action to be clicked.
Apply to	The record on which the UI action is applied.
Assert type	The assertion made by the step when executed. • **None**: Click a UI action and continue without making an assertion. • **Page reloaded or redirected**: Click a UI action, and assert that the page was reloaded or redirected. This assert type doesn't assert form submission.
Record	The record to which the UI action is applied. **Note:** This field appears when you choose Single record in the **Apply to** field.

REST category

Verify the functionality of REST calls.

Send REST Request - Inbound - REST API Explorer

This test step begins with the REST API Explorer. Use the REST API Explorer to create and specify the HTTP method, path, query parameters, request headers, and body of a REST request, and then send the REST request to the current instance.

When you have tested the request, the **Create Automated Test Step** button appears. Click **Create Automated Test Step** to create the test step. This button does not appear until after the request has been sent. You cannot create a test step when the request payload is larger than the maximum request payload size property.

This test step creates the same test record as the **Send REST Request - Inbound** test step. After the test step is created, you cannot go back and use the REST API Explorer to update the test. All changes must be made on the **Send REST Request - Inbound** test step form.

No HTTP response validation is performed as part of this step. The step fails if the response payload size is too big, the request parameters are invalid, or the request could not be sent. Use the assert steps to validate the response.

You cannot use this step to send a request to another instance or third party/remote address.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

These inputs are for the REST API Explorer. The fields you see depend upon the API selected. For more information on using REST APIs on your instance, see REST APIs.

Inputs

Field	Description
Namespace	Namespace for the request. Select from the list.
API Name	API to be used. Select from the list.
API Version	API version to be used. Select the version available on your instance from the list.
Path parameters	The part of the path after the API name. Path parameters are generally name-value pairs where the allowable values are in a list.
Query parameters	Name-value pairs of query parameters added to the URI after the path. The REST API Explorer encodes the URI, so it is not necessary to encode query parameters and values.
Request headers	Name-value pairs contained in the request header. The authentication header is set to **Send as me** to use the current user's credentials. To use the test step in production, you may need to change the **Authentication Type** field in the **Send REST - Inbound** test step. Do not encode the name or value.
Request Body	The request content. Some requests do not have a body.

Send REST Request - Inbound

Create a test step to send a REST request to the current instance. Specify the HTTP method, path, query parameters, request headers, and body of a REST request.

No HTTP response validation is performed as part of this step. The step fails if the response payload size is too big, the request parameters are invalid, or the request could not be sent. Use the assert steps to validate the response.

You cannot use this step to send a request to another instance or third party/remote address.

For more information on using REST APIs, see the REST APIs.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
	As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Authentication Type	The type of authentication to use. For public APIs, select **None**. To configure basic or mutual authentication, users need the atf_ws_designer role. For more information, see Automated Test Framework roles.
Basic authentication	The basic authentication profile to use when doing the test. You must create or select a basic authentication profile to assign to test steps to avoid authentication issues when running the test. For more information, see Create a basic auth profile using the Automated Test Framework.
Mutual authentication	An X.509 certificate for mutual authentication. You must create or select a client certificate to assign to test steps to avoid authentication issues when running the test. For more information, see Set up Certificate-based authentication.
Method	The HTTP method to be used: • **GET** • **POST** • **PUT**

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
	• **DELETE** • **PATCH**
Path	The path to be used. This field accepts only the portion of the URI after the instance name. If you use `https://<instance name>`, you get an error.
Query Parameters	Query parameter names and values. Do not encode the parameter names or values.
Headers	Header names and values. Do not encode the header names or values.
Body	The body of the request.

Assert Status Code

Assert that the HTTP response status code has the specified relationship to the specified value. You specify a numeric value of the status code and the relationship.

Assert steps must immediately follow a **Send REST Request - Inbound** step. You can have multiple REST assert steps following a **Send REST Request - Inbound** step, but the assert steps cannot be separated from the **Send REST Request - Inbound** step by steps from other test categories.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
Notes	Notes about the test step.
Operation	Comparison to be performed between values: • **is** • **is not** • **less than** • **greater than** • **less than or is** • **greater than or is**
Status code	Status code to be tested against the response code.

Assert Status Code Name

Assert that the HTTP response status code name has the specified relationship to the specified value. You specify a value of the status code name, and the relationship.

Assert steps must immediately follow a **Send REST Request - Inbound** step. You can have multiple REST assert steps following a **Send REST Request - Inbound** step, but the assert steps cannot be separated from the **Send REST Request - Inbound** step by steps from other test categories.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
Notes	Notes about the test step.
Operation	Comparison to be performed between values: • **contains** • **does not contain** • **is** • **is not**
Status code name	Status code name to be tested.

Assert Response Time

Assert that the HTTP response time has the specified relationship to the specified value. You specify a value of the response time and the relationship.

Assert steps must immediately follow a **Send REST Request - Inbound** step. You can have multiple REST assert steps following a **Send REST Request - Inbound** step, but the assert steps cannot be separated from the **Send REST Request - Inbound** step by steps from other test categories.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Operation	Comparison to be performed between **Response time** entered and actual response time:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
	• **less than** • **greater than**
Response time (ms)	Time in milliseconds to be compared to the actual response time.

Assert Response Header

Assert the HTTP response header exists, or the header has the specified relationship to the specified value.

Assert steps must immediately follow a **Send REST Request - Inbound** step. You can have multiple REST assert steps following a **Send REST Request - Inbound** step, but the assert steps cannot be separated from the **Send REST Request - Inbound** step by steps from other test categories.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Header	Header name.
Operation	Comparison to be performed between values: • **contains** • **does not contain** • **is**

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
	• **is not** • **is not empty**
Value	Element value to be used in the test. Not shown if the **Operation** is **is not empty**.

Assert Response JSON Payload is Valid

Assert that the response payload is in valid JSON format.

Assert steps must immediately follow a **Send REST Request - Inbound** step. You can have multiple REST assert steps following a **Send REST Request - Inbound** step, but the assert steps cannot be separated from the **Send REST Request - Inbound** step by steps from other test categories.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.

Assert Response XML Payload is Well-Formed

Assert that the response payload is well-formed XML.

Assert steps must immediately follow a **Send REST Request - Inbound** step. You can have multiple REST assert steps following a **Send REST Request - Inbound** step, but the assert steps cannot be separated from the **Send REST Request - Inbound** step by steps from other test categories.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.

Assert XML Payload Element

Assert the XML response payload element exists, or has the specified relationship to the specified value.

Assert steps must immediately follow a **Send REST Request - Inbound** step. You can have multiple REST assert steps following a **Send REST Request - Inbound** step, but the assert steps cannot be separated from the **Send REST Request - Inbound** step by steps from other test categories.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Element path	XML path to the element to be evaluated. For example, `/result/short_description` for `{ "result": { "number": "INC0020001", "short_description": "test" } }`
Operation	Comparison to be performed between values: • **contains** • **does not contain** • **is** • **is not** • **is not empty**
Value	Element value to be used in the test. Not shown if the **Operation** is **is not empty**.

Assert JSON Response Payload Element

Assert the JSON response payload element exists, or has the specified relationship to the specified value.

Assert steps must immediately follow a **Send REST Request - Inbound** step. You can have multiple REST assert steps following a **Send REST Request - Inbound** step, but the assert steps cannot be separated from the **Send REST Request - Inbound** step by steps from other test categories.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
	This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Element path	SNC path to the element to be evaluated. For example, `/result/short_description` for `{ "result": { "number": "INC0020001", "short_description": "test" } }` See Importing JSON files for more information.
Operation	Comparison to be performed between values: • **contains** • **does not contain** • **is** • **is not** • **is not empty**
Value	Element value to be used in the test. Not shown if the **Operation** is **is not empty**.

Assert Response Payload

Assert the HTTP response payload has the specified relationship to the specified value. You specify the value and the relationship.

Assert steps must immediately follow a **Send REST Request - Inbound** step. You can have multiple REST assert steps following a **Send REST Request - Inbound** step, but the assert steps

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

cannot be separated from the **Send REST Request - Inbound** step by steps from other test categories.

Note: The entire payload is used to look for a match. A large payload can affect performance.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Operation	Comparison to be performed between values: • **contains** • **does not contain** • **is** • **is not** • **is not empty**
Response body	The value of the response body to be used in the test. Must contain the name and value to be compared as it appears in the response payload. Must not contain any curly braces. Not shown if the **Operation** is **is not empty**.

To check the short description in the response payload

```
{"result":{"number":"INC0010040","short_description":"Test ATF
 Incident"}}
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

the **Response body** should contain

```
"short_description":"Test ATF Incident"
```

These formats are incorrect and the step fails.

- `{"short_description":"Test ATF Incident"}`
- `"{"short_description":"Test ATF Incident"}"`
- `short_description: Test ATF Incident`
- `short_description:"Test ATF Incident"`

Email category

Use Automated Test Framework (ATF) to test email notifications, outbound email flows, and inbound email responses.

Validate Outbound Email

Verify that a certain outbound email exists by searching for it in the Email [sys_email] table.

Use this step to test any email-generating script that isn't captured in the Validate Outbound Email Generated by Flow or Validate Outbound Email Generated by Notification test steps.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them. You can change this default order by editing the **Execution order** values.
Application	Application scope in which the system runs this step.
Active	Option to activate this test step for use.
Test	Name of the test that you're adding the step to.
Step config	Name of the step.
Notes	Notes about the test step.
Conditions	Conditions used to filter outbound emails. For example, to find an email that was sent to user@example.com, set the condition to **[Recipients] [is] [user@example.com]**.
Table	Table to be queried. By default, this step queries only the Email [sys_email] table.

Note: During testing, this step may take longer than expected to execute. The step times out after two minutes by default.

Validate Outbound Email Generated by Flow

Verify that a certain outbound email exists by searching for it in the Email [sys_email] table. Use this step to test that a flow is triggered as expected.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: This step searches only for emails that are created by a flow Send Email action. To find an email that was sent after a notification was triggered in a flow, use the Validate Outbound Email Generated by Notification step instead. To find an email that was created in any other flow action, such as a custom scripted action, use the Validate Outbound Email step instead.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them. You can change this default order by editing the **Execution order** values.
Application	Application scope in which the system runs this step.
Active	Option to activate this test step for use.
Test	Name of the test that you're adding the step to.
Step config	Name of the step.
Notes	Notes about the test step.
Source Flow	Flow that the outbound email was created from.
Conditions	Conditions used to filter outbound emails. For example, to find an email that was sent to user@example.com, set the condition to **[Recipients] [is] [user@example.com]**.
Table	Table to be queried. By default, this step queries only the Email [sys_email] table.

Note: During testing, this step may take longer than expected to execute. The step times out after two minutes by default.

Validate Outbound Email Generated by Notification

Verify that a certain outbound email exists by searching for it in the Email [sys_email] table. Use this step to test that an email notification is triggered as expected.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them. You can change this default order by editing the **Execution order** values.
Application	Application scope in which the system runs this step.
Active	Option to activate this test step for use.
Test	Name of the test that you're adding the step to.
Step config	Name of the step.
Notes	Notes about the test step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
Source Notification	Notification that the outbound email was created from. This field references the Notification [sysevent_email_action] table.
Conditions	Conditions used to filter outbound emails. For example, to find an email that was sent to user@example.com, set the condition to **[Recipients] [is] [user@example.com]**.
Table	Table to be queried. By default, this step queries only the Email [sys_email] table.

Note: During testing, this step may take longer than expected to execute. The step times out after two minutes by default.

Generate Inbound Email

Create an incoming email record to test an inbound email flow or an inbound email action.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them. You can change this default order by editing the **Execution order** values.
Application	Application scope in which the system runs this step.
Active	Option to activate this test step for use.
Test	Name of the test that you're adding the step to.
Step config	Name of the step.
Notes	Notes about the test step.
From	Address of the email sender. You can manually enter an email address or reference an email address from a user record. If you manually enter an email address, the system matches the address to an existing user record during testing.
To	Address of the email recipient. You can manually enter an email address or reference an email address from a user record. If you manually enter an email address, the system matches the address to an existing user record during testing.
Subject	Subject of the email. You can enter text or reference a string output from a previous test step.
Body	Content of the message body. You can enter text or reference a string output from a previous test step.

Output

Field	Description
output_email_record	Record in the Email [sys_email] table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Output (continued)

Field	Description
	The **Receive type** is **New** and the **Type** is **send - ready**. For more information on what these fields mean, see System email log and mailboxes.

Generate Inbound Reply email

Create a reply email record to test how the system handles a user response to an email notification.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them. You can change this default order by editing the **Execution order** values.
Application	Application scope in which the system runs this step.
Active	Option to activate this test step for use.
Test	Name of the test that you're adding the step to.
Step config	Name of the step.
Notes	Notes about the test step.
Target Table	Table of the target record.
Target Record	Record that the reply email updates. Selecting a target record also applies a watermark to the reply email.
From	Address of the email sender. You can manually enter an email address or reference an email address from a user record. If you manually enter an email address, the system matches the address to an existing user record during testing.
To	Address of the email recipient. You can manually enter an email address or reference an email address from a user record. If you manually enter an email address, the system matches the address to an existing user record during testing.
Subject	Subject of the email. You can enter text or reference a string output from a previous test step.
Body	Content of the message body. You can enter text or reference a string output from a previous test step.

Output

Field	Description
output_reply_email_record	Record in the Email [sys_email] table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Output (continued)

Field	Description
	The **Receive type** is **Reply** and the **Type** is **send - ready**. For more information on what these fields mean, see System email log and mailboxes.

Generate Random String

Generate a string of random alphanumeric characters that you can use as test data for another step.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them. You can change this default order by editing the **Execution order** values.
Application	Application scope in which the system runs this step.
Active	Option to activate this test step for use.
Test	Name of the test that you're adding the step to.
Step config	Name of the step.
Notes	Notes about the test step.
Length	Number of characters in the generated string. The maximum length is 10,000 characters. If you leave the field blank, the string length is 10 characters by default.

Output

Field	Description
random_string	String of random alphanumeric characters.

Server category

Perform server-side operations. For example, query and update a record, impersonate a user, or run a server-side script.

Create a User

Create a user with specified roles and groups for the test. The user record gets rolled back after the test completes.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
	As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them. You can change this default order by editing the **Execution order** values.
Application	Application scope in which the system runs this step.
Active	Option to activate this test step for use.
Test	Name of the test that you're adding the step to.
Step config	Name of the step.
Notes	Notes about the test step.
First name	First name of the user.
Last name	Last name of the user.
Roles	Assigned roles of the user.
Groups	Assigned groups of the user.
Impersonate this user	Option to impersonate the new user.

Outputs

Field	Description
user	The user ID of the user impersonated.

Impersonate

Impersonate the specified user for the test.

Impersonate specifies a user for executing subsequent steps in this test. It works for both server-side and browser-side steps and stays in effect until changed with another **Impersonate** step or until the test ends. The impersonation automatically ends when the test is over.

Note:

- Do not impersonate a user with the test author role. Doing so can lead to conflicts that interfere with executing the test.
- Tests which involve impersonated users which no longer exist fail.

  Tip: It is recommended to create a new user to avoid data dependencies. See Create a User, for more information.

- Do not rely on user IDs being consistent across different instances. The system dynamically assigns users IDs, so the ID for a particular user often differs from one instance to the next.
- When exporting and importing automated tests, keep in mind that update sets do not update the user field.
- Tests can impersonate users with the snc_external role, which allows testing users who do not have login access. See Explicit Roles for requirements of the snc_external role.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
User	The user ID for the test to impersonate.

Outputs

Field	Description
user	The user ID of the user impersonated.

Search for a Catalog Item

Searches for a catalog item or record producer in the specified catalog and category. You can perform this step both in Platform and Service Portal.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Search in Portal only	Selected if the search is restricted to the Service Portal only. Otherwise, not selected.
Search term	Term used to search for a catalog item.
Catalog	Catalog in which to search for the catalog item.
Category	Category in which to search for the catalog item.
Assert item	Catalog items that should be available in the search results.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
Assert Type	Specifies how searching the catalog item affects the test: • **Assert Item present in search result**: Test passes only if the assert item is present in the search result. • **Assert Item not present in search result**: Test passes only if the assert item is not present in the search result.

Outputs

Field	Description
catalog_item_id	The sys_id of the first catalog item found that the user can view.

Record Query

Query the database to verify that a record exists matching the conditions set in this step.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Enforce security	Selected to enforce ACLs. Otherwise, not selected.
Notes	Notes about the test step. **Note:** Use the condition builder to set the field value. The condition builder displays an appropriate control for the field data type. For example, a reference field displays a **Lookup record** control.
Assert type	Specifies how querying the record affects the test:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
	• **There is at least one record matching the query**: The test fails if there are no records matching the query. • **No records match the query**: The test fails if any records match the query.
Table	The table to be queried.
Conditions	Conditions used to run the query.

Outputs

Field	Description
table	The table queried.
first_record	The first record resulting from the query.

Note: If you don't update your record query test step, the original record query test step still functions the same way as before, irrespective of it being termed as **(Deprecated)**.

Record Insert

Inserts a record into a table with the field values you specify.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step. **Note:** Use the condition builder to set the field value. The condition builder displays an appropriate control for the field data type. For example, a reference field displays a **Lookup record** control.
Assert type	Specifies how inserting the record affects the test:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
	• **Record successfully inserted**: Test fails if the record was not successfully inserted. • **Record was not inserted**: Test fails if the record was successfully inserted.
Enforce security	Selected to enforce ACLs and the read-only role. Otherwise, not selected.
Table	The table into which the record should be inserted.
Conditions	Specific field values to be set when the test runs this step.

Outputs

Field	Description
table	The table to which the new record belongs.
record_id	The sys_id of the new record.

Record Update

Changes field values on a record on the server.

Note: To ensure that the changes were applied, follow this step with a **Record Validation** step.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step. **Note:** Use the condition builder to set the field value. The condition builder displays an appropriate control for the field data type. For example, a reference field displays a **Lookup record** control.
Assert type	Specifies how updating the record affects the test:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
	• **Record successfully updated**: Test fails if the record was not successfully updated. • **Record was not updated**: Test fails if the record was successfully updated.
Enforce security	Selected to enforce ACLs and the read-only role. Otherwise, not selected.
Table	The table containing the record to be updated.
Record	ID of the record to be updated.
Field values	Fields for which you want to set values and the values you want to set for those fields. **Note:** Use the condition builder to set the field value. The condition builder displays an appropriate control for the field data type. For example, a reference field displays a **Lookup record** control.

Note: Record Update step succeeds even if a field on the record is blocked by ACL. Use the Record Validation step after Record Update to check whether a particular field was modified, or use the Form steps to evaluate ACL conditions for individual fields on a record.

Record Delete

Deletes a specified record in a table.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Assert type	Specifies how updating the record affects the test: • **Record successfully deleted**: Test fails if the record was not successfully deleted. • **Record was not deleted**: Test fails if the record was successfully deleted.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
Enforce security	Selected to enforce ACLs and the read-only role. Otherwise, not selected.
Table	The table containing the record to be deleted.
Record	ID of the record to be deleted.

Record Validation

Validates that a record meets the specified conditions on the server side.

For the **Record Validation** step, specify the values you want to test using the standard conditions builder. You can apply several conditions to the same field.

This step passes if the overall condition is satisfied and fails if it is not. If you need to test the values of individual fields independently of each other, include a separate **Record Validation** step for each value to be tested.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step. **Note:** Use the condition builder to set the field value. The condition builder displays an appropriate control for the field data type. For example, a reference field displays a **Lookup record** control.
Assert type	Specifies how validating the record affects the test: • **Record successfully validated**: The test fails if the record does not match the conditions. • **Record not found**: Test fails if the record is found.
Table	The table that contains the field to be validated.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
Record	The record that contains the field to be validated.
Field values	Specific fields to be validated when the test runs this step.

Run Server Side Script

Executes a script on the server.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Jasmine version	The version of the Jasmine testing framework to use for running the server-side script. Any new scripts you create use Jasmine version 3.1. Your existing scripts can continue to use Jasmine version 1.3, or you can upgrade them to Jasmine version 3.1.
Test script	The javascript for the server to execute. Supports the Jasmine testing framework. **Note:** steps(SYS_ID) can be defined as a function to retrieve Output variable data from a step that executed earlier in the test. The Run Server Side Script test step now supports parameters as step inputs.

Outputs

Field	Description
record_id	The sys_id output by the server-side script.
table	The table output by the server-side script.

Note: If the script creates data, the system rolls back that data after all steps in the test finish. The output for the Run Server Side Script test step now indicates the line and the line number for the cause of failure.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Test script

```
//  Test step 1 - add data
var now_GR = new GlideRecord('sc_task');
//  this sample step's Step config has Output variables named
 table and record_id
outputs.table = 'sc_task';
outputs.record_id = gr.insert();
//  Test step 2 - access added data and validate
//  check that the record exists (or that business logic
 changed it)
var now_GR = new GlideRecord("sc_task");
gr.get(steps(PREVIOUS_STEP_SYS_ID).record_id);
assertEqual({name: "task gr exists", shouldbe: true, value:
 gr.isValidRecord()});
stepResult.setOutputMessage: Log a message to step results
 after step executes.
                              Can only be called once or will
 overwrite previous message
```

- Create a suite of test scripts

```
var now_GR = new GlideRecord('sc_task');
gr.setValue('short_description', 'verify task can be
 inserted');
var grSysId = gr.insert();
var justCreatedGR = new GlideRecord('sc_task');
if (justCreatedGR.get(grSysId)) {
stepResult.setOutputMessage("Successfully inserted task
 record");
return true; // pass the step
} else {
stepResult.setOutputMessage("Failed to insert task record");
return false; // fail the step
}
```

- Jasmine test

```
describe('my suite of script tests', function() {
it('should meet expectations', function() {
expect(true).not.toBe(false);
});
});
//  make sure to uncomment jasmine.getEnv().execute(); outside
 the function body
assertEqual: A function used to compare that assertion.shouldbe
 == assertion.value;
in case of failure it throws an Error and logs that the
 assertion by name has failed
```

Note: `describe` is only supported in Global scope. Use `describe` to create a suite of test scripts and `it` to define test expectations.

- Logs message to test step output

```
var testAssertion = {
name: "my test assertion",
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

```
shouldbe: "expected value"
value: "actual value",
};
assertEqual(testAssertion); // throws Error, logs message to
 test step output
```

- See Step Execution Scripts for Run Server Side script example.

Replay Request Item

Get the item and requester from an existing request item, add the item to a new cart for that user, and place an order.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Original Request Item	The request item to be replayed.

Outputs

Field	Description
table	The table to which the replayed request item belongs.
request	The replayed request item.

Related topics

Automated Test Framework use case: test a Service Catalog request

Log

Logs a message and stores it as a step result.

The log message can contain variables and other information pertaining to the test. The message is stored as a step result.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Log	The message to be logged. To include the value of an output variable from a previous step, click the input value icon ( ) and follow the procedure to Pass values from one automated test step to another.

Add Attachments to Existing Record

Add one or more mandatory attachments to the specified record. Use **Upload Attachments** to select from the attachments the test step adds to the record.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them. You can change this default order by editing the **Execution order** values.
Application	Application scope in which the system runs this step.
Active	Option to activate this test step for use.
Test	Name of the test that you're adding the step to.
Step config	Name of the step.
Notes	Notes about the test step.
Table	Valid table selection from the list.
Record	Pre-existing record either from before the test or inserted as a part of the test.
Upload Attachments	Button to add one or more mandatory attachments to the record.

Checkout Shopping Cart

Submits the cart and generates a request. You can perform this step both in the ServiceNow AI Platform and Service Portal.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Timeout	Number of seconds allowed before the step fails. If the validation fails, the system repeats the step until it reaches the duration of the timeout. If the validation fails after the timeout duration has passed, the step fails.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
Assert Type	Criteria for the test to pass. **Empty cart** Test fails if the cart is not empty. **Successfully Checkout cart** Test fails if the cart is not successfully checked out.
Requested For	User for whom the request is generated.
Delivery Address	Delivery address for the request.
Special Instructions	Special instructions for the request.

Outputs

Field	Description
request_id	The sys_id of the submitted request.

Custom Scripted StepConfig

Provides an example of scripts for a custom step configuration.

This example checks if the user name provided starts with the letter A. This step is useful primarily to users with the [atf_test_admin] role. Users with the [atf_test_admin] role can view the example scripts by opening the record for this step in Step configurations.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You can change this default order by editing the **Execution order** values.
Active	Option to activate this test step for use.
Application	Application scope in which the system runs this step.
Test	Read-only name of the test that you're adding the step to.
Step config	Read-only name of the step.
Description	Description of the test step. This field value is automatically set based on the field values of the test step. This field appears after the test step is submitted.
Notes	Notes about the test step.
User	The user whose name the system checks to see if it starts with the letter **A**. To include the value of an output variable from a previous step, click the input value icon ( ) and follow the procedure to Pass values from one automated test step to another.

Outputs

Field	Description
value	The name of the user.

Related topics

- Create custom step configuration
- Step execution scripts
- Step description generation script

UI category

Validate the functionality of the UI actions.

Run UI Test Script

Executes a client-side test script entirely in the browser without requiring server-side processing.

Inputs

Field	Description
Execution order	Integer specifying the order in which the test executes this step. As you create steps, the system automatically assigns each step an incremental value. This value causes the test to execute steps in the order that you created them in. You

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Inputs (continued)

Field	Description
	can change this default order by editing the **Execution order** values.
Application	Application scope in which the system runs this step.
Active	Option to activate this test step for use.
Test	Name of the test that you're adding the step to.
Step config	Name of the step.
Notes	Notes about the test step.
Script	Script to be executed by the client test runner. The Run UI Test Script API is called in this test step.

Automated Test Framework (ATF) reference

Reference information for the Automated Test Framework.

Watch https://players.brightcove.net/6274575390001/nUx4EKfUz_default/index.html?videoId=6286204365001 for some helpful tips to use ATF features.

Tables excluded from rollback after running an automated test

The Automated Test Framework tracks data created by running tests and rolls back changes after testing. The system excludes certain tables from being tracked during testing.

The system excludes certain tables from being tracked or rolled back:

- The History [sys_history_line] table
- The ECC Queue table [ecc_queue].
- The Email [sys_email] Email Log [sys_email_log] tables
- The Report Executions [report_executions] and ReportStats [report_stats] tables.
- The Execution Tracker [sys_execution_tracker] tables
- The Progress Worker [sys_progress_worker] table
- The Schema Change [sys_schema_change]
- The Upgrade History [sys_upgrade_history] and Upgrade History Log [sys_upgrade_history_log] tables
- The Mutex [sys_mutex] table
- The Plugin Log [sys_plugin_log] table
- The Status [sys_status] table
- The Number Counter [sys_number_counter] table
- The AMB Channel Presence [sys_amb_channel_presence] table
- Any table that extends an excluded table.
- Any table starting with the following prefixes are also excluded:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- syslog
- sys_amb_message
- sys_cluster
- cmdb_metric
- ts_
- v_
- sys_delete_recovery

If your test run changes (inserts/updates/deletes) any record on these excluded tables, the system does not roll back the change after testing.

Tests

The Tests module opens the Test table. From here, you can add, edit, and run tests. By opening an individual test record, you can view and edit the steps comprising that test.

Related topics

Building and running automated test suites

Test record form

In the Test record form, you view and edit values of fields for the test record.

Fields

Field / UI Element	Description
Name	Test name.
Description	Test description.
Active	If this test is active, **true**. Otherwise, **false**.
Application	Application scope in which the system runs this test or test suite.
Test steps related list	The steps that this test executes.
Test results related list	The results from individual executions of this test.
Fail on server error	The errors detected while performing UI steps on the server that occur and fail the test.

UI Actions

Name	Description
Update	Click to update the test record.
Run test	Click to run test.
Copy test	Click to copy test.
Delete	Click to delete this test.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Suites

The Suites module opens the Test Suites table. You can create, edit, and run test suites from this table.

Related topics

Building and running automated test suites

Test suite form

The Test Suite form contains information about one test suite.

Fields

Field / Element	Description
Name	The name of the test suite.
Active	To make this test suite active, check this field.
Description	(Optional) Enter a description to identify the purpose of this test suite.
Application	Application scope in which the system runs this test or test suite.
Parent Suite	To make this test suite a child of another test suite, enter the name of the parent test suite here.
Test Suite Tests (related list)	The tests included in this test suite.
Child Test Suites (related list)	Any test suites that are children of this test suite.
Test Suite Results (related list)	Results from executing this test suite.
Test Suite Schedules (related list)	Any test suite schedules that include this test suite.

Automated Test Framework design considerations

Create reliable, scalable, and efficient tests by following these design considerations.

General testing

Avoid modifying ServiceNow system tables or tables extending the Application File [sys_metadata] that can potentially change the behavior of the system. Avoid using or modifying any existing records to prevent unexpected results between tests. The following are some of the common examples of system data changes that can cause unexpected results.

- Impersonate an existing account
- Delete an existing record.
- Run a test that disables a business rule or system property
- Validate with an existing record

Parallel testing

Reduce test design time by running multiple tests and test suites in parallel. Design tests to run in parallel by avoiding resource conflicts and data dependencies.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Prevent resource conflicts between parallel tests

Prevent resource conflicts by running tests that create their own data. Tests that run with existing data prevent other tests that need the same data from running in parallel.

Note: If you have two or more resource conflicting tests, see Mark tests as mutually exclusive to create a mutual exclusion rule that prevents them from running in parallel.

Parameterized testing

Run a test multiple times with different test data for each run. Create parameters to store test data for each test run. See Parameterized test components for more information.

- Create parameters to store test data for each test run.
- Ensure that the parameterized tests support standard Automated Test Framework (ATF) features, such as reports, test suites, and data rollback. Copying a parameterized test copies all parameters, test run data sets, and test steps.

Note: If a parameterized test including Custom UI test steps is created, the system only uses the first data set to retrieve components.

Custom UI testing

Test customized user interfaces such as UI pages and UI macros by retrieving their HTML and JavaScript page components and identifying the test actions they support.

Use the page inspector to identify testable page components

The page inspector determines which page components are available for custom UI testing. Page components that are unavailable to the page inspector are unavailable to custom UI testing.

Navigate to the custom UI you want to test

Use existing test steps to navigate to the target custom UI. For example, to test a Knowledge Base article, use the existing test steps to navigate to a module or to open an existing record. Most custom UI testing requires using existing test step categories as part of the test.

Use the component area to identify page components

The component area describes the HTML layout element containing the component such as a `<div>` or `<section>` element. The area helps test designers distinguish between components by providing the location in the page layout.

Test your custom UI rather than ServiceNow AI Platform UI

The Automated Test Framework prevents custom UI testing of ServiceNow AI Platform features. For example, you cannot test dashboards or graphical designers. Instead, build tests to validate your custom UI pages and elements because you have direct control over these user interfaces.

Use HTML attributes to override page component testing properties

Change the testing properties of a particular page component using HTML attributes that are specific to Automated Test Framework. See Override component test actions.

Retrieve page components again when you move tests to another instance

Custom UI test steps don't store UI components as metadata. Testers must manually retrieve page components again when moving tests between instances.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Clone tests from production system

Move your tests to the production system to clone the most updated instances for testing. Speed up the testing time by directly copying or cloning a test from the production system to a subproduction instance.

Note: By default, the system property that is used to run automated tests is disabled to prevent you from accidentally running these tests on a production system. To avoid data corruption or an outage, run tests only on development, test, and other non-production instances.

Warning messages for all testing

Warning messages	Design considerations
`Impersonating an existing user may cause unexpected behavior for this test. Avoid potential issues by adding a 'Create a User' step instead. See the documentation for Test Design Considerations.`	Create a new user to ensure proper roles and groups and avoid using existing records. See General testing for more information.
`Using a table that extends Application File [sys_metadata] may cause unexpected behavior for other tests running in parallel. See the documentation for Test Design Considerations.`	Avoid running a test with a table that extends the Application File because it might affect other tests. See Parallel testing for more information.
`Using a system table may cause unexpected behavior for other tests running in parallel. See the documentation for Test Design Considerations.`	Avoid using a system table because it might affect other tests running in parallel. See Parallel testing for more information.
`Using an existing record may cause unexpected behavior for this test. See the documentation for Test Design Considerations.`	Avoid using existing records because these records might not have the state and values as expected by the test. Use records created during the test to ensure proper state and values. See General testing for more information.
`Modifying an existing record may cause unexpected behavior for other tests running in parallel. See the documentation for Test Design Considerations.`	Avoid using existing records because it might affect other tests. Use records created during the test. See General testing for more information.
`Using assert type '--None--' may cause unexpected behavior for server UI actions. Avoid potential issues by setting the assert type and using a timeout. See the documentation for Test Design Considerations.`	Server UI actions cause the current form to submit and the page to reload. Select an assert type other than **None** to avoid any unexpected behavior for server UI actions. Set a timeout to ensure that your test waits for the form to be submitted or not submitted before moving on to the next step. When testing server UI actions, the **None** assert type

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Warning messages	Design considerations
	configures automatically to **Form submitted to server**.

Domain separation testing

When testing domain separation, you must set the domain first. This should be part of the first impersonation step of each of the ATF test steps when they are dependent on a domain being set. To learn more about domain separation recommended practices, see Domain separation recommended practices for service providers.

Test results

Each time you run a test, the automated test framework creates a record of the test results. Use the Test Results module to view details about the results of individual tests and individual steps within a test. If the test generated screenshots, they appear as attachments on the test results record.

Each Test Results record displays detailed results information about one test execution, and contains links to detailed step results and test logs. Step results records display information about one step in a test result, while test log (test results item) records contain detailed console logging and test execution information.

By default, the system deletes test and test suite results data 30 days after creation. You can modify this default retention policy as needed in the Table Cleanup module.

Related topics

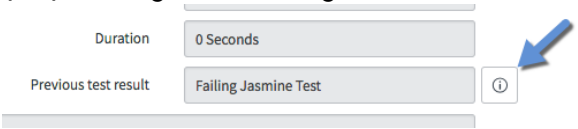
- Modify data retention policy for ATF test results
- Table Cleanup module

Test results record

A Test Results record contains detailed results information about one test execution. Client Error Details and Failure Details sections appear when a test fails due to error conditions, and include detailed error information you can use for troubleshooting purposes.

Field	Description
Test	Name of the test.
Status	Result of the Test execution: **Success** Test passed. **Failure** Test failed. **Success with warning(s)** Test run passed but encountered client error(s) that were allowed as warnings in the Allowed Client Error table. **Waiting** Test waiting to start. **Running** Test is running.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
	Skipped Occurs if an earlier test in the suite failed and **Abort on failure** is set to **true**. **Error** An error occurred in the test framework. For example, the test runner halted, or the server encountered an unintended scenario. The error message appears in the Summary field of the steps results record for the step that threw the error. **Cancelled** User canceled the Test.
Retain indefinitely	Check box to keep this record even after the specified data retention period has passed. For more information, see Table Cleanup.
Start time	Date and time of day at which this test started executing.
End time	Date and time of day at which this test finished executing.
Duration	Elapsed time it took to execute this test.
Previous test result	Results of the previous test execution, indicating if the test passed or failed. Only appears if this test result is for a re-run of a failed test and you are logged on with the admin, atf_test_admin, or atf_test_designer role. Click the information icon (ⓘ) to navigate to the original test result. 

Note: If this test created screenshots, they appear as attachments in **Manage attachments**.

Client Error Details

This section only displays when a test fails due to a client error.

Field	Description
Failing step	Reference to the first test step result that failed during this test. Click the information icon (ⓘ) to view detailed step results and error information in the Step Results form.
Summary	The following text indicates that a client error occurred at this step, and includes the detailed error message: `This step failed because the client error 'DETAILED ERROR MESSAGE' was detected on the page being tested. See failing Test Logs. To ignore these errors in the next test run, use 'Add all client errors to warning/ ignored list' links.`

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
	Test designers and developers should always investigate client errors to determine if there are issues with your business process. For more details, see Identify and resolve client errors.
First failing client error	Reference to the first client error that failed during the test. Click the information icon ( ) to view detailed test results and error information in the Test Logs form. **Note:** For details on how to allow client errors as ignored or warning entries, refer to Allowed client errors
Failing step screenshot	Screenshot of the step that failed. Click to download the screenshot.

Failure Details

This section only displays when a test fails due to a cause other than a client error.

Field	Description
Failing step	Reference to the first step result that failed during this test. Click to download the screenshot.
Summary	Output of the step that failed.
Failing screenshot	Screenshot of the step that failed.

Additional information

This section is visible for all test results.

Field	Description
Output	Output generated for the test step. • Test result (often `Passed` or `Failed`). • For failed tests, includes the same error message detail displayed in the Summary field for the first test step that failed. See Client Error Details above.
Browsers involved	User agent strings returned by browsers that ran the test.

Related lists

Related list	Description
Step Results	Record for each step result and log entry for this test.
Test Log	Record for each test log related to this test results record.
Test Transactions	Record for each transaction (from the system transaction log [syslog_transaction] related to this record).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Related lists (continued)

Related list	Description
	To view the step results associated with a transaction, click the appropriate link in the **Step Results** column. To view the transaction logs associated with a transaction, click the appropriate link in the **Transaction** column. **Note:** The system may not be able to log some transactions with short durations.
Warnings	List of test logs containing client errors with a warning status. The Warnings related list only appears on test results with warnings.

Related topics

- View test results and automated test results
- Test logs record
- Step results record
- Allowed client errors
- Allow client errors from step results
- Allow client errors from the test logs

Step results record

The Step Results record contains information about one step in a test result. You access specific step results from the Step Results related list in the Test Results record.

Field	Description
Status	Result of the step: **Success** Test step passed. **Failure** Test step failed. **Success with warning(s)** Test step passed but encountered client error(s) that were allowed as warnings in the Allowed Client Error table. **Skipped** Occurs if an earlier test in the suite failed and **Abort on failure** is set to **true**. **Error** An error occurred in the test framework. For example, the test runner halted or the server encountered an unintended scenario. **Cancelled** User canceled the test or suite.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Type	Type of test result item: Step Result
Summary	Summarized version of the Output field. If an error occurred in the test step, the following text indicates that a client error occurred at this step, and includes a detailed exception/error message. `This step failed because the client error 'DETAILED ERROR MESSAGE' was detected on the page being tested. See failing Test Logs. To ignore these errors in the next test run, use 'Add all client errors to warning/ignored list' links.` Test designers and developers should always investigate client errors to determine if there are issues with your business process. For more details, see Identify and resolve client errors. For a Send REST Request - Inbound step, the URL for the request endpoint, and a response code. For example: `The HTTP request has been sent to the endpoint 'https://demonightlyus.service-now.com/api/now/table/kb_knowledge'`
Output	Generated output for the test step. • For a step result, the outputs from the step, including any explanation why a step failed or was skipped. • For a Send REST Request - Inbound step, the REST request and response including the response body. The output field is truncated at 4096 characters. ○ The REST request and response headers are filtered to prevent sensitive information from being added to the log. A filtered header text is replaced with the text "Header redacted for security." ○ See Filter REST request and response headers for information on how to add a list of REST request and response headers to be filtered. For additional console logging and test execution information for a test step, see the Test Logs record.
Step	Name of the step executed.
Test Result	Reference to the test result associated with this step result. Click the information icon ( ) to view detailed test results information.
Description	Description of the specific actions performed (for example, Open Create Incident Report Producer, Validate that the form matches the following condition) for this test step. For a client log, blank.
Start Time	Date and time of day at which this step or log entry started.
End Time	Date and time of day at which this step or log entry ended.
Duration	Elapsed time it took to execute this test.
Step Transactions (related list)	Record for each transaction (from the system transaction log [syslog_transaction] related to this step result).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Related topics

- Test results record
- Test logs record
- Allowed client errors
- Allow client errors from step results

Test logs record

The Test Results Item (test log) record contains console logging and test execution information.

During test execution, any information reported to the environment is recorded in the Test Log (sys_atf_test_result_item) table. This information can include browser console logging, results, and error messages recorded by step environments. You access specific test logs from the Test Log related list in the Test Results form.

Field	Description
Status	The result of the test log: **Success** Test step run passed. **Failure** Test step run failed. **Success with warning(s)** Test step passed but encountered client error(s) that were allowed as warnings in the Allowed Client Error table. **Ignored** Client error that has been allowed with a report level of Ignored. **Warning** Client error that has been allowed with a report level of Warning. **Waiting** Test or suite waiting to start. **Running** Test or suite is running. **Skipped** Occurs if an earlier test in the suite failed and **Abort on failure** is set to **true**. **Error** An error occurred in the test framework. For example, the test runner halted or the server encountered an unintended scenario. The error message appears in the Summary field of the Steps Results record for the step that threw the error. **Cancelled** User canceled the test or suite.
Type	Type of test log:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
	• Step Result • Client Log • Client Error **Note:** If a Client Error, you can optionally add it as an ignored or warning entry in the Allowed Client Errors. Doing so prevents the allowed client errors from affecting ATF test executions when they recur in future test runs. For more details, refer to Allowed client errors.
Output	Output generated for the test log. • For a step result, the console logging and test execution outputs from the step, including any explanation why a step failed or was skipped. • For a client log, the log entry text. • For a client error, the actual client error is displayed.
Step	Name of the step executed. This field may be blank for a client log.
Test Result	Reference to the test result associated with this step result. Click to view detailed test result information.
Description	For a step result, the actions performed in this test step. For a client log, blank. For a client, this message displays: `This client error occurred on the page in Browser Type Browser Version` (for example, Chrome 62.0.3202.62).
Start time	Time at which this test step started executing.
End time	Time at which this test step finished executing.
Duration	Time duration it took to execute this test step.
Recorded at	Time at which this step or log entry was recorded.
Allow listed client error	Reference to the Allowed Client Error record (if any). Click to view the Allowed Client Error record for this client error.

Related topics

Test results record

Step results record

Allow client errors from the test logs

Allowed client errors

Suite results

The Suites Results module opens the Suites Results table.

You can drill down to the Test results record to view details about the results of individual test suites, the individual tests within those test suites, any child test suites, and so on.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Related topics

Building and running automated test suites

Test suite results record

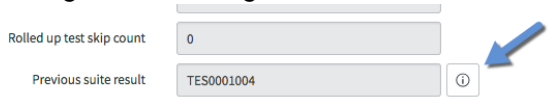
The Test Suite Results record displays information about the results of one execution of one test suite.

Fields

Field / Element	Description
Start Time	(Appears in list of records, but not on default record form). The time this test suite started.
Test Suite	The test suite that was run.
Number	Unique ID for this test suite results record.
Base Test Suite Result	If this test suite is a child in a hierarchy of test suites, the base test suite result is the unique ID of the result record for the suite at the top-most level of the hierarchy. For more information, see Example: Base test suite result.
Parent	If this test suite has a parent, this is the test result record for the parent suite. For more information, see Example: Parent test suite results.
Status	Result of the test or test suite execution: • Canceled – user canceled the test or test suite run. • Error – an error occurred in the test framework. For example, the test runner halted or the server encountered an unintended scenario. The error message appears in the summary field of the Step Results form for the test step that contains the error. • Failure – test, test suite run, or test step failed due to an error. • Ignored – test step passed but contains client-side JavaScript errors entered into the Allowed Browser Error table with report level of Ignored. • Running – test or test suite is running. • Skipped – occurs if an earlier test in the suite failed and **Abort on failure** is set to **true**. • Success – test, test suite, or test step run passed. • Success with warning(s) – test or test suite run passed but contains a test step with client-side JavaScript errors entered into the Allowed Browser Error table with report level of Warning. • Waiting – test or suite run is waited to start. • Warning – test or suite contains client-side JavaScript errors entered into the Allowed Browser Error table with report level of Warning.
Run time	The duration it took to execute this test suite.
Retain indefinitely	Check box to keep this record even after the specified data retention period has passed. For more information, see Table Cleanup.
Rolled up test	How many tests were successful. The tests counted as part of the roll up are all tests included in this suite, plus all others included in suites that are descendents of this one. For more information, see Rolled up counts for test suites results.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Fields (continued)

Field / Element	Description
success count	
Rolled up test failure count	How many tests failed. The tests counted as part of the roll up are all tests included in this suite, plus all others included in suites that are descendents of this one. For more information, see Rolled up counts for test suites results.
Rolled up test error count	How many tests resulted in an error. The tests counted as part of the roll up are all tests included in this suite, plus all others included in suites that are descendents of this one. For more information, see Rolled up counts for test suites results.
Rolled up test skip account	How many tests were skipped. The tests counted as part of the roll up are all tests included in this suite, plus all others included in suites that are descendents of this one. For more information, see Rolled up counts for test suites results.
Test Results (related list)	Results of the individual tests included in this test suite.
Child Test Suites Results (related list)	The results of any test suites that are children of this test suite.
All Test Suite Results (related list)	Results from this test suite and all descendent test suites.
Failed Tests in Suite (related list)	Results from any failed tests included in this test suite.
Previous suite result	Only appears if this suite result is for a re-run of failed tests and you are logged on with the atf_test_admin, atf_test_designer, or admin role. Click the information icon to navigate to the "original" suite result record. 

Related topics

Test results

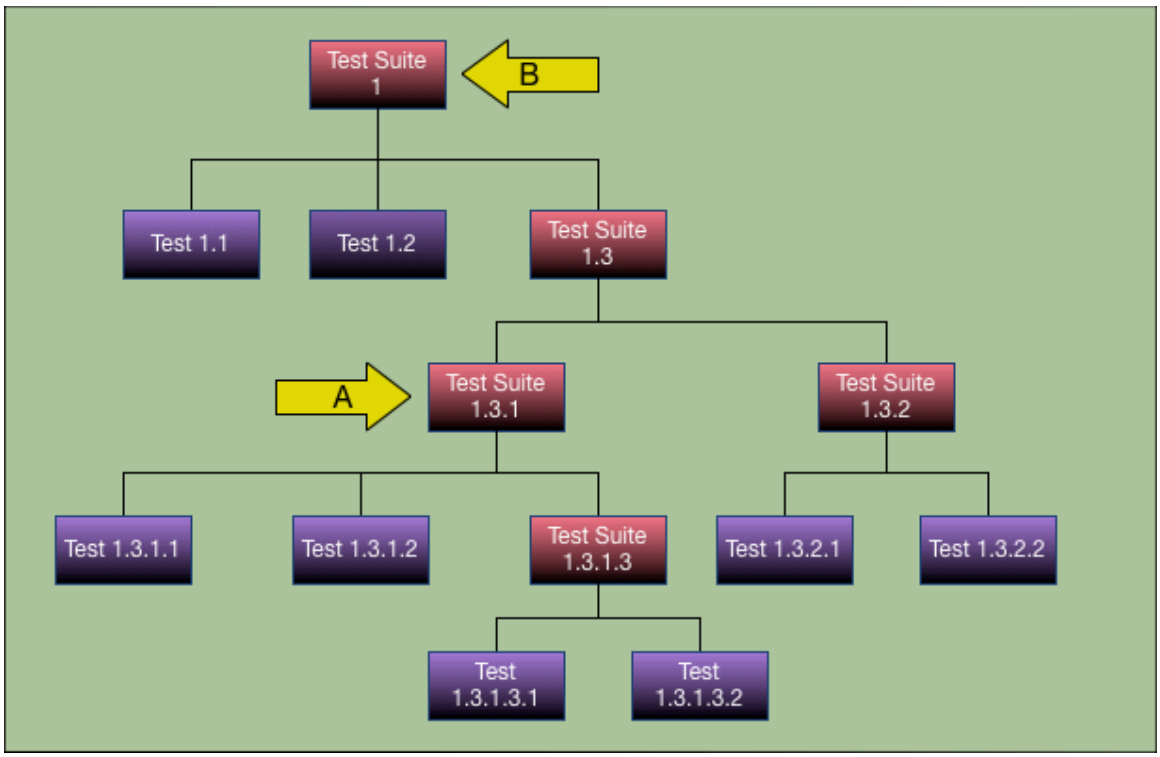
Test suite results examples

Examples of relationship terms and how aggregated results roll up for test suites.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example: Base test suite result

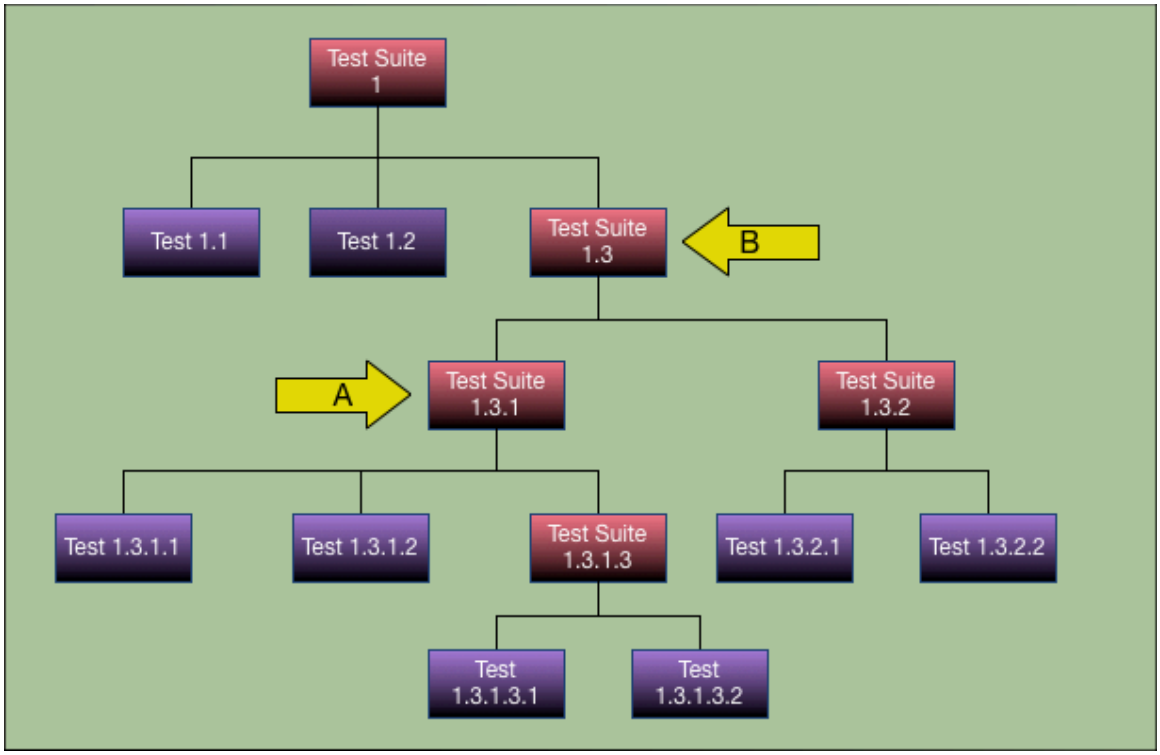
The base test suite is the top-level test suite in the hierarchy. For example, if you are viewing the test suite results record for (A) Test Suite 1.3.1, the **base test suite result** field links to the test suite results record for (B) Test Suite 1.



Example: Parent test suite results

The parent test suite is the test suite immediately above the one you are currently viewing. For example, ff you are viewing the test suite results record for (A) Test Suite 1.3.1, the **parent** field links to the test suite results record for (B) Test Suite 1.3.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



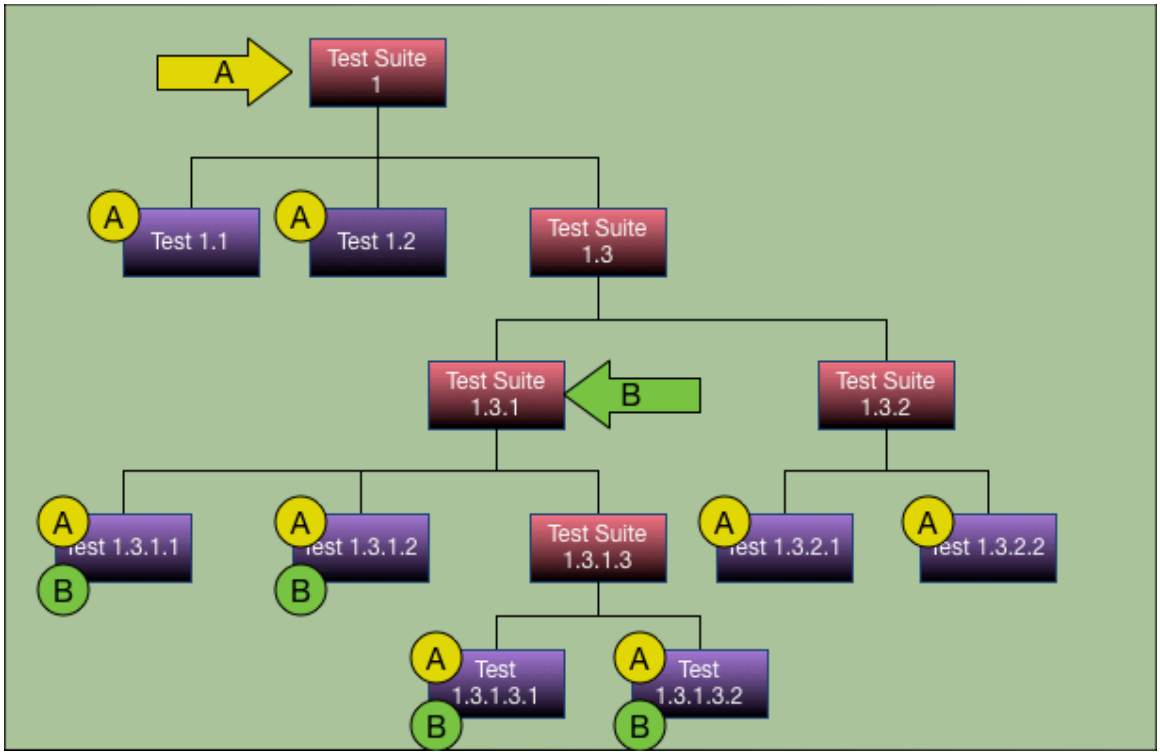
Rolled up counts for test suites results

You can view rolled up counts for a test suite for four metrics: test successes, failures, errors, and skips. Each of these sum data from all tests in the test suite plus all tests in the test suite's descendents.

For example, if you are viewing the test suite results record for (A) Test Suite 1.3.1, the **Rolled up test success count** field shows the total number of successes counting results from all the tests represented by boxes labeled A.

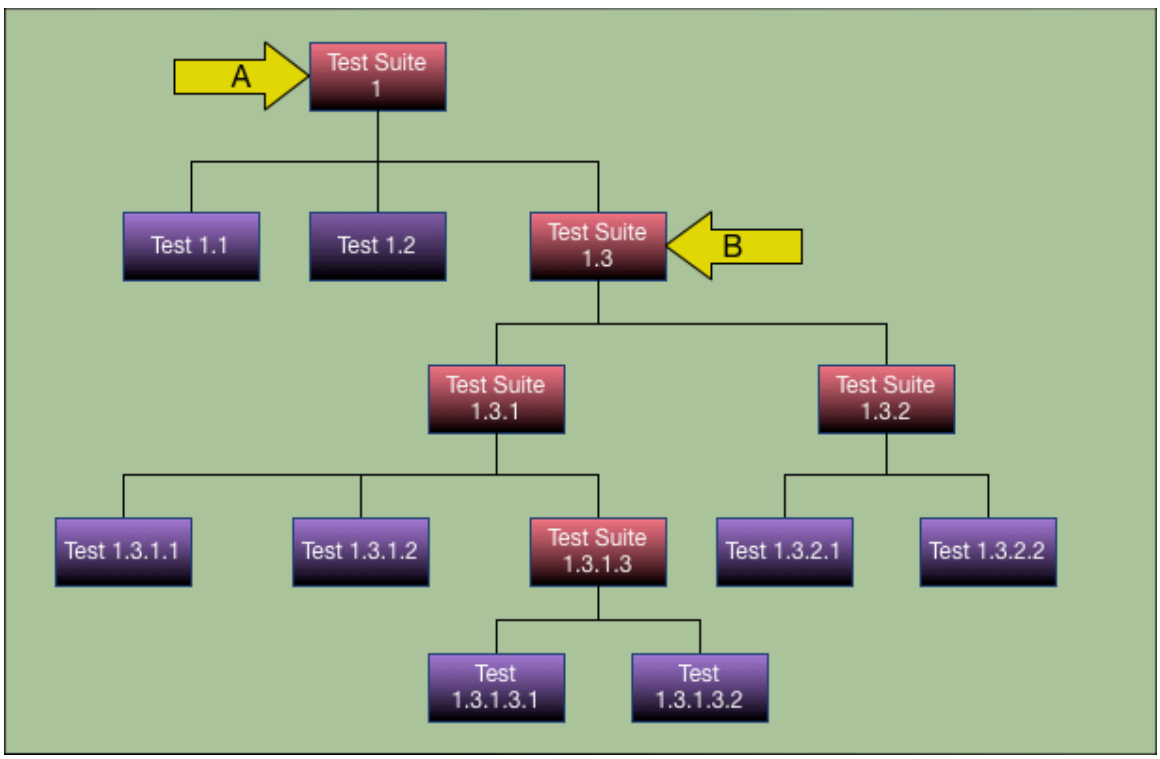
If you are viewing the test suite results record for (B) Test Suite 1.3.1, the **Rolled up test success count** field shows the total number of successes counting results from all the tests represented by boxes labeled B.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Child Test Suites Results

If you are viewing the test suite results record for (A) Test Suite 1, the **Child Test Suites Results** related list shows result records for Test Suite 1.3 (B).

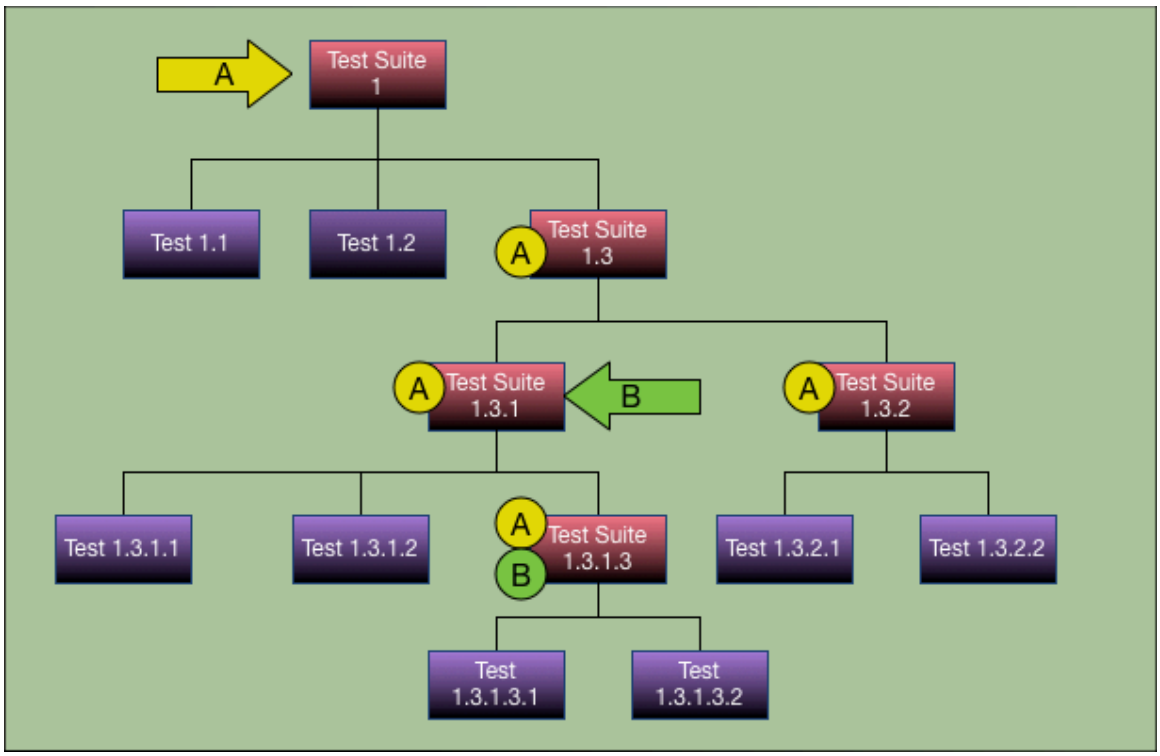


All Test Suites Results

If you are viewing the test suite results record for (A) Test Suite 1, the **All Suites Results** related list shows result records for all Test Suites represented by boxes labeled A

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

If you are viewing the test suite results record for (B) Test Suite 1.3, the **All Suites Results** related list shows result records for all Test Suite 1.3.1 only (represented by the box labeled B)



Suite schedules

Open the Suites Schedules table. You can drill down to see details about the results of individual schedules or create a new schedule.

Related topics

Building and running automated test suites

Suite schedule record

The Suite Results record displays information about one test suite schedule.

Fields

Field / Element	Description
Name	A name for this test suite schedule.
Description	A description for this test suite schedule.
Active	If this schedule is active **true**. Otherwise, **false**.
Application	The application scope for this test suite schedule.
Run	The frequency with which the system runs test suites belonging to this schedule.
Run as tz	The timezone used for the **Time** field.
Time	The time of day at which the system runs test suites belonging to this schedule.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Fields (continued)

Field / Element	Description
Day	If **Run** is set to Weekly, the day of the week on which the system runs test suites belonging to this schedule. If **Run** is set to monthly, the day of the month.
Repeat interval	If Run is set to **Periodically,**the number of days and hours that constitute the repeat interval for running suites in this schedule.
Conditional	Check to enable a script to define conditions under which the system should run the test suites in this schedule.
Condition	If **Conditional** is checked, the script to execute for determining the conditions under which the system should run the test suites in this schedule.
Scheduled Suite Runs (related list)	The test suites the system should run on this schedule.

Related topics

Schedule an automated test suite

Scheduled suite run record

A Scheduled Suite Run record associates a Suite Schedule record with a Test Suite.

Fields

Field / Element	Description
Schedule	The schedule to use.
Test Suite	The test suite to run.
Application	The application scope for this scheduled suite run.
Watch list	Users the system notifies when this scheduled suite run completes.
Browser name	If the test suite has UI components, the browser that must be used. If no client test runner is available with this browser, the system does not run the suite.
Browser version starts with	If the test suite has UI components, the browser version for the client test runner must start with this string. If no client test runner is available with this browser version, the system does not run the suite.
OS name	If the test suite has UI components, the client test runner must run under this OS. If no client test runner is available with this OS, the system does not run the suite.
OS version starts with	If the test suite has UI components, the OS version for the client test runner must start with this string. If no client test runner is available with this OS version, the system does not run the suite.
Test Suite Results (related list)	All test suite results from this scheduled run.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

To determine the browser name and version of a browser you want to use, start a scheduled test runner with that browser, then inspect that runner's record in the Active Scheduled Test Runners Module.

Related topics

Schedule an automated test suite

Run

Start a client test runner and view information about test runners and test runs.

For information on how to work with test runners, see Working with client test runners.

Client test runner

The Client Test Runner opens a browser window or tab for running manually-started client automated tests.

You can toggle a client test runner to act as either a manual or scheduled client test runner. For more information about scheduling test suites, see Schedule an automated test suite.

Fields on the Client Runner Test window

Field / UI Element	Description
Form preferences icon ()	Click to display the form preferences panel.
Form preferences panel: Screenshots mode	Choose among: • Enable for all steps • Enable for failed steps • Disable for all steps For additional information, see Set the system property to control when the Automated Test Framework captures screenshots
Form preferences panel: Run scheduled tests only	Click to toggle between: • On (green): Use this client test runner to run only scheduled tests and suites. • Off (gray): Use this client test runner to run only manually-started tests and suites.

Scheduled client test runner

The Scheduled Client Test Run opens a browser window for running scheduled client automated tests.

For information about scheduling automated tests, see Schedule an automated test suite and Working with scheduled test suites. You can toggle a client test runner to act as either a manual or scheduled client test runner.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Fields on the Client Runner Test window

Field / UI Element	Description
Form preferences icon ( )	Click to display the form preferences panel.
Form preferences panel: Screenshots mode	Choose among: • Enable for all steps • Enable for failed steps • Disable for all steps For additional information, see Set the system property to control when the Automated Test Framework captures screenshots
Form preferences panel: Run scheduled tests only	Click to toggle between: • On (green): Use this client test runner to run only scheduled tests and suites. • Off (gray): Use this client test runner to run only manually-started tests and suites.

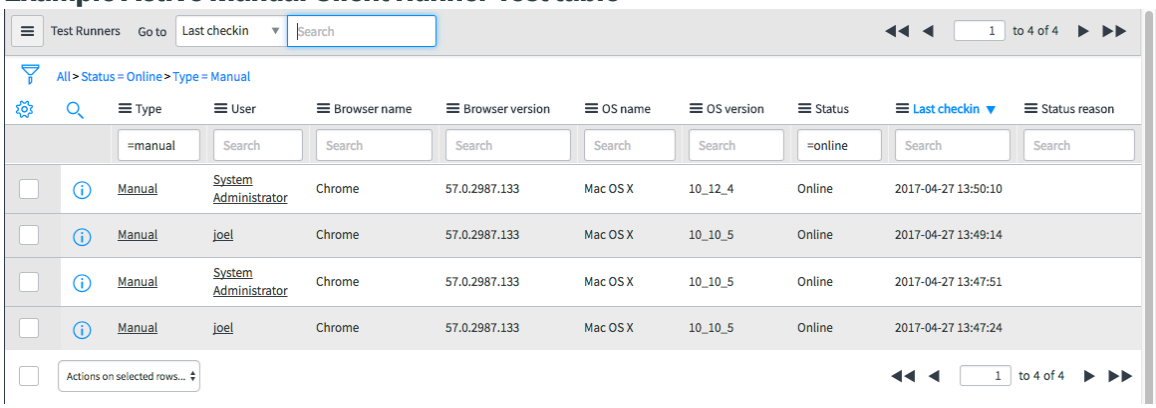
Active manual test runners

View the client test runners table filtered to show only those runners available to run manually-started tests.

When you start a manual client test runner, the system registers that runner in the Test Runners table as active.

The data in this table is transient. While the runner is active, it reports in to the system at a specified interval. If the runner does not report in at the expected time, the system marks the runner as inactive. After a period of time the system deletes the runner. You can modify these intervals on the Automated Test Framework properties page.

Example Active Manual Client Runner Test table



Fields on the Client Runner Test table

Field / UI Element	Description
Type	Whether this test runner is for manual or scheduled tests. In the Active Manual Test Runners module, Type is always **Manual**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Fields on the Client Runner Test table (continued)

Field / UI Element	Description
User	The user logged into the browser session.
Browser name	The browser name.
Browser version	The browser version.
OS name	The name of the operating system running the browser.
OS version	The version of the operating system running the browser.
Status	Whether this runner is currently online or offline. In the Active Manual Test Runners module, Status is always **Online**.
Last checkin	The time/date this runner most recently reported in to the system.
Status reason	If the Status is Offline, the reason why. In the Active Manual Test Runners module, Status reason is always empty.

Active scheduled test runners

View the client test runners table filtered to show only those runners available to run tests to be started by a schedule.

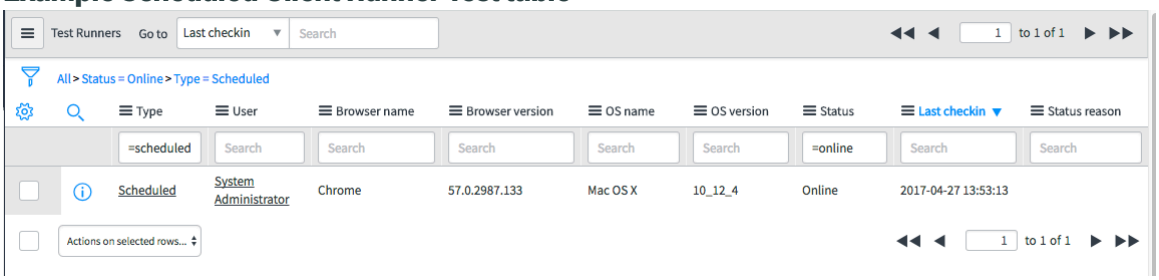
When you start a scheduled client test runner, the system registers that runner in the Active Scheduled Test Runners table.

The Active Scheduled Test Runner module is useful when you create a scheduled suite run. For scheduled suite runs, you can specify the browser to use. To determine the name and version of a browser you want to use, start a scheduled test runner with that browser, then inspect that runner's record in the Active Scheduled Test Runners module.

The data in this table is transient. While the runner is active, it reports in to the system at a specified interval. If the runner does not report in at the expected time, the system marks the runner as inactive. After a period of time the system deletes the runner.

You can modify these intervals on the Automated Test Framework properties page.

Example Scheduled Client Runner Test table



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Fields on the Client Runner Test table

Field / UI Element	Description
Type	Whether this test runner is for manual or scheduled tests. In the Active Scheduled Test Runners module, Type is always **Scheduled**.
User	The user logged into the browser session.
Browser name	The browser name.
Browser version	The browser version.
OS name	The name of the operating system running the browser.
OS version	The version of the operating system running the browser.
Status	Whether this runner is currently online or offline. In the Active Manual Test Runners module, Status is always Online.
Last checkin	The time/date this runner most recently reported in to the system.
Status reason	If the Status is Offline, the reason why. In the Active Scheduled Test Runners module, Status reason is always empty.

Waiting/running test runs

The Waiting/Running Test Run module opens a list of records showing the tests waiting to be run.

Note: To prevent conflicts, the system allows only one test to run at a given time. This is true even if you have multiple client test runner windows open. If you submit tests to run when another test is already running, the system holds the new tests in a queue to run later. If a test remains in the queue for more than ten minutes, the system cancels the test.

You can cancel execution of a waiting automated test.

Waiting/running suite runs

The Waiting/Running Suite Runs module opens a list of records showing the test suites waiting to be run.

Allowed client error records

Review the list of existing Allowed Client Error [sys_atf_whitelist] records to see which client errors produce warnings and which are ignored. Modify existing Allowed Client Error records as needed or create new ones.

Allowed Client Error form

Field	Description
Report level	Report action to take when the client error is encountered. Options include:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Allowed Client Error form (continued)

Field	Description
	• **Warning – Step & Test will report Success with Warning(s)**: Test steps containing the allowed client error report a status of `Success with warning(s)`. The error message appears in the test result output, and is recorded in the test logs with the status `Warning`. • **Ignored – Step & Test will report Success**: Test steps containing the allowed client error report a status of `Success`. The error is recorded in the test logs with an `Ignored` status.
Active	Check box to enable or disable allowing of a client error.
Error message	Message of the client error you want to allow.
Description	Description of the error you want to allow. If this client error was allowed from a test result, step result, or a test log, the test log description is copied into this field.

Related topics

Allowed client errors

Reported client errors

The Reported Client Error module lists test logs across all tests that are client errors and have failed. You can review individual test log records, and allowed client errors as needed to prevent them from adversely impacting future test executions.

Field	Description
Start time	Time at which this test step started executing.
Step	Name of the step executed.
Status	The result of the test log. Failure: Test step run failed.
Output	Output generated for the test log. For a client error, the actual client error is displayed.
Duration	Time duration it took to execute this test step.
Type	Type of test log. Client Error **Note:** You can optionally add client errors as an ignored or warning entry in the Allowed Client Errors. Doing so prevents the allowed client errors from affecting ATF test executions when they recur in future test runs. For more details, see Allowed client errors.
Recorded at	Time at which this step or log entry was recorded.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Related topics

- Allowed client errors
- Manually allow client errors
- Identify and resolve client errors

Automated Test Framework use case examples

Use cases can help you construct tests for common scenarios.

Automated Test Framework use case: test basic form operations

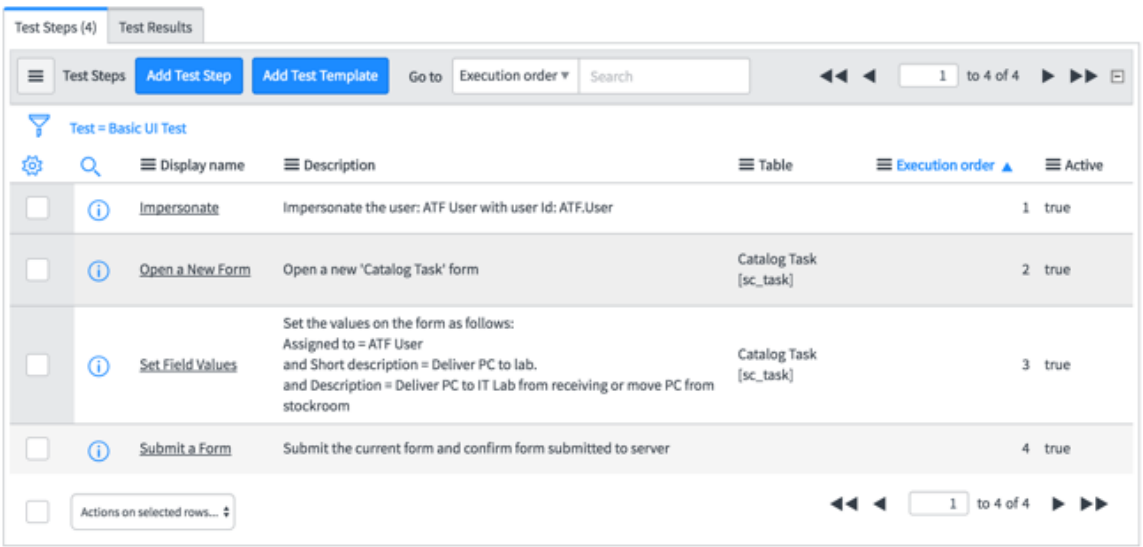
This use case illustrates testing basic form operations with the Automated Test Framework.

Before you begin

Role required: atf_test_admin

About this task

Steps in test

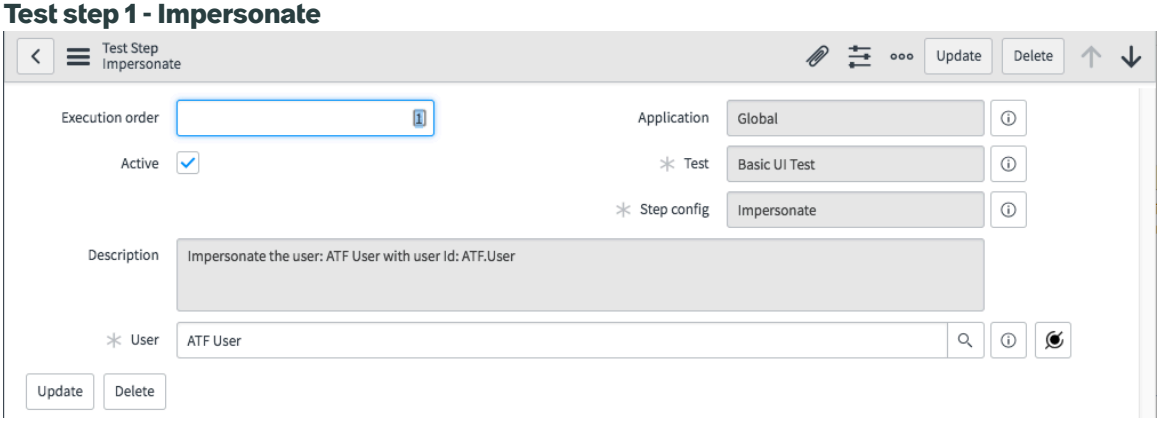


Procedure

1. Impersonate a user with the permissions needed to perform these steps, in this example ATF.User.

Example

Test step 1 - Impersonate

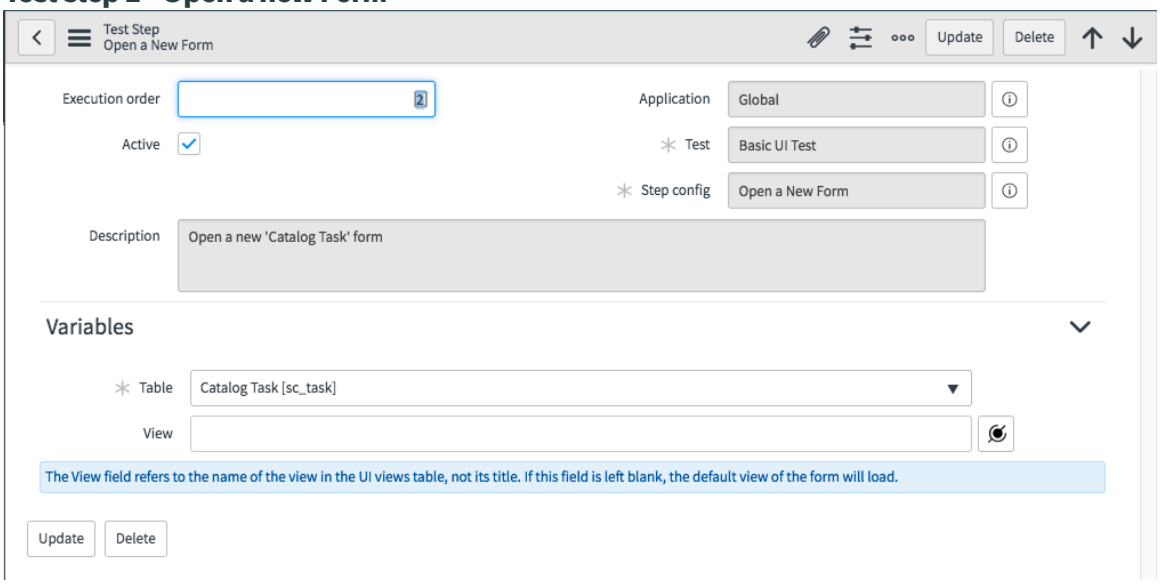


© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

2. Open a form, in this example a Catalog Task form.

Example

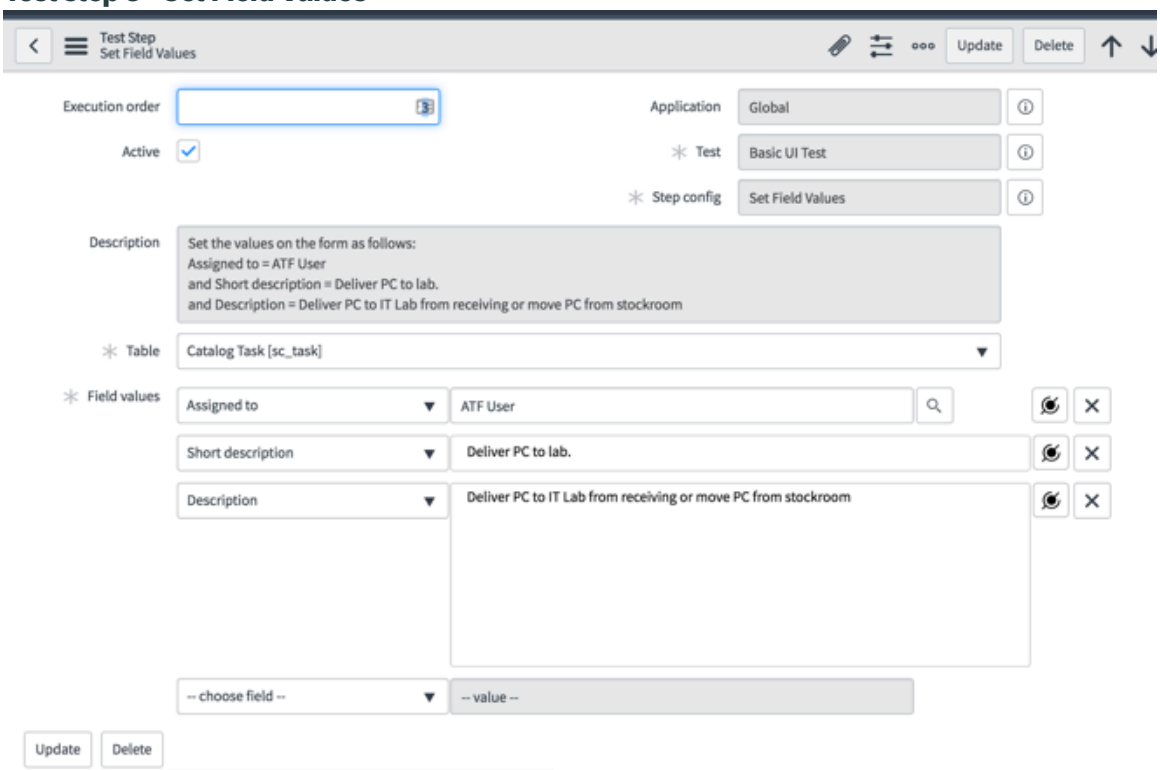
Test step 2 - Open a new Form



3. On the open form, set field values, including for any mandatory fields.
This example sets field values for **Assigned to**, **Short description**, and **Description**.

Example

Test step 3 - Set Field Values

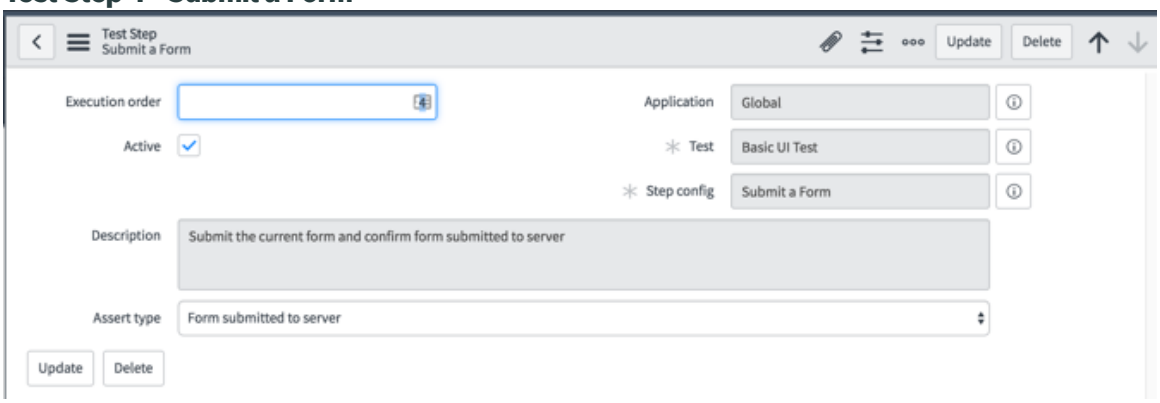


© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

4. Submit the open form.

Example

Test Step 4 - Submit a Form



Automated Test Framework use case: reference a value from a previous step

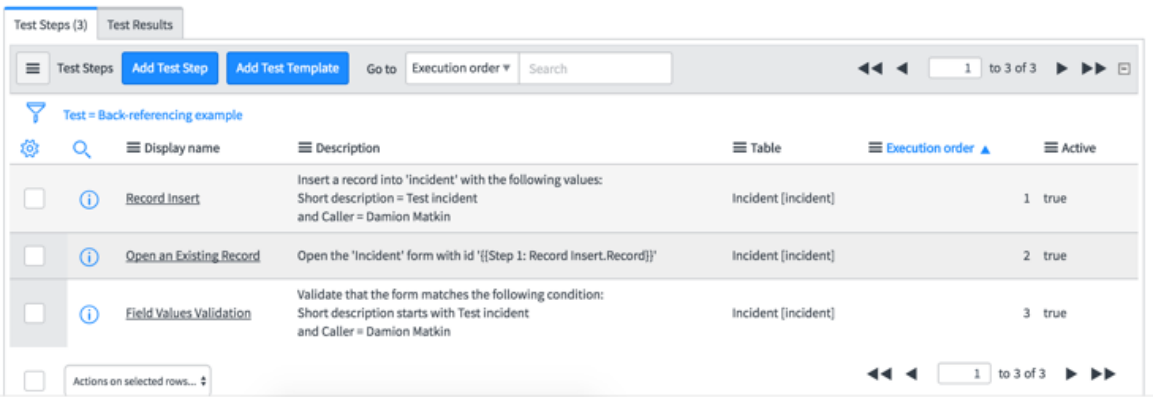
This use case illustrates assigning a form field the value of an output variable from a previous step.

Before you begin

Role required: atf_test_admin

About this task

Automated Test Framework: In this example, the second step references an output value from the first step. Pass values from one step to another example.



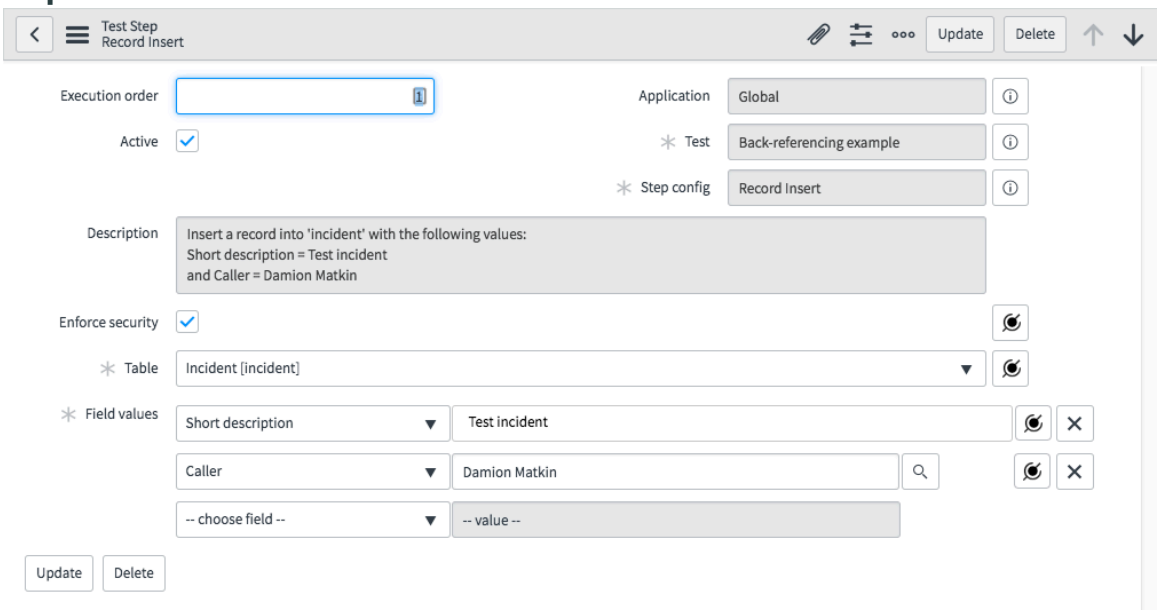
Procedure

1. Insert a record into the incident table.
   This example inserts a record into the Incident table.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example

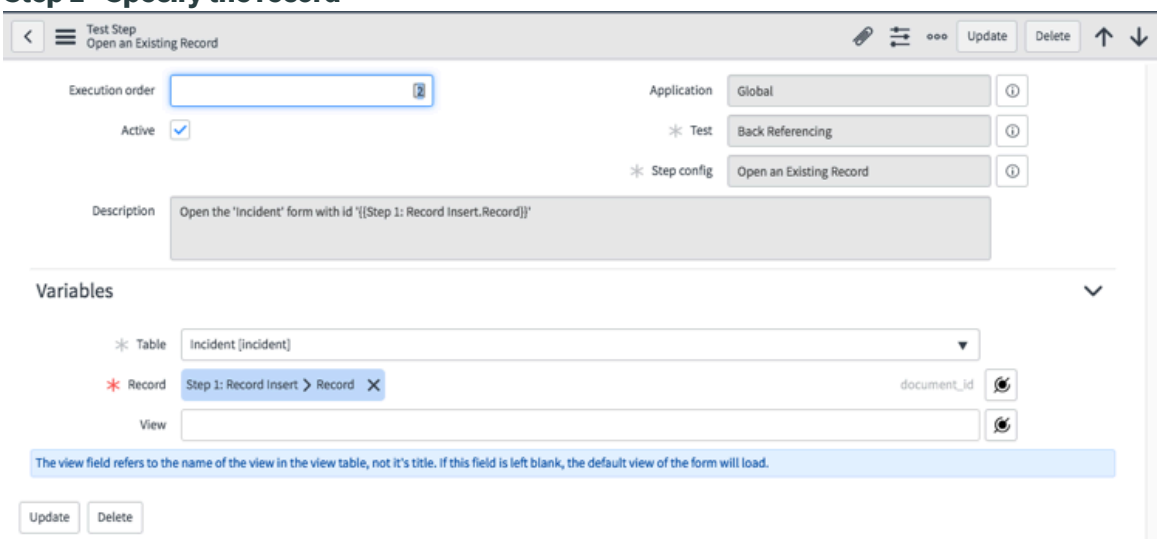
Step 1 - Record Insert



2. Open the record just inserted by assigning to the *Record* field the output variable from Step 1.

Example

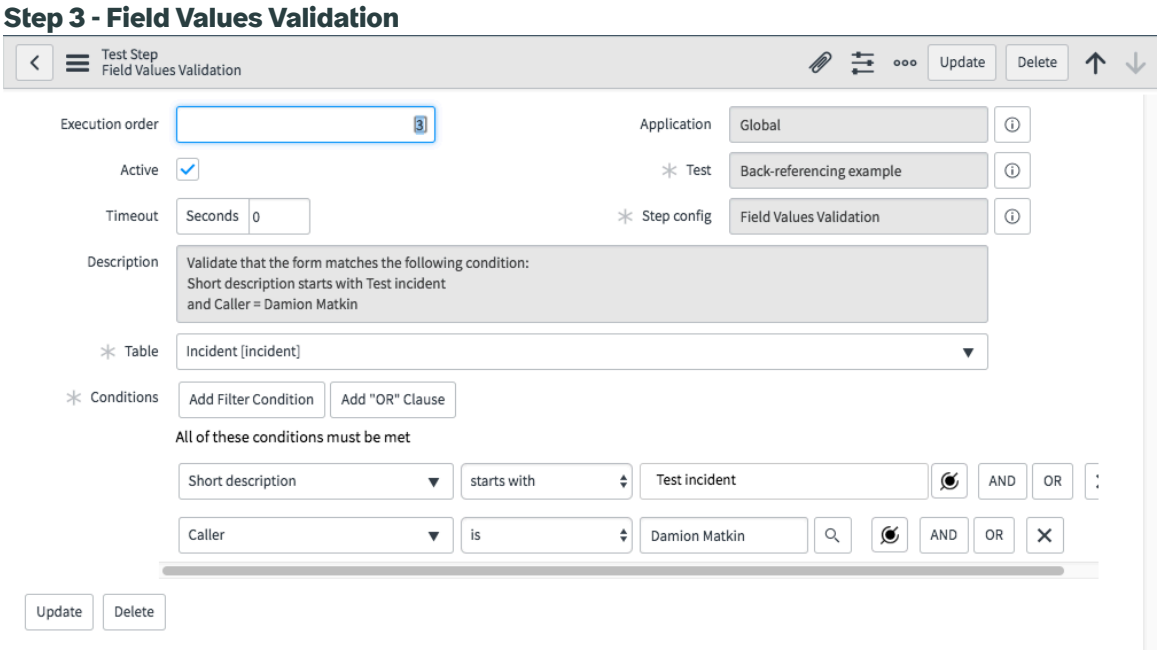
Step 2 - Specify the record



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

3. Validate that fields on the open record have the values you expect.

Example



Related topics

Pass values from one automated test step to another

Automated Test Framework use case: test a business rule

This use case illustrates testing a business rule with the Automated Test Framework.

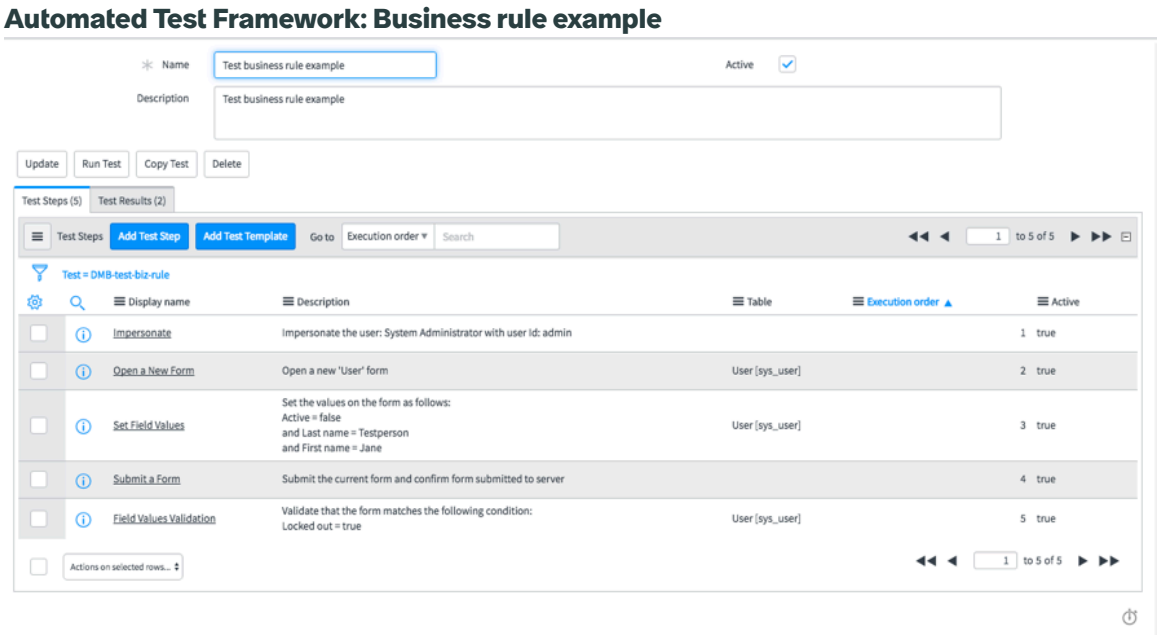
Before you begin

Role required: atf_test_admin

About this task

This example tests a business rule that sets the value of **Locked out** to **true** when **active** is set to **false**.

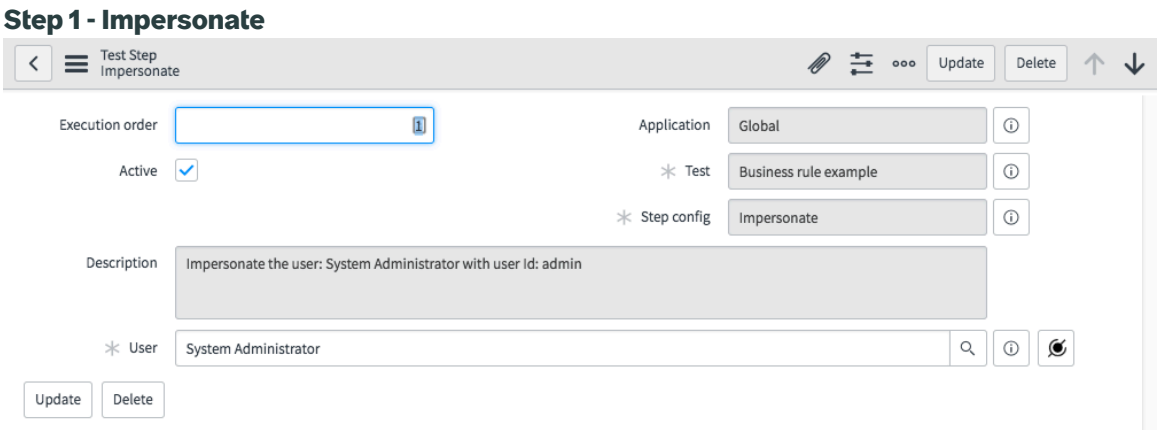
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Procedure

1. Impersonate a user with the necessary permissions.
   In this example, the step impersonates the admin user.

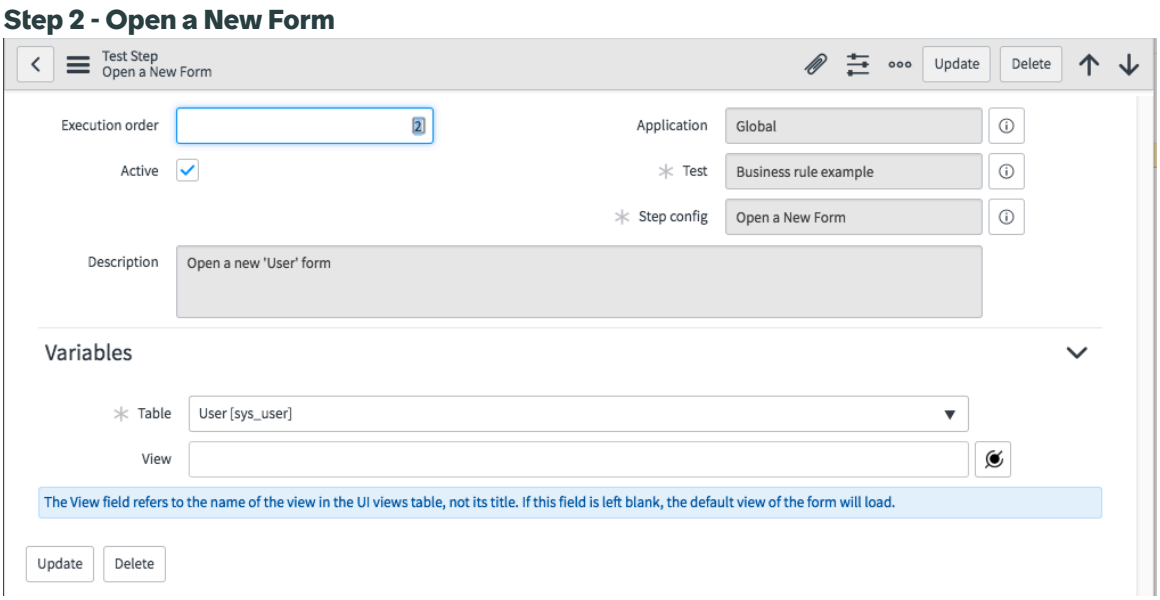
   Example

2. Open a form for the table to which this business rule applies.
   This example opens a new User form.

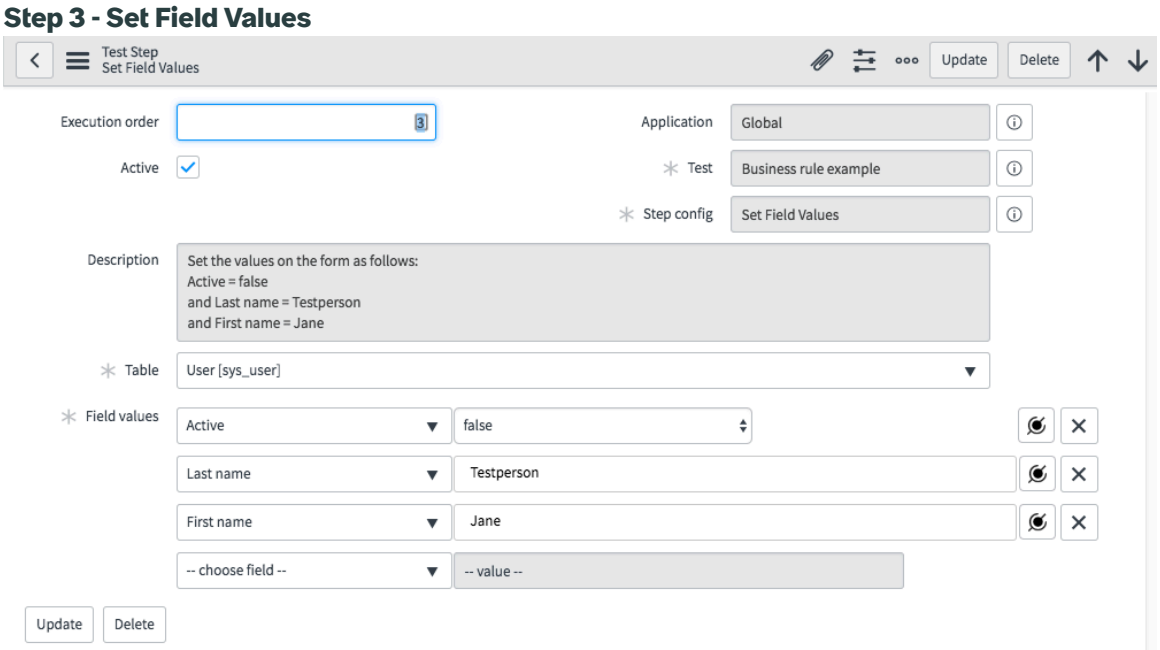
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example



3. Set values on the form that meet the requirements for submitting the form and for triggering the business rule.
   This example sets values for the **Active**, **Last name**, and **First name** fields.

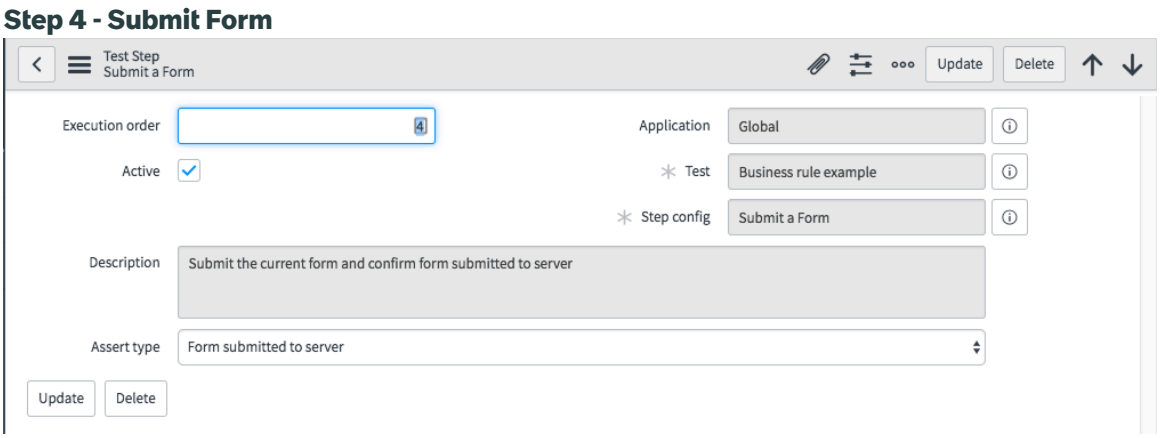
Example



4. Submit the form.

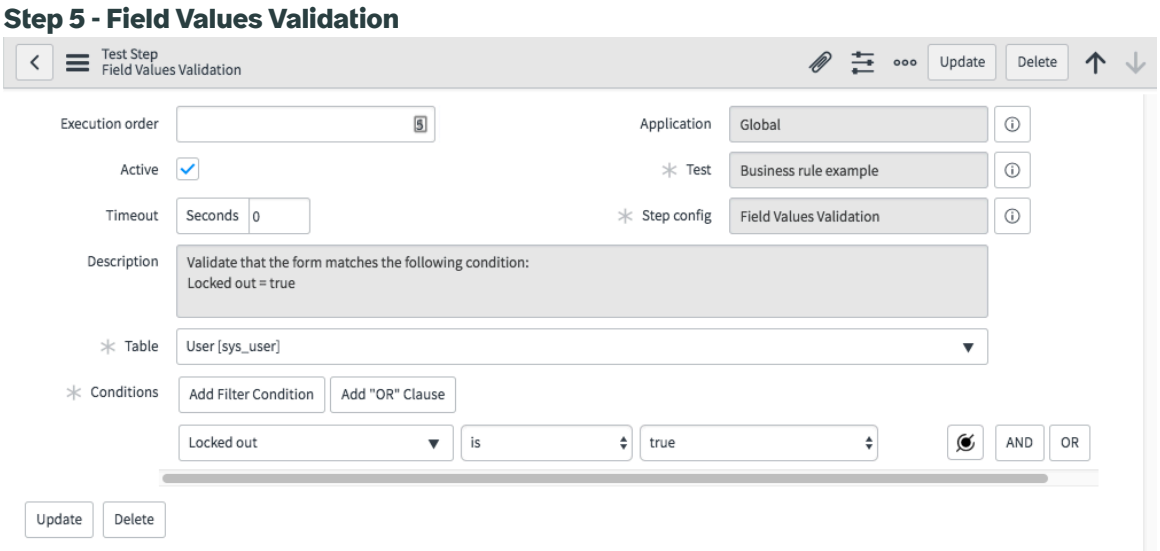
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example



5. Validate that the business rule ran.
   In this example the business rule tested sets **Locked out** to **true** if **Active** is set to **false**.

Example



Automated Test Framework use case: test a data policy

This use case illustrates testing a data policy with the Automated Test Framework.

Before you begin

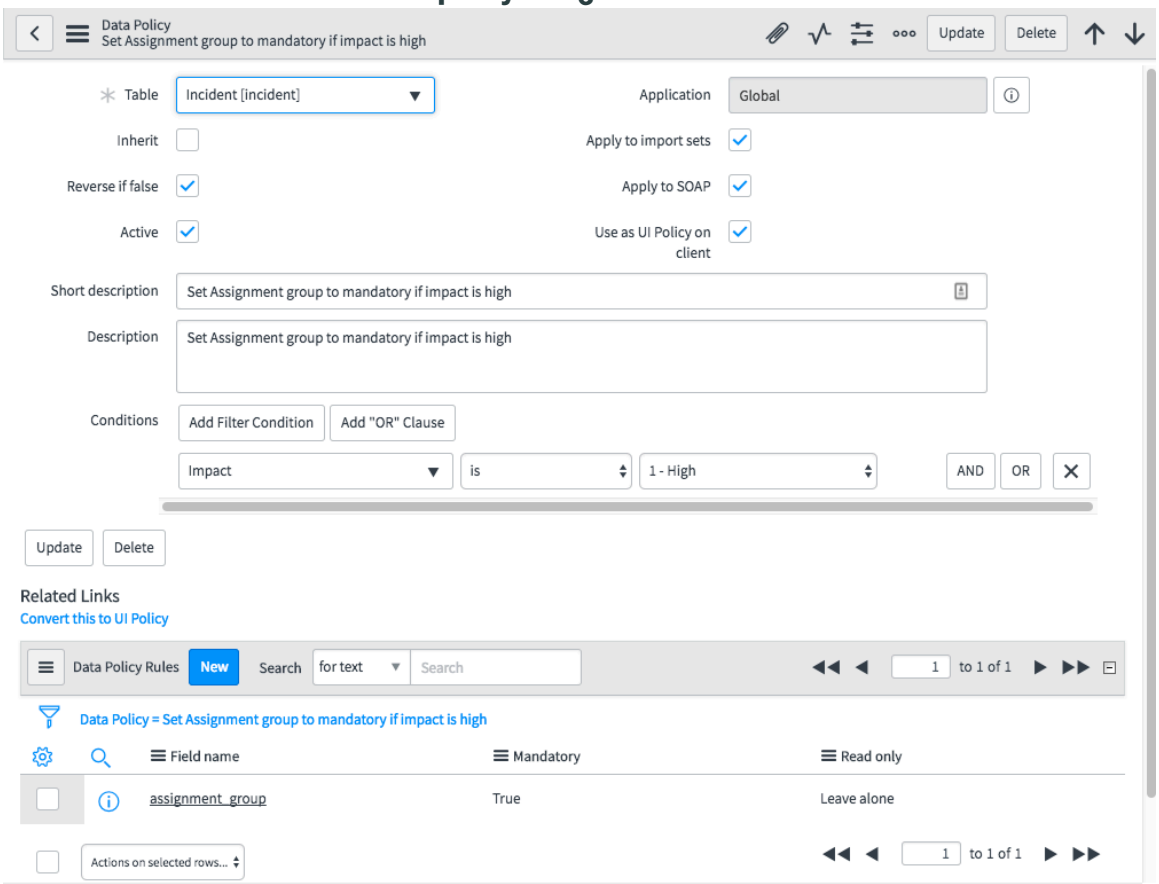
Role required: atf_test_admin

About this task

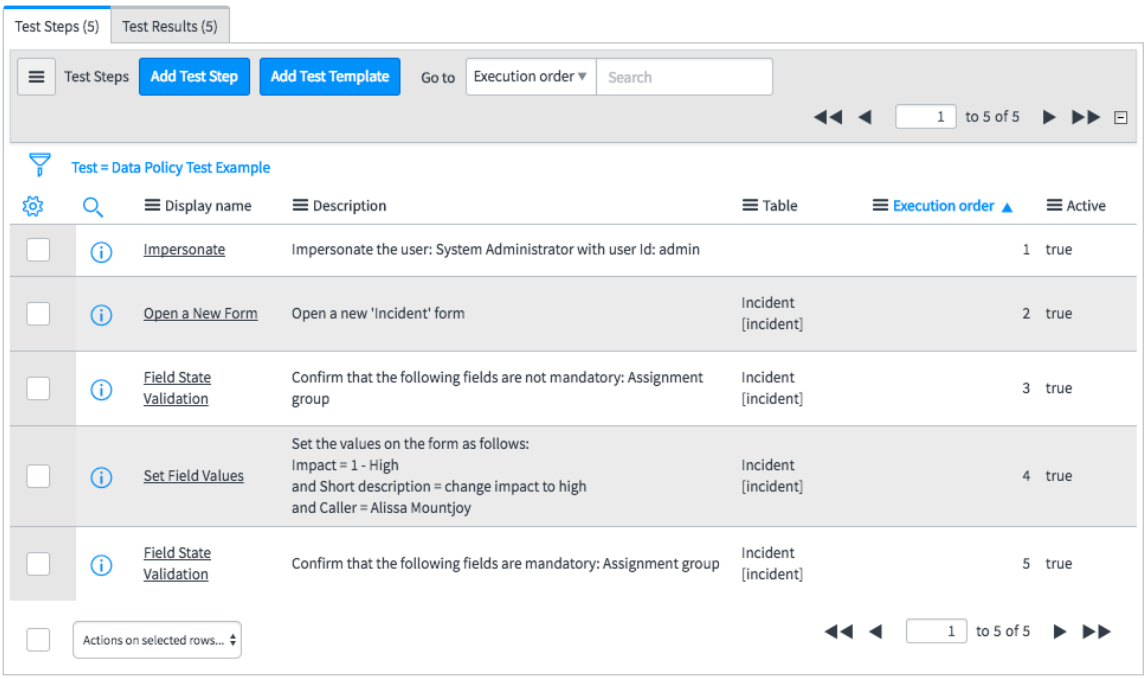
This example tests the data policy that sets the field **Assignment Group** to **mandatory** if **impact** is **high**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Automated Test Framework: Data policy being tested



Automated Test Framework: Data policy example



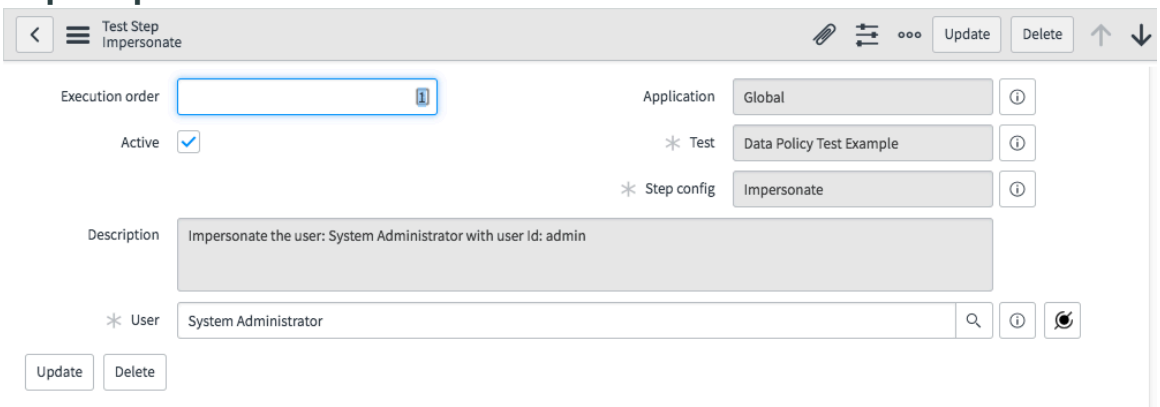
Procedure

1. Impersonate a user with the necessary permissions.
   In this example, the step impersonates an admin user.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example

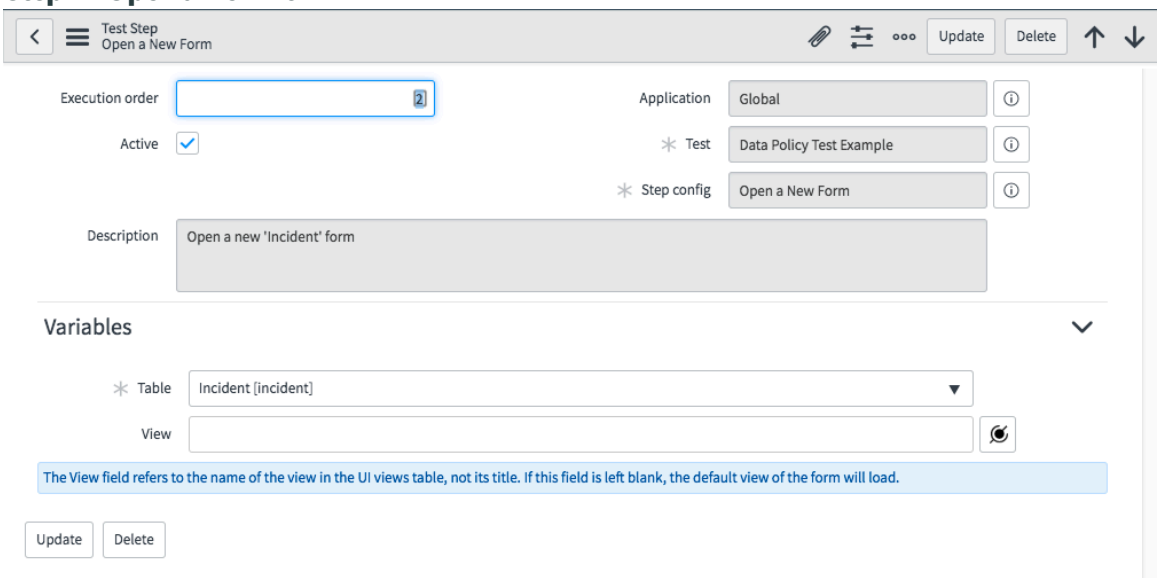
Step 1 - Impersonate



2. Open a form for the table to which this data policy applies.
   This example opens a new incident form.

Example

Step 2 - Open a New Form

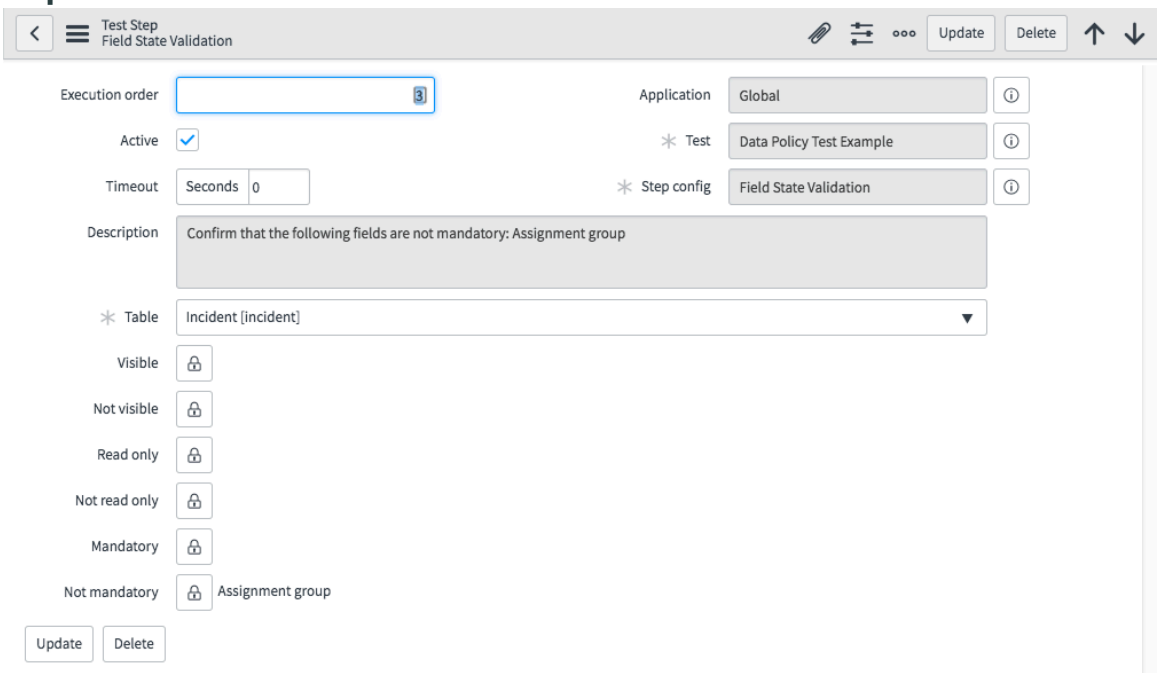


3. Check that the data policy has not yet been triggered.
   In this example, the step checks to confirm that **Assignment group** is not **mandatory**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example

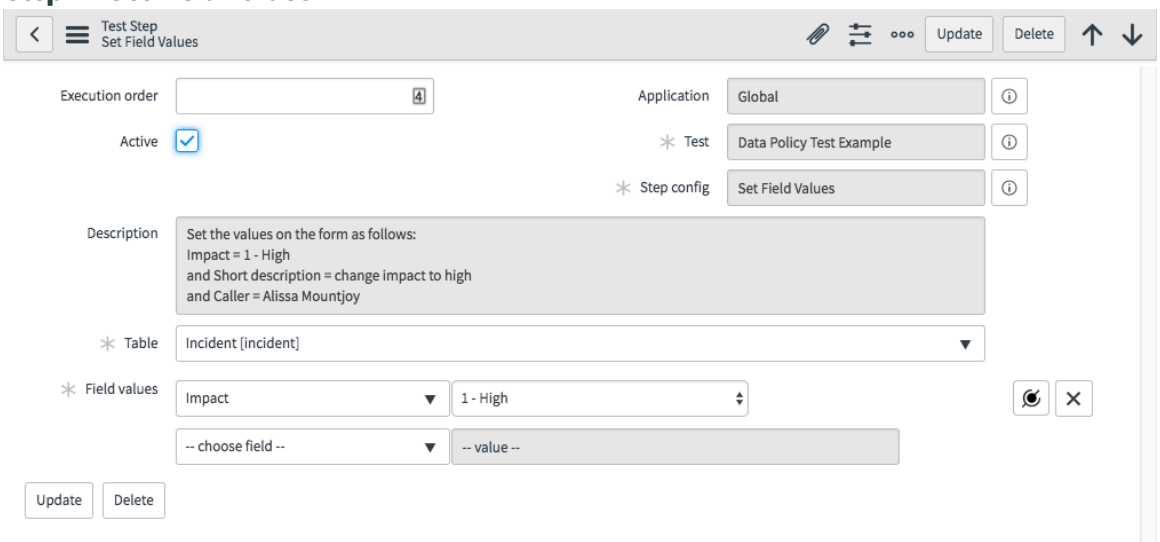
Step 3 - Field State Validation



4. If applicable, set the conditions that trigger the data policy.
   This example sets **Impact** to **High**.

Example

Step 4 - Set Field Values



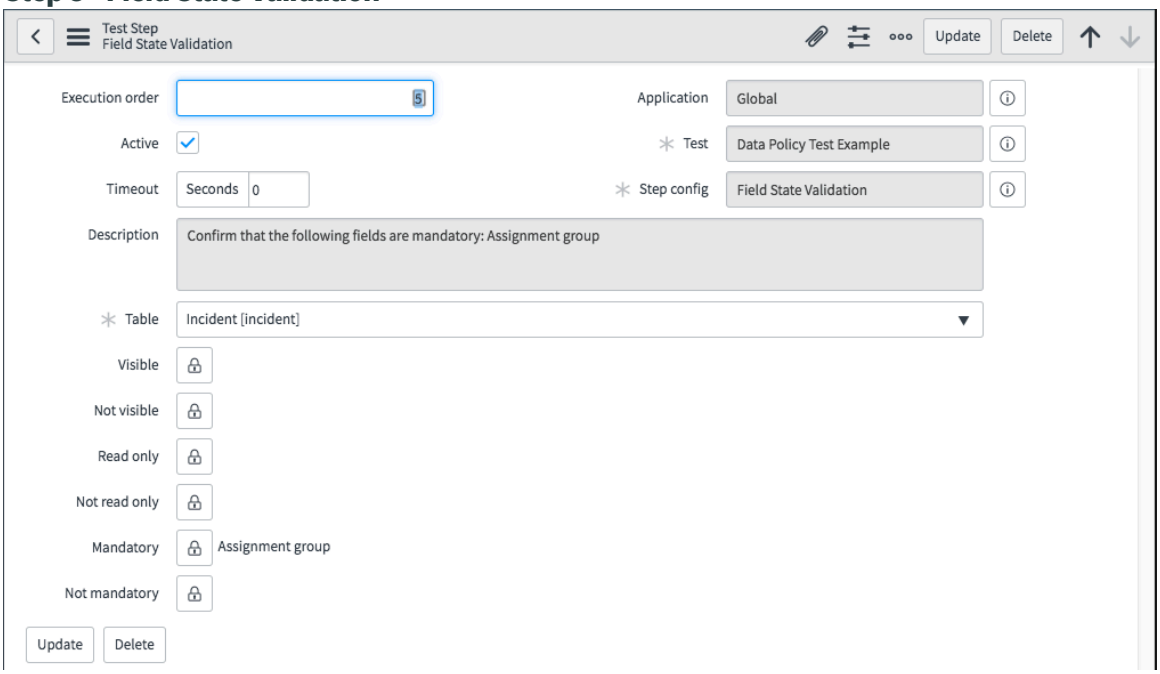
5. Validate that the data policy is enforced.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

In this example, the test confirms that the data policy set **Assignment group** to **High** after the previous step set **Impact** to **High**.

Example

Step 5 - Field State Validation



Automated Test Framework use case: test a script include

This use case illustrates testing a script include with the Automated Test Framework.

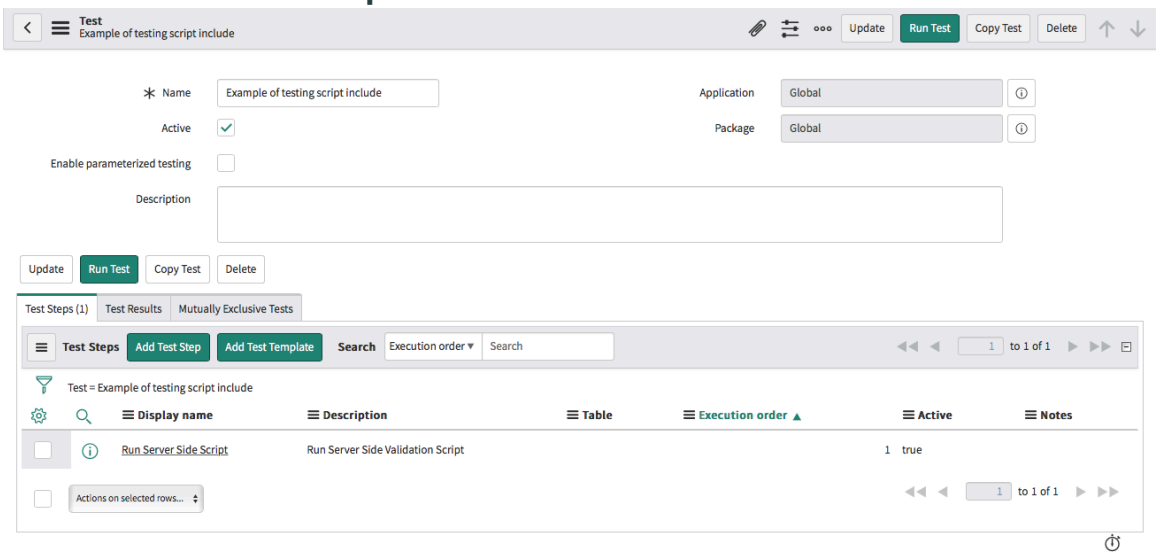
To test a script include with the Automated Test Framework, create a test that performs these steps:

- a step that causes the system to execute the script include. Examples:
  - Run Server Side Script test step that calls the script include.
  - Form action step script – such as open a form, submit or form, or set a field value – that invokes the script include.
- A step that validates that the script include took the expected actions. The specific test step for this validation depends on what the script include is designed to do. Examples:
  - Field values validation, if the script sets a field value
  - Field state validation, if the script changes a field state
  - Record Query, if the script generates a new record

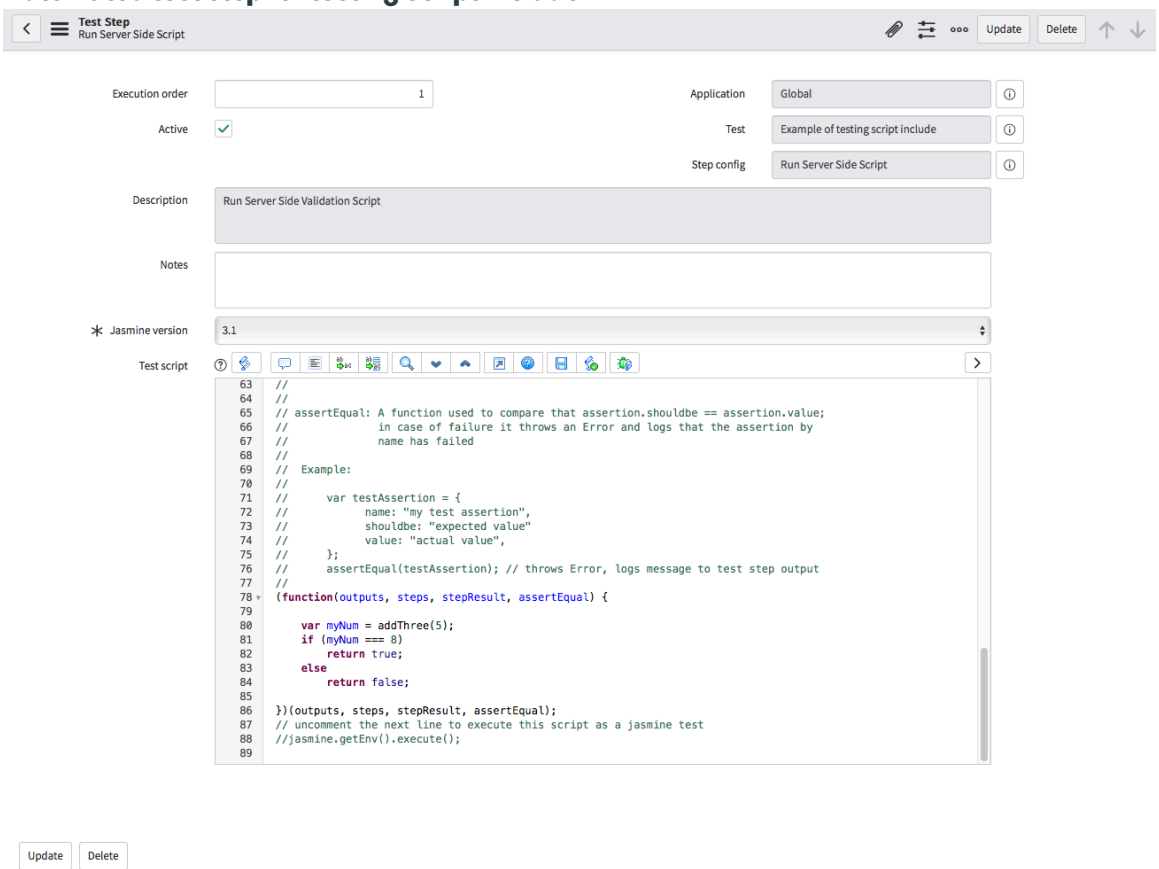
This example shows a test with one test step: Run Server Side Script. The script associated with this test step calls a script include that returns the value of its argument plus three. If the value returned from the script include is 8, the script include has worked as intended and the test passes.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Automated Test to test a script include

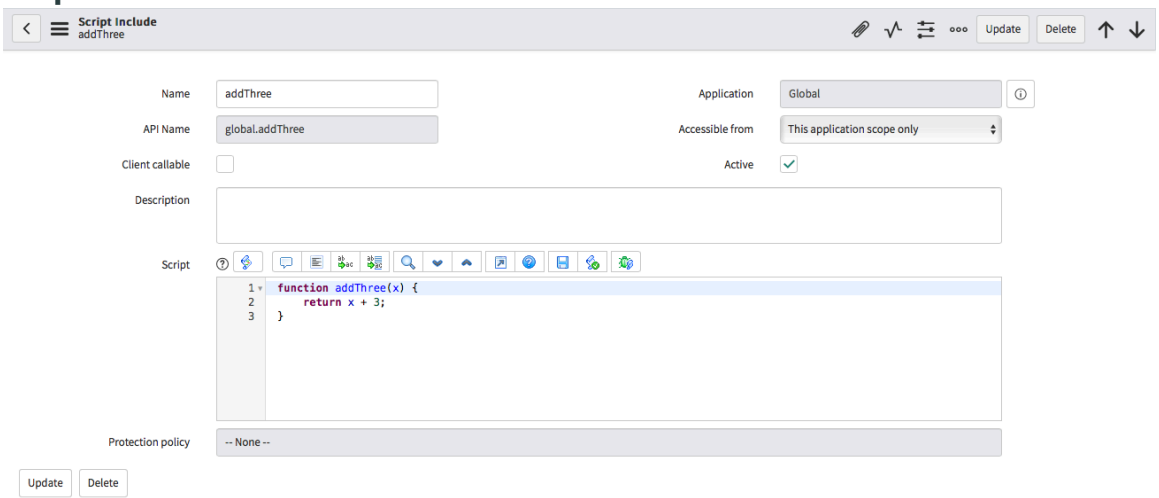


Automated test step for testing script include



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Script include to test with Automated Test Framework



Automated Test Framework use case: test a Service Catalog request

This use case illustrates testing a service catalog request with the Automated Test Framework.

Before you begin

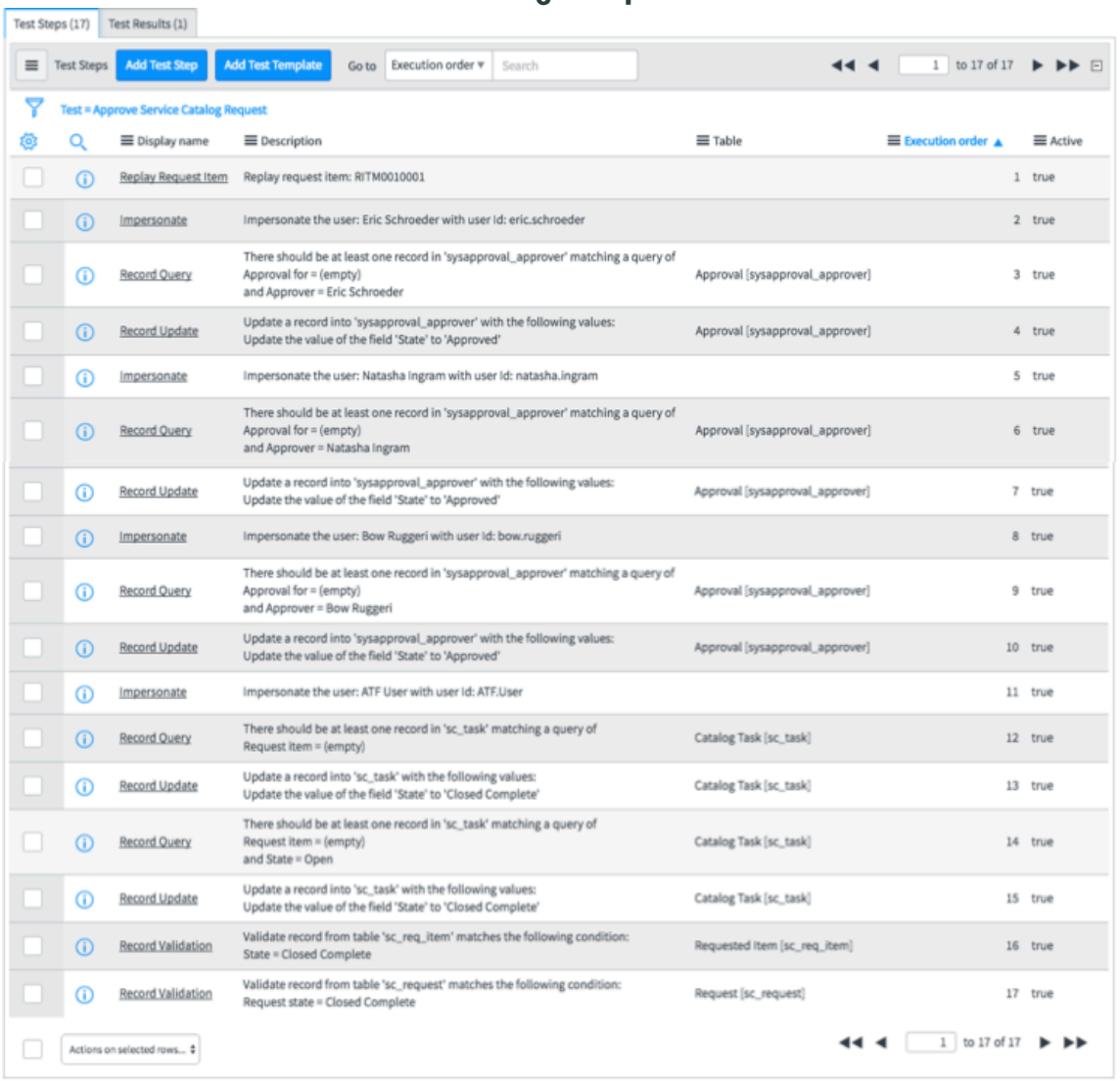
Role required: atf_test_admin

About this task

With the Replay Request Item test step, you can test the service catalog ordering process once a request exists and has a record in the request item table. In the Brazil release, you cannot create an automated test for the process by which the user creates a new request.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Automated Test Framework: Service catalog example



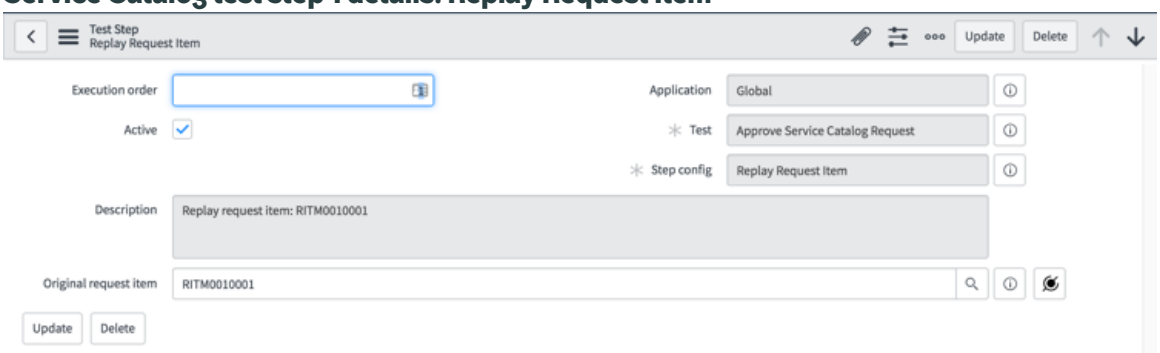
Procedure

1. Replay an existing service catalog request item.

   This test step inserts a new record in the [sc_request] table for the catalog request item RITM0010001.

   ### Example

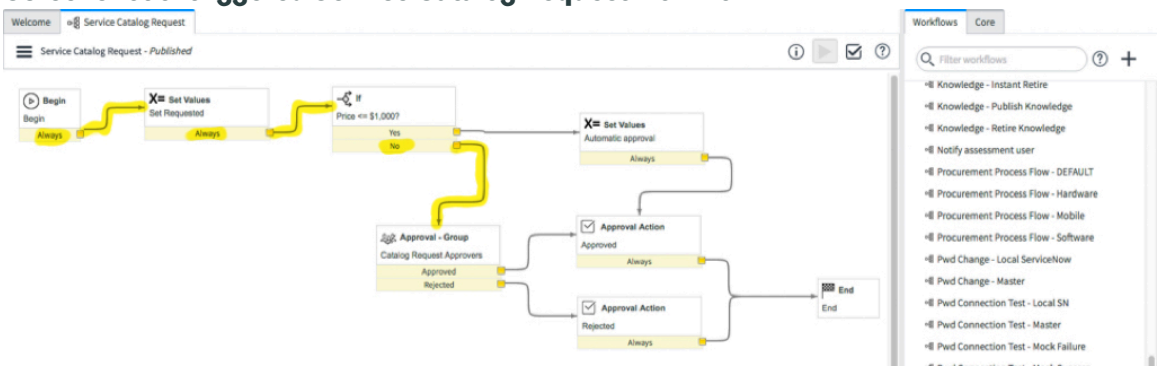
   #### Service Catalog test step 1 details: Replay Request Item



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

This insertion triggers the Service Catalog Request workflow, which checks the price of the item, determines that it exceeds $1000.00, and therefore generates approval records for users belonging to the Catalog Request Approvals group. In this example, only one user– Eric Schroeder – belongs to this group.

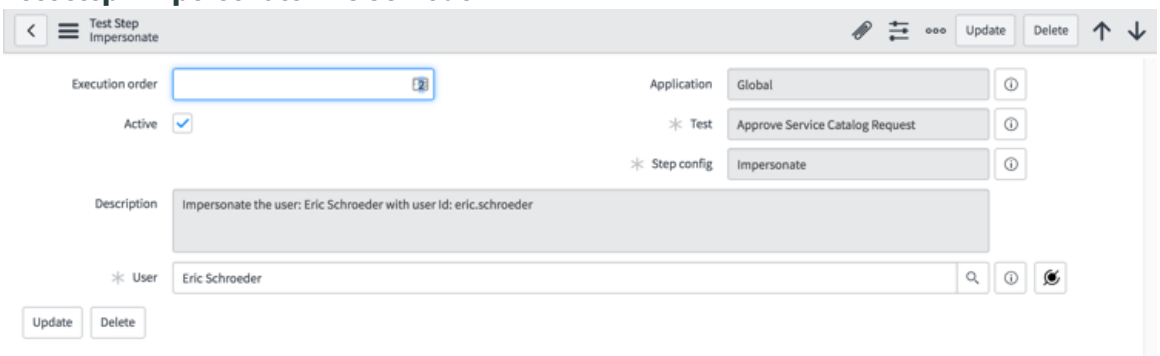
Screenshot of triggered Service Catalog Request workflow



2. Impersonate Eric Schroeder, the user who needs to approve this Service Catalog Request.

Example

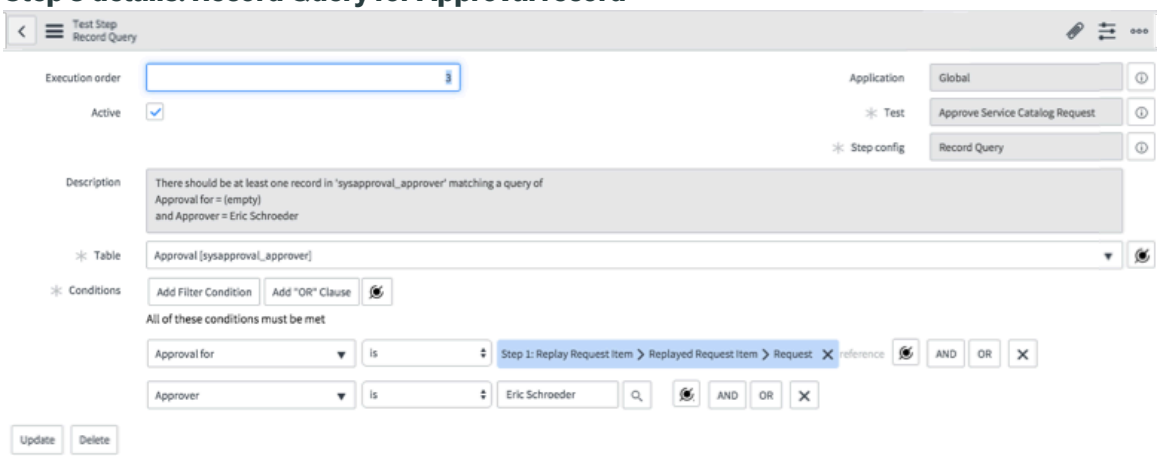
Test step - Impersonate Eric Schroder



3. Verify that the system created an approval record for Eric Schroeder and this request. Note that for the **Approval for** field, you assign the output value from Step 1.

Example

Step 3 details: Record Query for Approval record

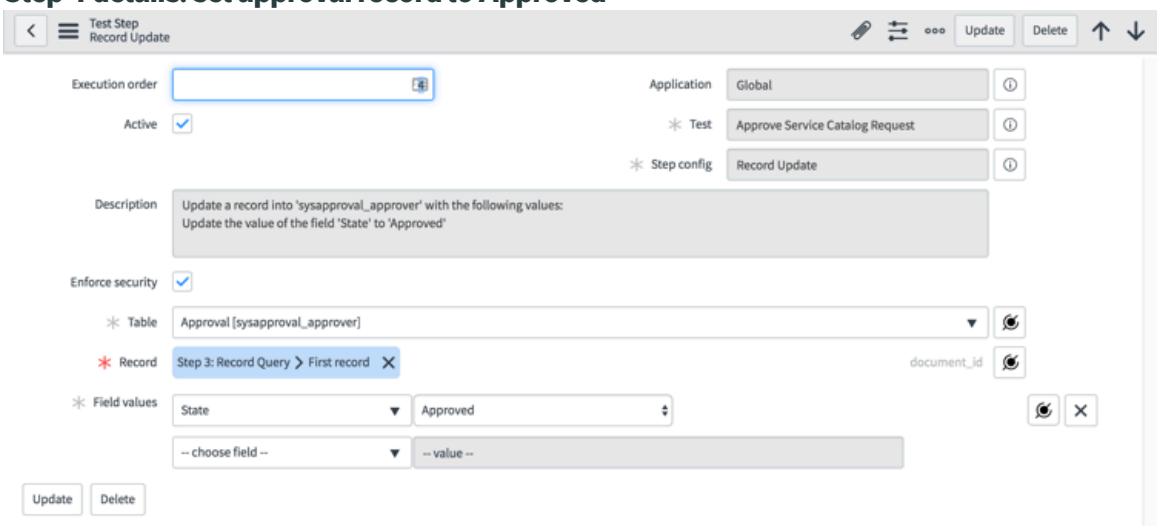


© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

4. Set the state of this approval record to Approved.

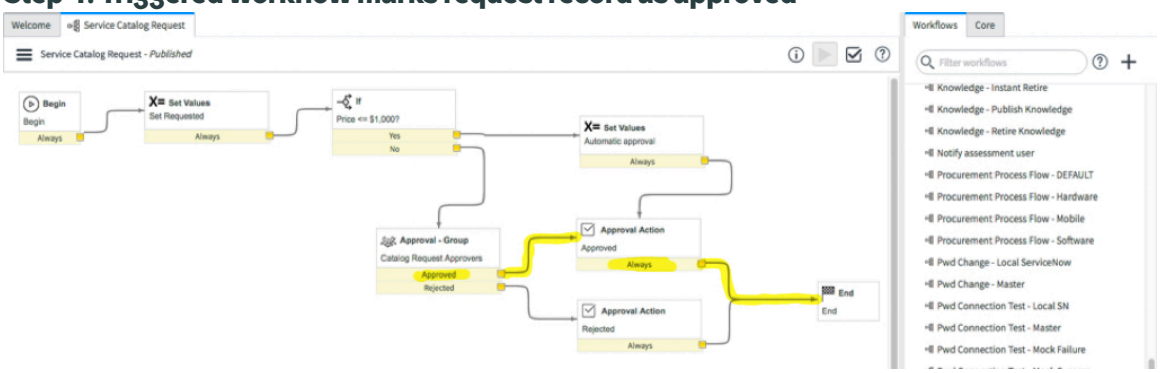
Example

Step 4 details: set approval record to Approved



The Service Catalog Request workflow sees that all required approval records have the state of Approved and transitions to the Approval Action which marks the request record [sc_request] as Approved.

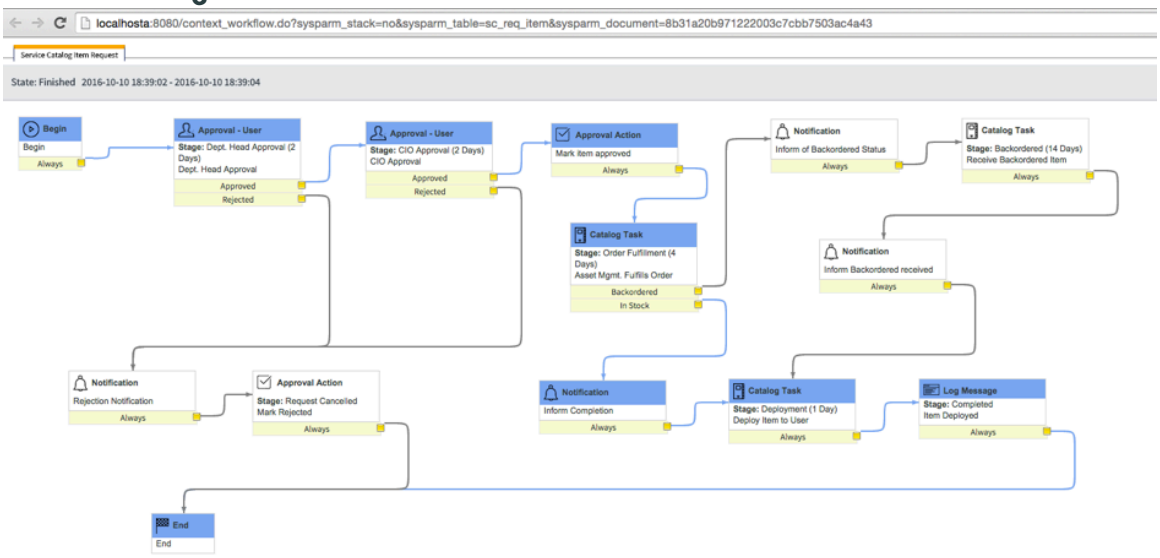
Step 4: Triggered workflow marks request record as approved



When the record in [sc_request] changes to the Approved state, an associated business rule generates request items [sc_request_item] for each item in the request. In this example, the request contains only one item, so the business rule inserts one record into the [sc_request_item] table. This insertion triggers the Service Catalog Item Request workflow.

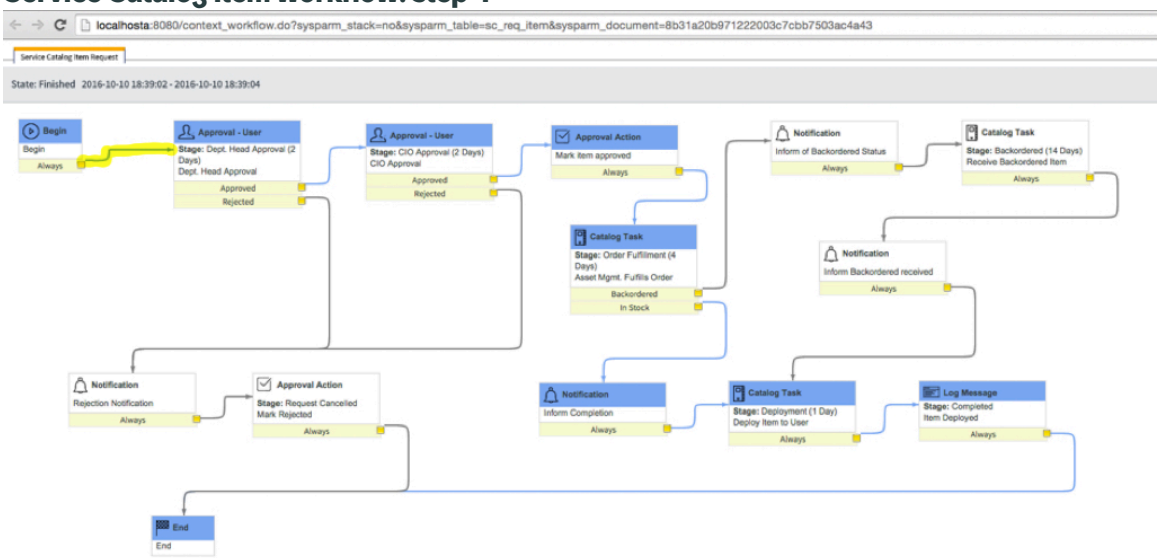
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Service Catalog Item workflow



The first activity in the Service Catalog Item Request workflow generates an approval record for the head of the department in which the requesting user works. In this example, the department head is Natasha Ingram.

Service Catalog Item workflow: step 4

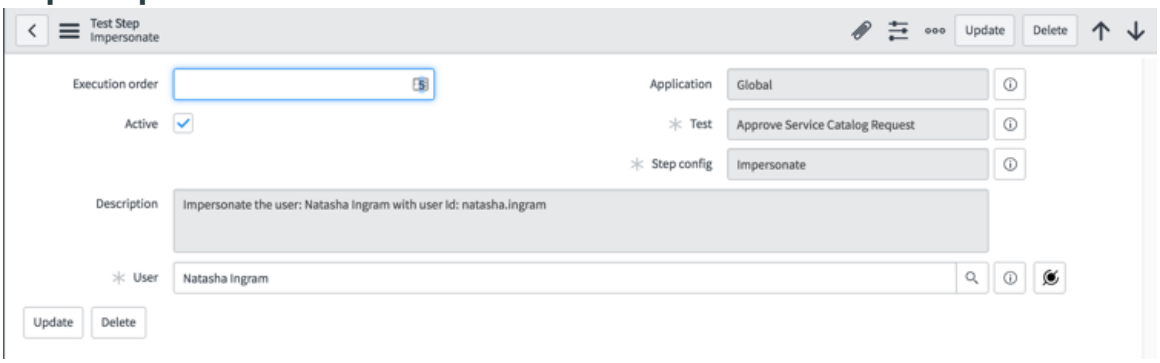


5. The workflow does not continue until the department head approves it, so the next test step impersonates Natasha Ingram.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example

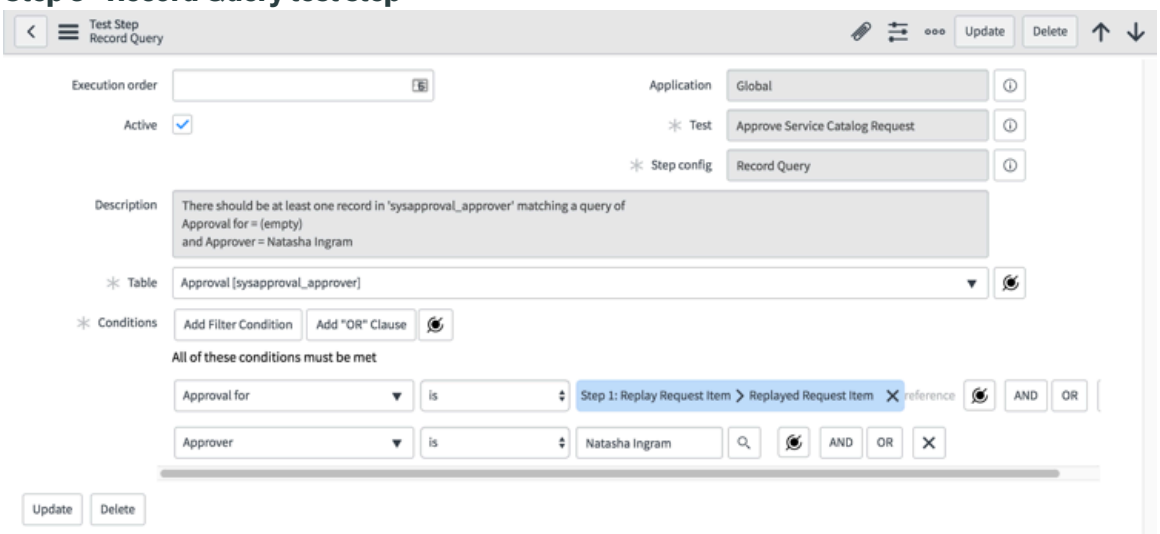
Step 5 - Impersonate User



6. Obtain the sys_id for the new approval record with the Record Query step.
   Note that Record Query creates an output variable with the sys_id of the first record returned from the query.

Example

Step 6 - Record Query test step

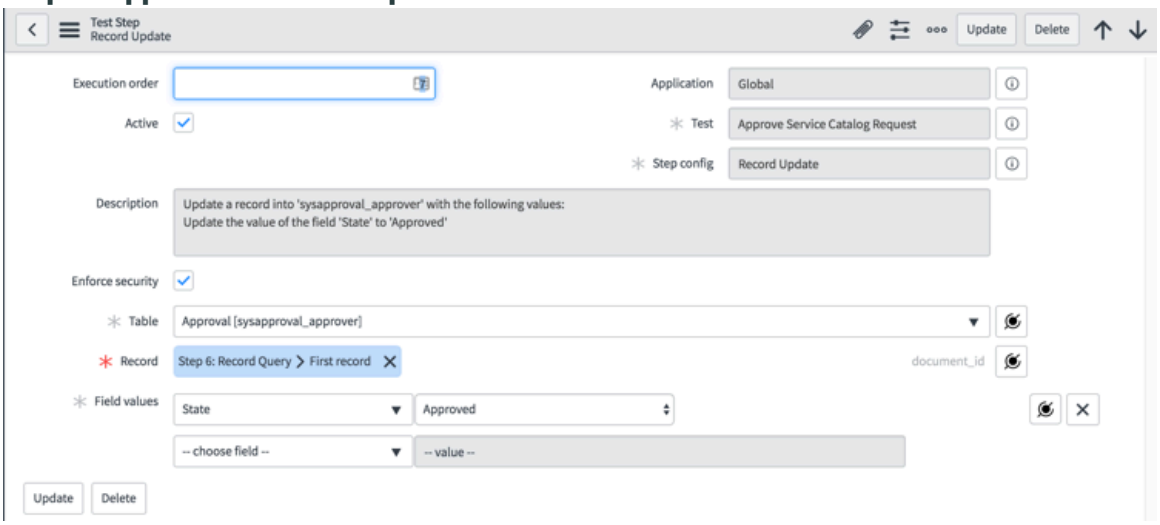


7. Set the approval record to Approved.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

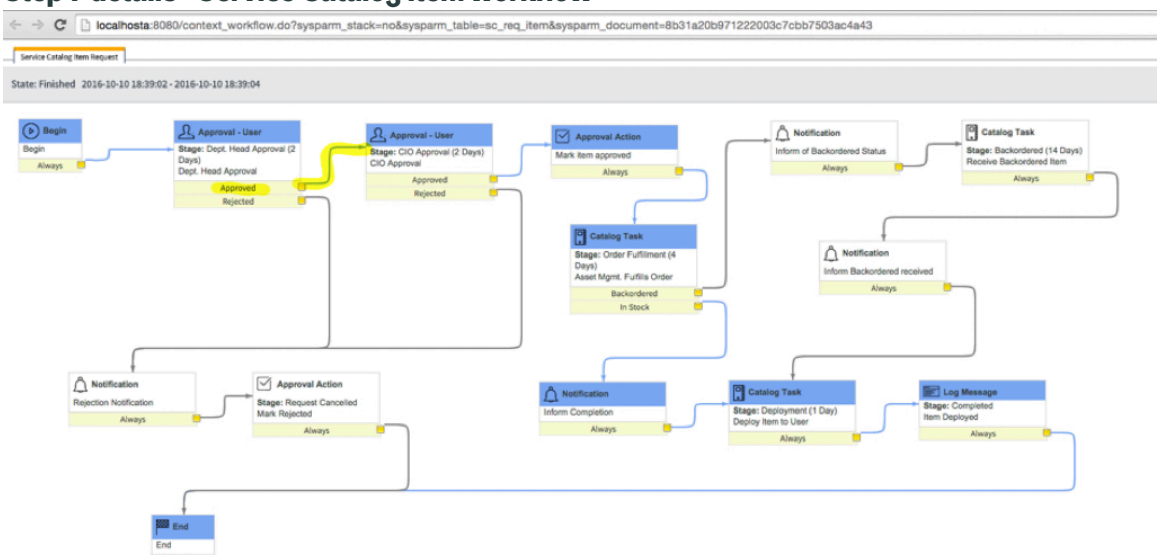
Example

Step 7 - Approval User test step



Note how Step 7 refers to the *`First record`* output variable from Step 6 to specify which record to approve. When the record is approved, the workflow transitions to the next Approval - User activity, which generates an approval record for the CIO. In this example, the CIO is Bow Ruggeri.

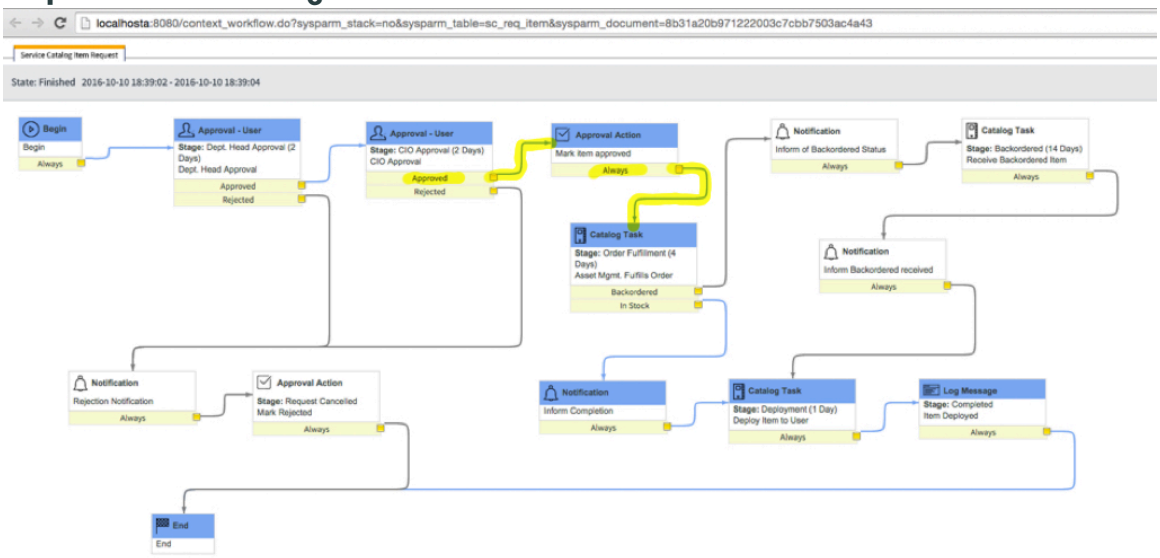
Step 7 details - Service Catalog Item workflow



8. Impersonate Bow Ruggeri.
9. Obtain the sys_id for the approval record for Bow Ruggeri.
10. Set the approval record to Approved.
    When the record is approved, the workflow transitions to the Approval Action activity which sets the record for this item in the [sc_request_item] table to *`Approved`*. The workflow transitions to the Catalog Task activity labelled Asset Mgmt. Fulfills Order. This Catalog Task activity generates a new record in the [sc_task] table that instructs a user in the Fulfillment group to order the item.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

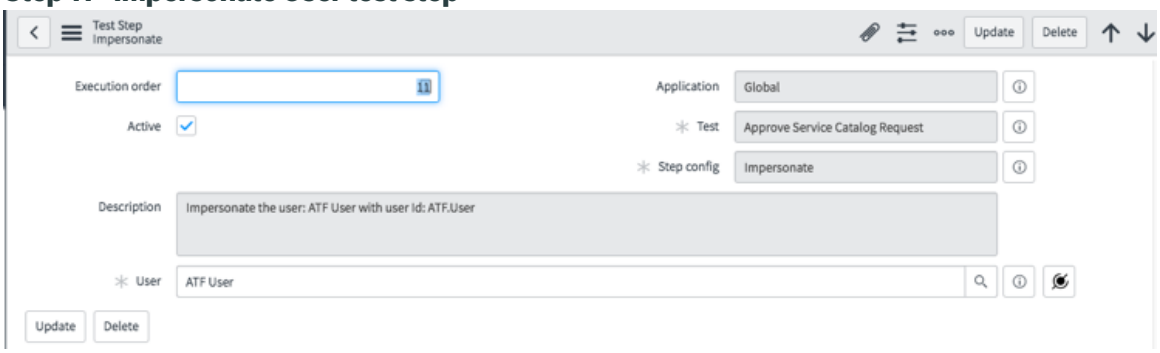
Step 10 - Service Catalog Item workflow



11. Impersonate a user in the Fulfillment group, in this example ATF.User.

Example

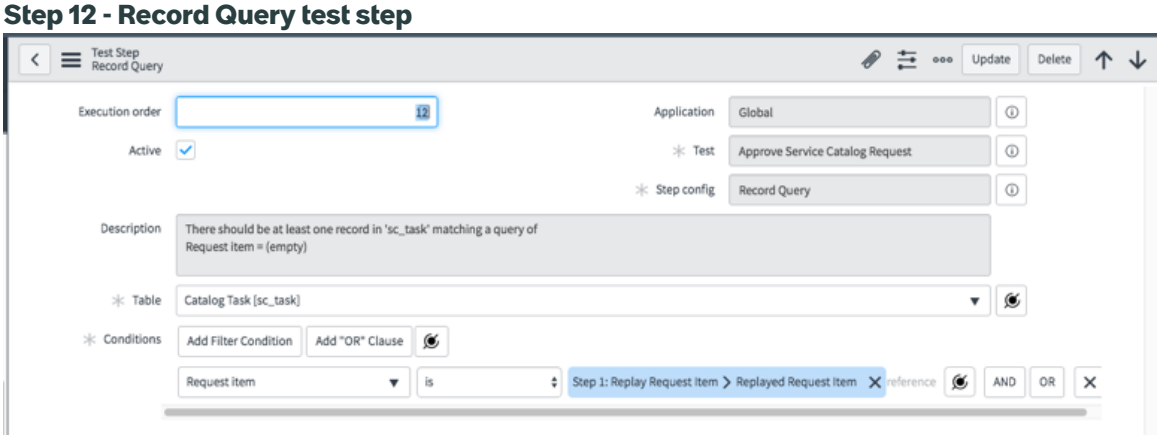
Step 11 - Impersonate User test step



12. Obtain the sys_id for the new catalog task with the Record Query step.
Note that Record Query creates an output variable with the sys_id of the first record returned from the query.

Example

Step 12 - Record Query test step

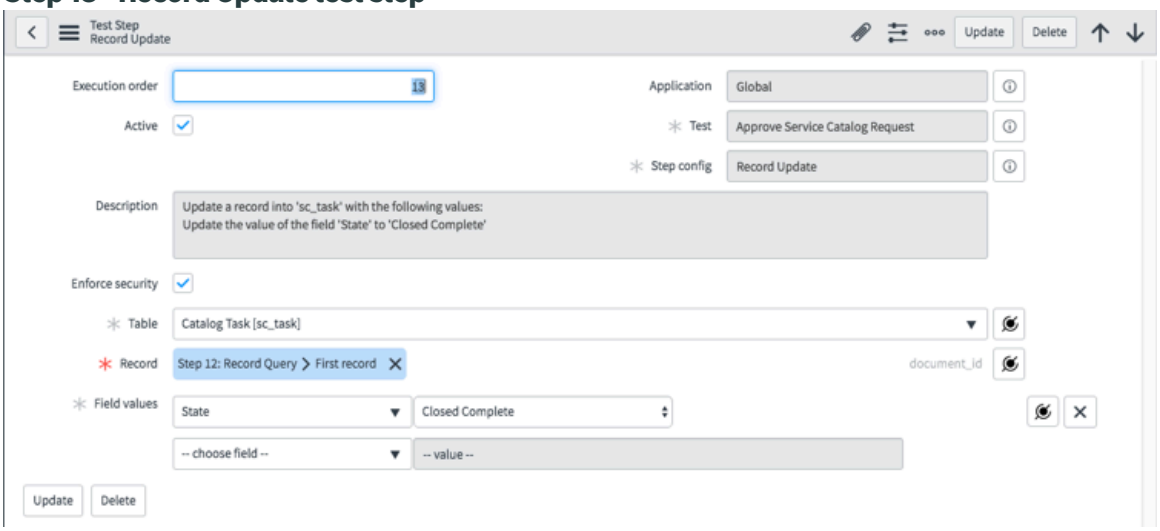


© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

13. Mark the [sc_task] record as *Closed Complete*.

Example

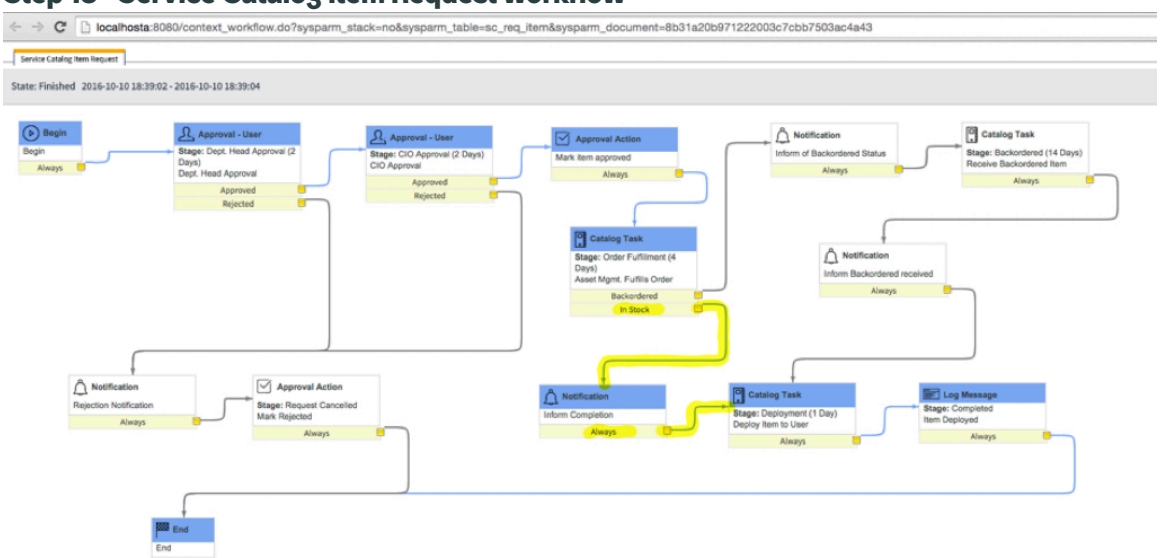
Step 13 - Record Update test step



Note how Step 13 uses the *First record* output variable from Step 12 to specify which record to mark as *Closed Complete*.

When the record is marked *Closed Complete*, the workflow exits the Catalog Task activity along the In Stock exit path.

Step 13 - Service Catalog Item Request workflow



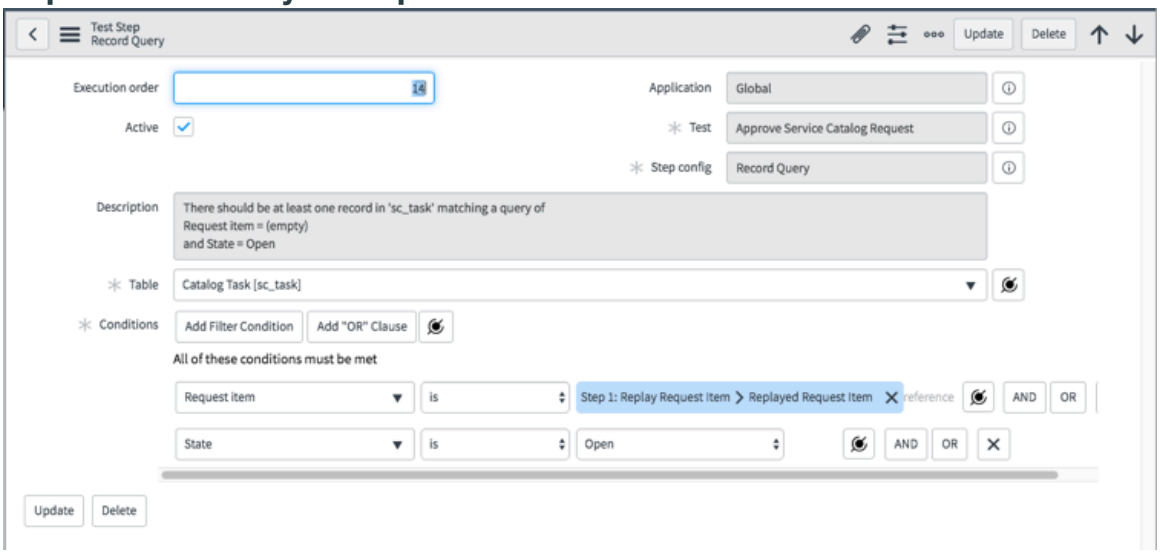
The workflow transitions to the Notification activity, then to the Catalog Task activity labelled Deploy Item to User.The Deploy Item to User Catalog Task activity inserts a new record into the [sc_task] table that instructs a user in the Deployment group to deliver the item.

14. Obtain the sys_id for the new catalog task with the Record Query step.
Note that Record Query returns an output variable with the sys_id of the first record returned from the query.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example

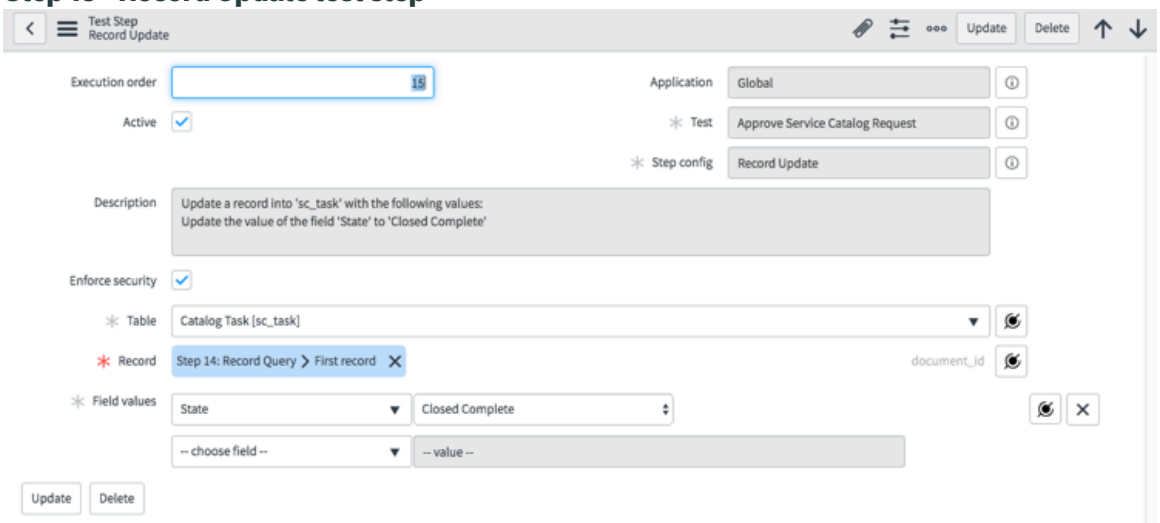
Step 14 - Record Query test step



15. Mark the [sc_task] record as Closed Complete.
    Note how Step 15 uses the *`First record`* output variable from Step 14 to specify which record to mark as Closed Complete.

Example

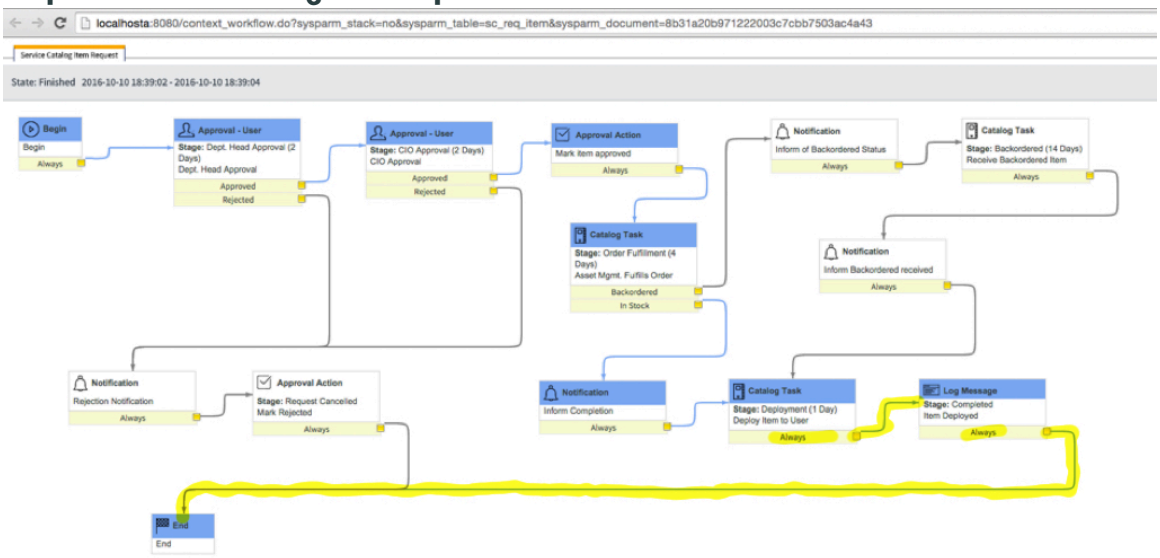
Step 15 - Record Update test step



When the record is marked Closed Complete, the workflow exits the Catalog Task activity, logs a message, and exits.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

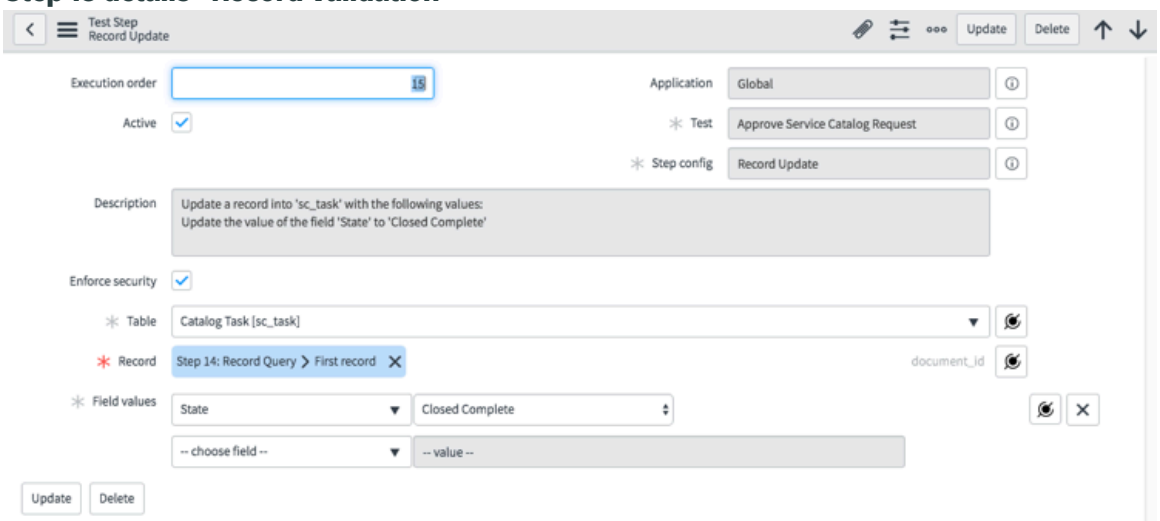
Step 15 - Service Catalog Item Request workflow



16. Verify that the request item in [sc_request_item] has the state Closed Complete.

Example

Step 15 details - Record Validation

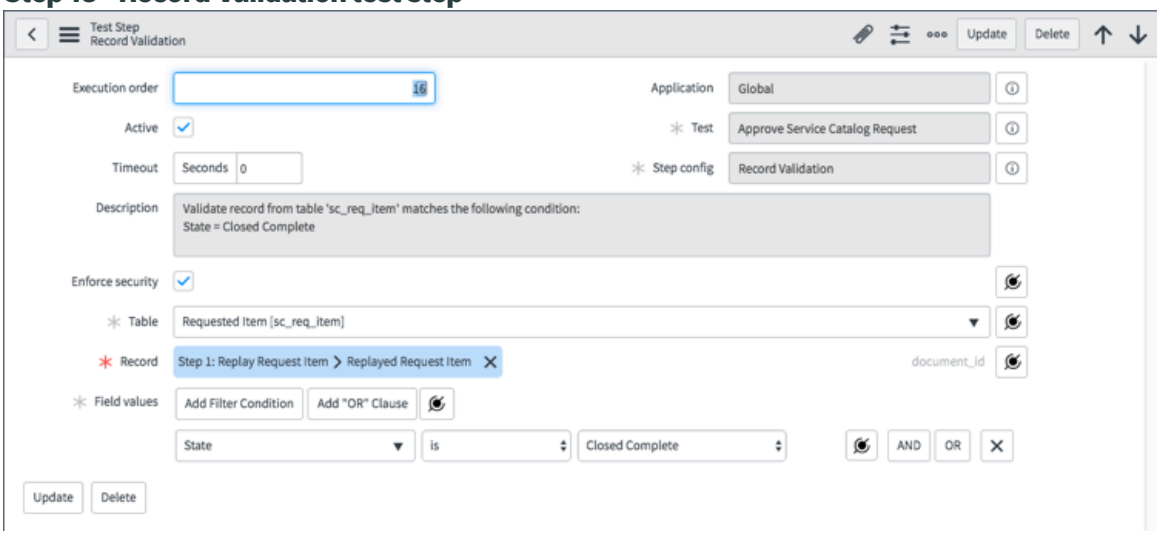


© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

17. Verify that the request [sc_request] has the state Closed Complete.

Example

Step 16 - Record Validation test step



Related topics

Replay Request Item

Automated Test Framework use case: retrieve an incident using REST-Inbound

The **Get Newly Created Resource via REST API Test** test is provided with the Automated Test Framework, and uses the REST - Inbound and assert steps.

Before you begin

Review the REST APIs and the Send REST Request- Inbound- REST API Explorer configuration step information before creating this test.

Role required: atf_ws_designer

About this task

This test creates an incident record, uses a REST- Inbound step to retrieve the record, and then uses assert steps to determine whether the request was successful.
You can use the Send REST Request- Inbound REST API Explorer step to build and test the request, or you can manually create the request using the Send REST Request- Inbound step. In either case, you must specify the basic authentication information on the Send REST Request- Inbound step form.

When creating your test, start with your REST API and determine what behavior you want to validate. You can then determine what test data to create. You can use other test step configuration categories to create, update, or delete records, and then use a Send REST Inbound step to retrieve, update, or delete the test data. Conversely, you can use the Send REST Inbound step to create records, and then use other test step configuration categories to validate that the records were created correctly.

This task steps you through creating the **Get Newly Created Resource via REST API Test** test.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Create a test.

   a. Navigate to **Automated Test Framework** > **Tests**, click **New**.
   The **Test** new record form is shown.

   b. Enter a test name and a description, and click **Submit**.

   c. Click the test created in the previous step.
   The **Test** form is shown.

2. Create test data.

   a. Click the **Add Test Step** button.
   The **Add Test Step** form is shown.

   b. Click a test configuration category, and then click a test configuration.
   To replicate the **Get Newly Created Resource via REST API Test** test, click **Server** > **Record Insert**, and then **Next**.
   The test configuration form you selected is shown.

   c. Fill in the information needed for the test configuration you selected.
   To replicate the **Get Newly Created Resource via REST API Test** test, on the **Record Insert** form, specify the incident table, and add a value for the **Short description** field, and click **Submit**.

3. Create the REST request.

   a. Click the **Add Test Step** button.
   The **Add Test Step** form is shown.

   b. Click a test configuration category, and then click a test configuration.
   To replicate the **Get Newly Created Resource via REST API Test** test, click **REST** > **Send REST Request - Inbound REST API Explorer**, and then **Next**.
   The **REST API Explorer** is shown.

   c. Fill in the information needed for the REST request, and click **Send**.
   To replicate the **Get Newly Created Resource via REST API Test** test, fill in the fields.

Name	Value
Namespace	now
API Name	Table API
API Version	latest
tableName	incident

   When you do not specify the **Basic authentication**, the REST API Explorer uses your credentials.

   The **Create Automated Test Step** is shown after you click **Send**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

d. When ready, click **Create Automated Test Step**.
The Send REST Request - Inbound test step is created.

e. Click the Send REST Request - Inbound step, and in the **Basic authentication** field, specify a basic auth configuration.
If no configurations are available, you can create a basic auth configuration by clicking **New** on the **Basic Auth Configurations** form.

f. To replicate the **Get Newly Created Resource via REST API Test** test, in the **Path** field, click the contextual search button, and then click **Record Insert** > **Record**.

g. Click **Update**.

4. Create assert steps to verify the REST response.

a. Click the **Add Test Step** button.
The **Add Test Step** form is shown.

b. Click a test configuration category, and then click a test configuration.
To replicate the **Get Newly Created Resource via REST API Test** test, click **REST** > **Assert Status Code**, and then **Next**.
The **Assert Status Code** form is shown.

c. Enter the information needed for the form, and click **Submit**.
To replicate the **Get Newly Created Resource via REST API Test** test, in the **Operation** field, select **is** and in the **Status Code** field, enter **200**.

To replicate the **Get Newly Created Resource via REST API Test** test, repeat this step to create **Assert Response JSON Payload is Valid**, and **Assert JSON Response Payload Element** test steps.

Testable Configurable Workspace components

Learn about the components and its associated actions in the configurable workspace.

For more information, see Testable configurable workspace components.

Accordion item

Actions	Data values and description	Data types
Check if accordion collapsed	Determines if the accordion is currently collapsed	Boolean
Toggle accordion content	Toggles the accordion item's content	
Get header text	Expected return value: Returns the header text value in the accordion's item trigger	String
Get caption text	Expected return value: Returns the caption text value in the accordion's item trigger	String

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Action bar

Actions	Data values and description	Data types
Click action	Clicks an action	N/A
Has action	Checks if an action exists	Boolean

Activity Stream

Actions	Data values and description	Data types
Select Option In Filter Set Modal Field	Selects the given option in the given field in the filter set modal	N/A
Apply Filter Set With Name	Applies the filter set with the given name	N/A
Get Journal Tile Content	Returns the content of a journal tile at the given index **Note:** It can contain either HTML or text.	String
Open Filter Set Edit Modal	Opens the edit filter set modal with the given name	N/A
Open Filter Set Modal	Opens the filter set modal	N/A
Set Filter Set Name	Inputs the given value as the new filter set's name	N/A
Create New Filter Set From Existing Selections	Creates a new filter set from the current filter selections	N/A
Select All Options In Filter Set Modal Field	Selects all options in the given field in the filter set modal	N/A
Clear Option In Filter Set Modal Field	Clears the item at the given index in the given field in the filter set modal	String
Get Journal Tile Field Name	Returns the type of a Journal Tile (work_notes or comments) at the given index	String
Get Activity Tile Type	Returns the type of an activity tile displayed in the activity stream	N/A
Click Save Filter Set Button	Clicks the save button in the filter set modal	N/A
Get Filter Set Name	Inputs the given value as the new filter set's name	String
Open Filters	Opens the filter section	N/A
Can Find Filter Set With Name	Expected return	Boolean

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Actions	Data values and description	Data types
	• True: Returns True if a filter set with the given name exists • False: Returns False if a filter set with the given name doesn't exists	
Clear All Options In Filter Set Modal Field	Clears all options in the given field in the filter set modal	N/A

Activity Stream Compose

Actions	Data values and description	Data types
Select Field	Selects a compose journal field tab or dropdown menu	Boolean
Is Journal Popped Out	Tests if the journal field composition for the given field name is shown as being edited outside of compose	N/A
Is Stacked View Enabled	Tests if Stacked View is enabled	Boolean
Is Stacked View Available	Tests if Stacked View is available	Boolean
Is Email Popped Out	Tests if the email composition is shown as if being edited outside of compose	N/A
Check If Tab Selected	Tests if a compose tab is selected	Boolean
Set Journal Text	Enters text into the editor of the provided field	N/A
Click pop-in button for journal editor	Clicks on the pop in button for journal editor	N/A
Select Tab	Selects a compose journal field tab	Boolean
Click pop-out button for journal editor	Clicks on the pop out button for journal editor	N/A
Click Post Button	Clicks on the editor post button	N/A
Click pop-in button for email editor	Clicks on the pop in button for email editor	N/A
Get Journal Text	Reads the text in the editor of the provided field	String

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Actions	Data values and description	Data types
Is Using Tabs	Tests if the component is using now-tabs to display the fields	N/A
Toggle Stacked View	Clicks the Stacked View Toggle	N/A
Is Field Selected	Tests if a compose field is selected	N/A
Click pop-out button for email editor	Clicks on the pop out button for email editor	N/A

Alert

Actions	Data values and description	Data types
Check if alert collapsed	Checks if the alert is currently collapsed	Boolean
Toggle alert collapse state	Toggles the collapse state of the alert if its collapsible	N/A
Check if alert contains action	Determines if the alert has an action button	Boolean
Check if alert is collapsible	Checks if the alert can be collapsed	Boolean
Click alert button	Clicks the action button for the alert	N/A
Get alert title	Retrieves the title of the alert	String
Check if the alert text is focusable	Checks if the alert text can be focused	Boolean

Avatar

Actions	Data values and description	Data types
Click the avatar	Clicks the avatar. **Note:** The avatar gets clicked only if the interaction mode is being set to a button or a link.	N/A
Get the initials of the avatar	Returns the text value of the avatar if it has initials.	N/A
Check if the avatar has an image	Returns true if the avatar displays an image.	Boolean
Check if the avatar has initials	Returns true if the avatar returns the user initials.	Boolean

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Actions	Data values and description	Data types
Check if the avatar is anonymous	Checks if the avatar doesn't have an image or the user initials.	Boolean

Badge

Actions	Data values and description	Data types
Get the badge text	Returns the text content of the badge	String

Breadcrumb

Actions	Data values and description	Data types
Click item	Clicks the matching breadcrumb item	N/A

Button

Actions	Data values and description	Data types
Click	Clicks the button	N/A
Get Text	Expected return value: Retrieves the text of the button	String

Button bare

Actions	Data values and description	Data types
Click	Clicks the button	N/A
Get Text	Expected return value: Retrieves the text of the button	String

Button circular

Actions	Data values and description	Data types
Click	Clicks the button	N/A
Get Text	Expected return value: Retrieves the text of the button	String

Button iconic

Actions	Data values and description	Data types
Click	Clicks the button	N/A

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Actions	Data values and description	Data types
Get Text	Expected return value: Retrieves the text of the button	String

Button stateful

Actions	Data values and description	Data types
Click	Clicks the button	N/A
Get Text	Expected return value: Retrieves the text of the button	String

Card Base Container

Actions	Data values and description	Data types
Click	Clicks the card **Note:** This is applicable only for interactive cards.	N/A
is Selected	Expected return • True: Returns True if the card is currently marked as selected • False: Returns False if the card is currently not marked as selected	Boolean

Card Base Header

Actions	Data values and description	Data types
Get Caption Text	Expected return value: Retrieves the text of the card caption	String
Get Heading Level	Expected return value: Retrieves the heading level	String
Get Heading Text	Expected return value: Retrieves the text of the card header	String
Get Tagline Text	Expected return value: Retrieves the text of the card tagline	String

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Checkbox

Actions	Data values and description	Data types
Click checkbox	Clicks the checkbox	N/A

Collapse

Actions	Data values and description	Data types
Check if collapse is collapsed	Expected return • True: Returns True if the collapse is currently not expanded • False: Returns False if the collapse is currently expanded	N/A

Color selector

Actions	Data values and description	Data types
Get trigger label	Retrieves the trigger label	String
Open color selector panel	Opens the color selector panel popover	N/A
Check if color selector panel is displayed	Expected return • True: Returns True if the color selector panel is displayed • False: Returns False if the color selector panel is not displayed	Boolean

Custom popover

Actions	Data values and description	Data types
Click popover trigger	Simulates a click on the slotted trigger element for the popover	N/A
Popover panel visibility	Checks if the popover is opened and its panel is visible	Boolean

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Date-Time

Actions	Data values and description	Data types
Check if item is disabled	Expected return • True: Returns True if the matching item in the calendar is disabled • False: Returns False if the matching item in the calendar is enabled	Boolean
Set input value	Sets value on native input, includes dispatch of event	N/A
Click Action	Clicks the first matching footer action	N/A
Check if calendar item is focused	Expected return • True: Returns True if the matching item in the calendar is focused • False: Returns False if the matching item in the calendar is not focused	Boolean
Check if calendar panel is displayed	Expected return • True: Returns True if the calendar panel is displayed • False: Returns False if the calendar panel is not displayed	Boolean
Select date item	Selects the matching date item in the calendar	N/A
Click on the input	Clicks on the input of now-date-time and opens the calendar	N/A
Check if item is selected	Expected return • True: Returns True if the matching item in the calendar is selected • False: Returns False if the matching item in the calendar is not selected	Boolean

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Date-Time-Interval

Actions	Data values and description	Data types
Set the end input value	Expected return • True: Returns True if the matching item in the calendar is disabled • False: Returns False if the matching item in the calendar is enabled	N/A
Check if calendar panel is displayed	Sets value on native input, includes dispatch of event	Boolean
Check if item is selected	Clicks the first matching footer action	Boolean
Click the start input	Expected return • True: Returns True if the matching item in the calendar is focused • False: Returns False if the matching item in the calendar is not focused	N/A
Check if calendar item is focused	Expected return • True: Returns True if the calendar panel is displayed • False: Returns False if the calendar panel is not displayed	Boolean
Check if preset is selected	Selects the matching date item in the calendar	Boolean
Check if item is disabled	Clicks on the input of now-date-time and opens the calendar	Boolean
Click Action	Expected return • True: Returns True if the matching item in the calendar is selected • False: Returns False if the matching item in the calendar is not selected	N/A
Click the end input	Clicks on the start input of now-date-time and opens the calendar	N/A

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Actions	Data values and description	Data types
Select a preset	Selects the matching preset item from the preset list	N/A
Select a date item	Selects the matching date item in the calendar	N/A
Set the start input value	Sets start value on native input, include dispatch of events	N/A

Dropdown

Actions	Data values and description	Data types
Click trigger	Click the dropdown	N/A
Number of selected items	Returns the number of selected items in the dropdown menu	Numeric
Get text	Retrieves the text content of the dropdown menu	String
Check selection	Returns true of the matching item of the menu is selected **Note:** You also need to select a timeframe for the matching item	Boolean
Check panel display	Returns true of the menu is selected	Boolean
Select first matching item	Selects the first matching item in the menu **Note:** You also need to select a timeframe for the matching item	N/A

Flyout Menu

Actions	Data values and description	Data types
Check if the flyout menu panel is displayed	Returns true if the flyout menu is opened and the list panel is visible	Boolean
Check selection	Returns true if the matching item of the panel is selected	Boolean
Click flyout menu trigger	Clicks the slotted trigger element for the flyout menu	N/A
Select first matching item	Selects the first matching item found in the panel	N/A

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Actions	Data values and description	Data types
Number of selected items	Returns the number of selected items in the flyout menu panel	Numeric

Heading

Actions	Data values and description	Data types
Get Text	Expected return value: Retrieves the text of the heading	String

Highlighted value

Actions	Data values and description	Data types
Get the highlighted-value text	Expected return value: Retrieves the text of the highlighted value	String

Input

Actions	Data values and description	Data types
Set value	Sets value on native input, includes dispatch of event	String

Input password

Actions	Data values and description	Data types
Set value	Sets value on native input, includes dispatch of event	String

Input phone

Actions	Data values and description	Data types
Set value	Sets value on native input, includes dispatch of event	String

Input URL

Actions	Data values and description	Data types
Set value	Sets value on native input, includes dispatch of event	String

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

List

Actions	Data values and description	Data types
Apply filter on the list	Applies filter on the list using encoded query	Boolean
Open a record in the list	Opens a record in the list by clicking on first cell link of record row	Boolean
Validate record visibility in the list	Validates if the record with the given sysid is present in the list	Boolean
Click a list UI action button	Clicks the UI action button with the given sysid	Boolean
Validate the visibility of list's declarative actions	Validates the visibility of declarative action button For example, the **New** button. It is validated based on its sysid.	Boolean

Loader

Actions	Data values and description	Data types
Click the loader action	Clicks the action associated with the loader	N/A
Get the loader text	Returns the text of the loader	String
Check if loader has action	Expected return • True: Returns True if the loader has an action defined • False: Returns False if the loader doesn't have an action defined	Boolean

Loader Custom

Actions	Data values and description	Data types
Get the loader custom text	Returns the text of the loader custom	String

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Modeless Dialog

Actions	Data values and description	Data types
Click window action	Clicks the matching window action. **Note:** The matcher could correspond to the index of the button within the buttons collection.	N/A
Check if dialog is closed	Expected return • True: Returns True if the dialog is opened • False: Returns False if the dialog is not opened	Boolean
Get the now-modeless-dialog text	Returns the text of the now-modeless dialog	String

now-alert-list

Actions	Data values and description	Data types
Toggle alert list collapsed state	Checks if the alert is currently collapsed	N/A
Check if alert list can collapse	Toggles the collapse state of the alert if its collapsible	Boolean
Dismiss alerts	Determines if the alert has an action button	N/A
Check if alert list collapsed	Checks if the alert can be collapsed	Boolean

now-card-footer

Actions	Data values and description	Data types
Get Footer End Text	Returns the text of the card footer's right side **Note:** This is the end part of the footer text	String
Get Footer Text	Returns the text of the card footer	String
Get Footer Start Text	Returns the text of the card footer's left side	String

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Actions	Data values and description	Data types
	Note: This is the start part of the footer text	

now-collapse-trigger

Actions	Data values and description	Data types
Finish collapse transition	Allows now-collapse to finish transitioning	N/A

now-color-selector-panel

Actions	Data values and description	Data types
Get color name	Returns the name of the color	String
Select color swatch by id	Selects the color swatch from color groups by the color id	N/A
Set hex value	Sets the hex value. **Note:** The valid value is either 3 or 6 hexadecimal characters without #.	N/A
Set color name	Sets the name of the color in the color name edit input field	N/A
Set slider value	Sets the value of hue slider	N/A
Click action	Clicks the first matching footer action	N/A
Set rgb value	Sets the rgb value	N/A
Set hsl value	Sets the hsl value	N/A

now-date-time-calendar

Actions	Data values and description	Data types
Check if item is disabled	Expected return • True: Returns True if the matching item in the calendar is disabled • False: Returns False if the matching item in the calendar is enabled	Boolean
Check if item is focused	Expected return	Boolean

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Actions	Data values and description	Data types
	• True: Returns True if the matching item in the calendar is focused • False: Returns False if the matching item in the calendar is not focused	
Get active month	Returns a numeric value of the currently selected month	Numeric
Check if item is selected	Expected return • True: Returns True if the matching item in the calendar is selected • False: Returns False if the matching item in the calendar is not selected	Boolean
Get active year	Returns a numeric value of the currently selected year	Numeric
Select date item	Selects the matching date item in the calendar	N/A

now-dropdown-custom-target

Actions	Data values and description	Data types
Check panel display	Expected return • True: Returns True if the panel is displayed • False: Returns False if the if the panel is not displayed	Boolean
Select first matching item	Selects the first matching item found in the panel	N/A
Check selection	Expected return • True: Returns True if the matching item in the panel is selected • False: Returns False if the matching item in the panel is not selected	Boolean
Number of selected items	Returns the number of selected items in the dropdown panel	Numeric

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

now-dropdown-list

Actions	Data values and description	Data types
Check if item is selected	Checks if the item is selected in the panel	Boolean
Get selected items count	Retrieves the number of items currently selected within the dropdown panel	Numeric
Check panel display	Expected return • True: Returns True if the panel is displayed • False: Returns False if the panel is not displayed	Boolean
Get panel item count	Returns the number of items in the dropdown panel	Numeric
Select first matching item	Selects the first matching item in the panel	N/A

now-flyout-menu-list

Actions	Data values and description	Data types
Get number of selected items	Returns the number of selected items in the flyout-menu panel	Numeric
Get number of items	Returns the number of items in the flyout-menu panel	Numeric
Check if the flyout menu panel is displayed	Expected return • True: Returns True if the panel is displayed • False: Returns False if the panel is not displayed	Boolean
Select item	Selects the first matching item in the panel	N/A
Check if item is selected	Expected return • True: Returns True if the matching item in the panel is selected • False: Returns False if the matching item in the panel is not selected	Boolean

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

now-input-date-time

Actions	Data values and description	Data types
Set input value	Sets value on native input, includes dispatch of event	N/A

now-modal

Actions	Data values and description	Data types
Is Displayed	Expected return • True: Returns True if the modal is currently displayed • False: Returns False if the modal is currently not displayed	Boolean
Get Content	Returns the text of the modal	N/A
Has Footer	Expected return • True: Returns True if the modal contains a footer • False: Returns False if the modal doesn't contain a footer	Boolean
Dismiss	Clicks the dismiss action of the modal	N/A
Get Heading	Returns the heading value of the modal	String
Click Action	Clicks the first matching footer action	N/A

now-record-list-connected

Actions	Data values and description	Data types
Apply filter on the list	Applies the filter in the form of encodedQuery	N/A
Click a list UI action button	Selects a UI action button in the list	N/A
Validate record visibility in the list	Validates if the selected record is visible in the list	Boolean
Validate the visibility of list's declarative actions	Validates if the list's declarative action is visible or not	Boolean

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Actions	Data values and description	Data types
Open a record in the list	Opens the selected record from the list	N/A

now-tabs

Actions	Data values and description	Data types
Select tab by matcher	Selects a tab based on the provided criteria **Note:** It also allows substring matching	N/A
Check if matching tab is selected	Checks if the tab matching the provided criteria is currently selected	Boolean

now-tabs-vertical

Actions	Data values and description	Data types
Check if matching tab is selected	Checks if the tab matching the provided criteria is currently selected	N/A
Select tab by matcher	Selects a tab based on the provided criteria **Note:** It also allows substring matching	Boolean

now-uxf-tab-set

Actions	Data values and description	Data types
Assert tab selected by label	Returns true if the tab with the given label is present. An error is shown if the tab is not found. **Note:** You must enter the exact name in the label to check if it exists.	Boolean
Select tab by order	Selects a tab by its order in the tab set. The order starts from 1.	N/A

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Actions	Data values and description	Data types
	Note: You must enter the required order of the tab you want to be selected.	
Select tab by label	Selects a tab by its label. **Note:** You must enter the required label of the tab you want to be selected.	N/A

pagination-control

Actions	Data values and description	Data types
Get page label	Returns page label	String
Check if page selected	Checks if the page matching the provided criteria is currently selected	Boolean
Select page size	Selects a page size from the dropdown control	N/A
Click go-to-next-page button	Simulates a click on the go-to-next-page button	N/A
Set input value	Sets value on input field and dispatches associated events	N/A
Click go-to-last-page button	Simulates a click on the go-to-last-page button	N/A
Click go-to-previous-page button	Simulates a click on the go-to-previous-page button	N/A
Click go-to-first-page button	Simulates a click on the go-to-first-page button	N/A

pill

Actions	Data values and description	Data types
Get the pill text	Returns the text content of the pill	String
Dismiss pill	Clicks the dismiss button of the pill	N/A
Check if pill is dismissible	Expected return	Boolean

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Actions	Data values and description	Data types
	• True: Returns True if the pill is dismissible • False: Returns False if the pill is not dismissible	
Click pill	Clicks the pill	N/A

Radio buttons

Actions	Data values and description	Data types
Select a radio button	Clicks the matching radio button, toggling its selected state	N/A

Select

Actions	Data values and description	Data types
Click trigger element	Simulates a click on the trigger element	N/A
Check if dropdown displayed	Determines if the dropdown list is currently visible on the screen	Boolean
Select matching item	Selects the matching item in the dropdown list	N/A

sn-canvas-tabs

Actions	Data values and description	Data types
Assert tab with label exist	Returns true if the tab with the required label exist	Boolean
Assert tab with label is selected	Returns true if the tab with the required label is selected	Boolean
Close tab by order	Closes the tab by its order in the tab set. The order starts from 1. **Note:** You must enter the required order of the tab you want to be closed.	N/A
Close tab with label	Closes the tab by its label in the tab set.	N/A

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Actions	Data values and description	Data types
	Note: You must enter the required label of the tab you want to be closed.	
Select tab by order	Selects a tab by its order in the tab set. The order starts from 1. **Note:** You must enter the required order of the tab you want to be selected.	N/A
Select tab with label	Selects a tab by its label. **Note:** You must enter the required label of the tab you want to be selected.	N/A

sn-canvas-toolbar

Actions	Data values and description	Data types
Assert button exists by group and tool tip	Verifies that a button with the given tooltip is present within the group. • group: top or bottom • toolTip: • Expected return ○ True: Returns True if a button with the given tooltip is present within the group ○ False: Returns False if a button with the given tooltip is not present within the group	Boolean
Assert button selected by group and tool tip	Selects a button with the given tooltip within the group.	Boolean
Select button by group and order	Selects a button within the toolbar group by its ordering. The order starts from 1. Groups can either be top or bottom.	N/A

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Actions	Data values and description	Data types
Select button by group and tool tip	Selects a button within the toolbar group by its toolTip values. The order starts from 1. ToolTip value	N/A

Split button

Actions	Data values and description	Data types
Is Panel Displayed	Expected return • True: Returns True if the panel is displayed • False: Returns False if the panel is not displayed	Boolean
Is Item Selected	Returns true if the panel's matching item is selectedExpected return • True: Returns True if the matching item in the panel is selected • False: Returns False if the matching item in the panel is not selected	Boolean
Click	Clicks the primary action of the split-button	N/A
Get button text	Returns the text content of the split-button	String
Click Trigger	Clicks the split-button's dropdown trigger	N/A
Select Item	Selects the first matching item found in the panel	N/A
Count Selected Items	Returns the number of selected items in the dropdown panel of the split-button	Numeric

Stepper

Actions	Data values and description	Data types
Is Step Selected	Expected return	Boolean

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Actions	Data values and description	Data types
	• True: Returns True if the step is currently selected • False: Returns False if the step is currently not selected	
Click next page button	Clicks the next page button	N/A
Click previous page button	Clicks the previous page button	N/A
Select a step	Selects a step	N/A

Tabs

Actions	Data values and description	Data types
Select tab by order	Selects a tab by its order in the tab set **Note:** The order starts at 1.	Boolean
Select tab by label	Selects a tab by its label **Note:** An error is shown if the tab is not found.	N/A
Assert tab selected by label	Asserts that a tab with the given label is selected. **Note:** An error is shown if the tab is not found.	N/A

Test link

Actions	Data values and description	Data types
Get text	Returns the text of the text-link	String
Click	Clicks the link	N/A

Text area

Actions	Data values and description	Data types
Set textarea value	Sets value on native textarea, includes dispatch of events	N/A

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Toggle

Actions	Data values and description	Data types
Click trigger element	Simulates a click on the trigger element	N/A

Typeahead

Actions	Data values and description	Data types
Set input value	Sets value on typeahead input and dispatches associated events	N/A
Select item	Selects one of the items of the typeahead	N/A
Click the input	Clicks on the input of the typeahead	N/A
Check if panel is displayed	Expected return • True: Returns True if the panel is opened and visible • False: Returns False if the panel is not opened and visible	Boolean

Typeahead multi

Actions	Data values and description	Data types
Select item	Selects one of the items of the typeahead	N/A
Click the input	Clicks on the input of the typeahead	N/A
Check if panel is displayed	Expected return • True: Returns True if the panel is opened and visible • False: Returns False if the panel is not opened and visible	Boolean
Set input value	Sets value on typeahead input and dispatches associated events	N/A

Configurable workspace components examples

To grasp how to interact with configurable workspace components, review these examples.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: When you submit an action in a component, a test step is created.

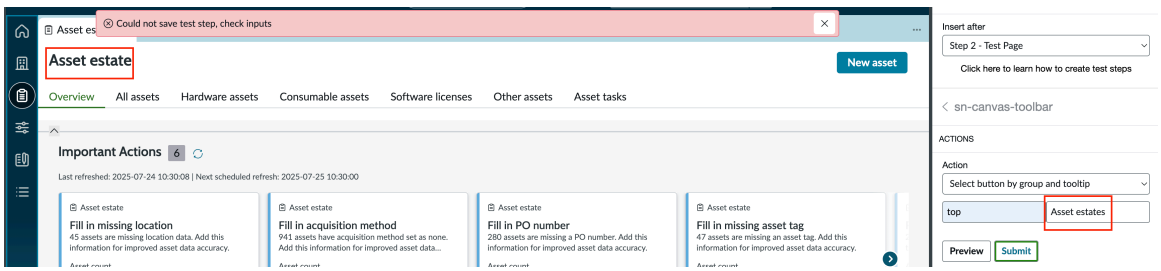
Example: sn-canvas-toolbar

For sn-canvas-toolbar component, it has the following four actions:

- Assert button exists by group and tool tip
- Assert button selected by group and tool tip
- Select button by group and order
- Select button by group and tool tip

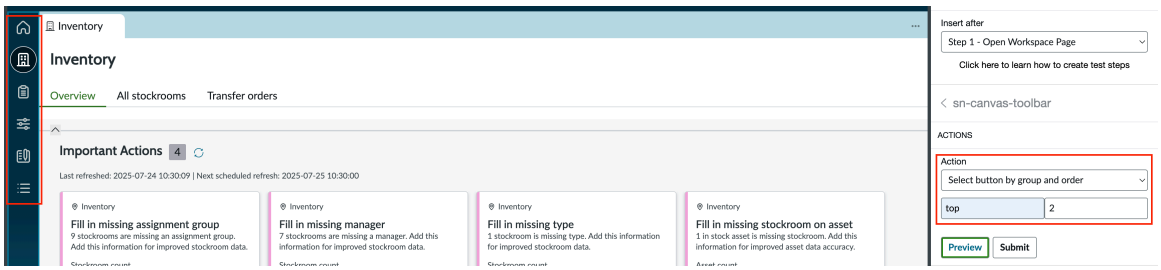
These are the expected values for the actions:

- group: Enter either 'top' or 'bottom' depending on which tab in the toolbar you want to interact with.
- toolTip: Name of the tab in the toolbar you want to open.



Note: An error is shown if the toolTip name doesn't exactly matches the tab name you want to open up.

- order: Depending on the group entry, the order number is decided. The order starts at 1. For instance, if you've selected 'top' for a group in a toolbar with six items and want to access the second tab from the top, you'd enter '2' as the order number. Similarly you'd enter '5' as the order number if you've selected 'bottom' for a group.



Note: When you submit an action to select a specific tab (based on its group and order), that tab becomes focused or highlighted. When you enter the exact name of the tab in the toolTip, that tab opens up.

Example: now-uxf-tab-set

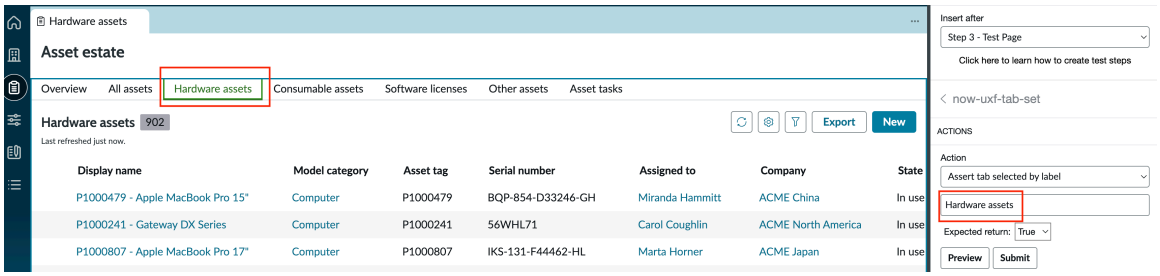
For now-uxf-tab-set component, it has the following three actions:

- Assert tab selected by label
- Select tab by order
- Select tab by label

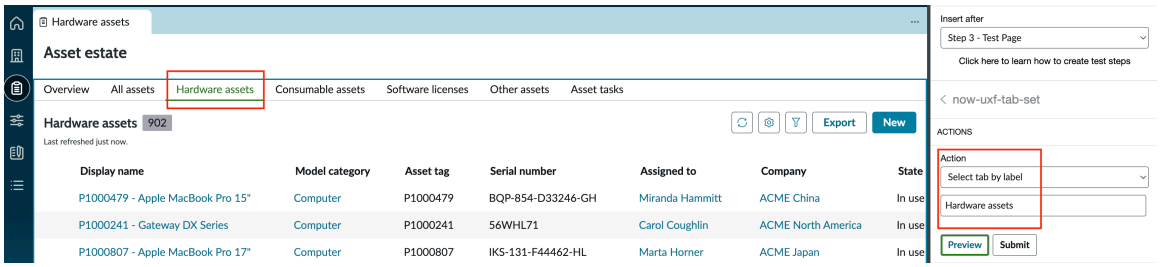
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

These are the expected values for the actions:

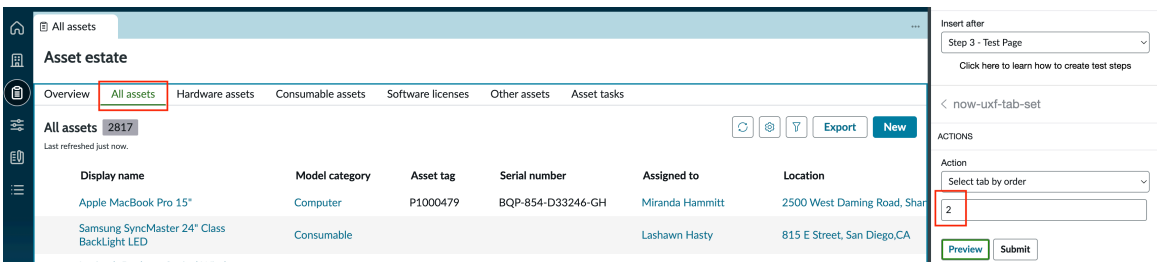
- label: Enter the exact name of the tab that you want to assert if it exists. For example, after selecting the Assert tab selected by label action, enter 'Hardware assets' to check if the 'Hardware assets' tab is present. The expected return value is True if it exists. If you select Select tab by label action, enter the name of the tab in the toolbar you want to open.
  - Assert tab selected by label



  - Select tab by label



- order: The order number you enter determines which tab opens on the selected toolbar.



Example: now-record-list-connected

For now-record-list-connected component, it has the following five actions:

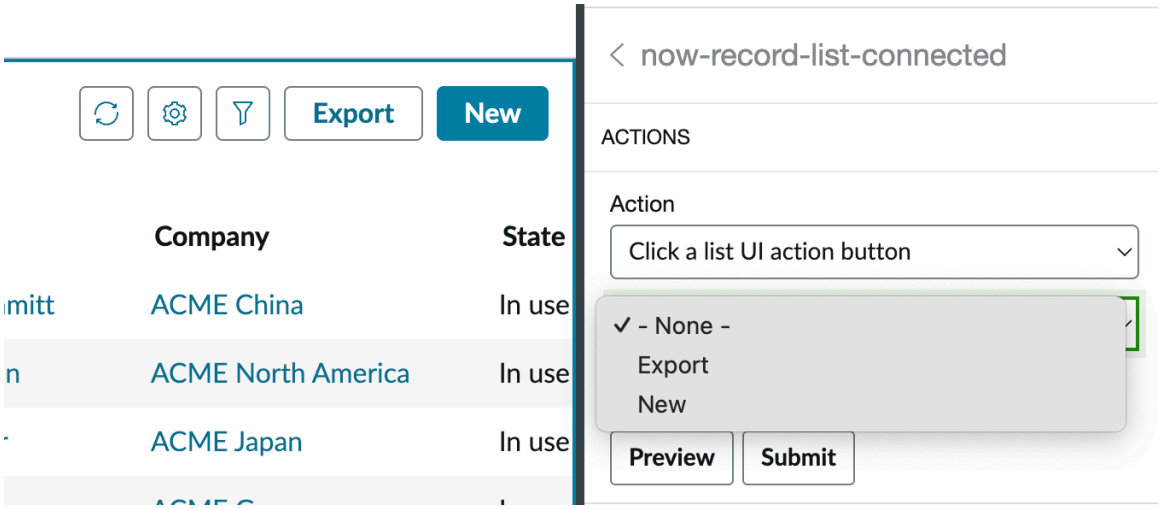
- Apply filter on the list
- Click a list UI action button
- Validate record visibility in the list
- Validate the visibility of list's declarative actions
- Open a record in the list

These are the expected values for the actions:

- Select the required UI action.

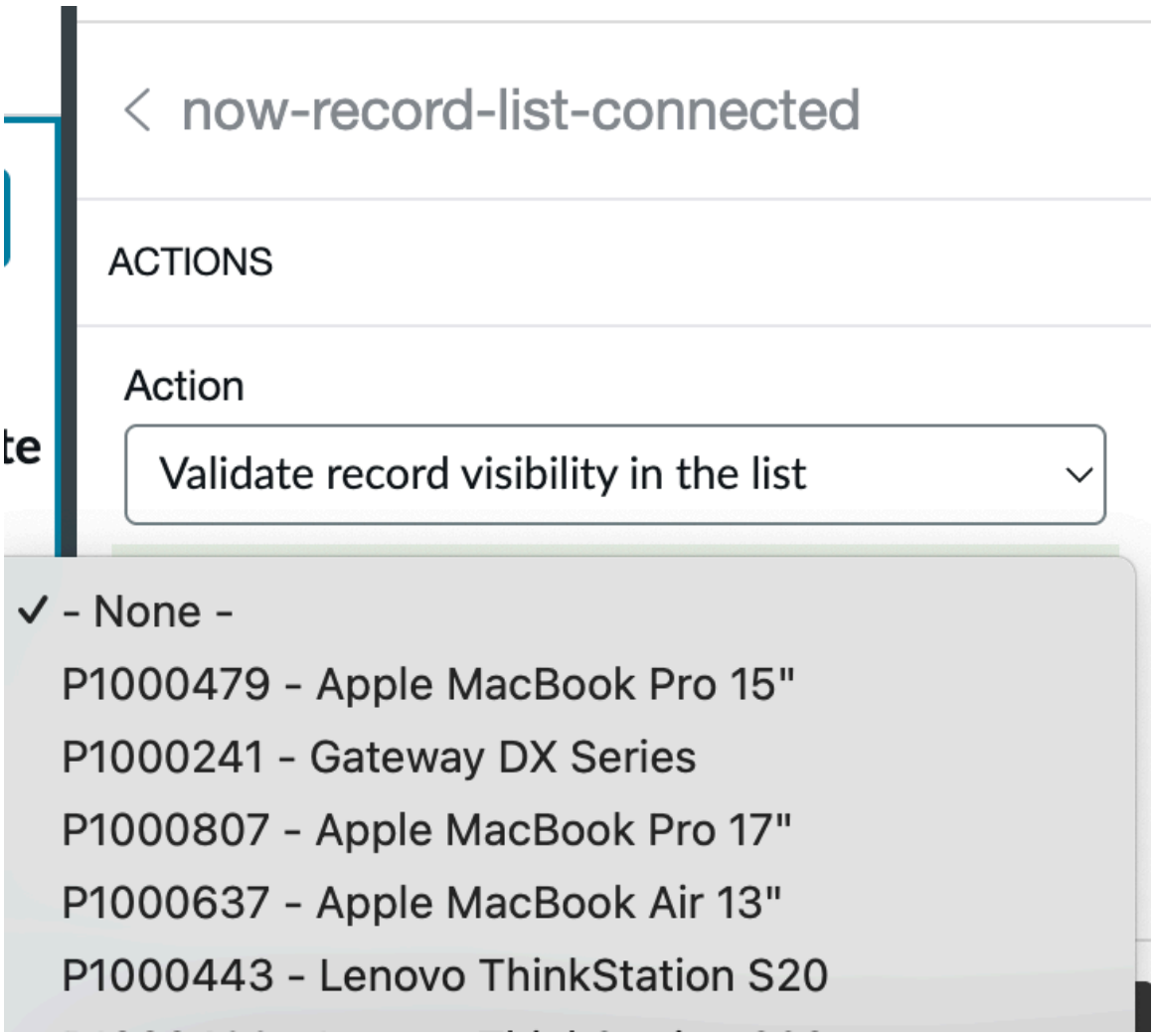
> **Note:** Only the UI actions which are available in the list shows up.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



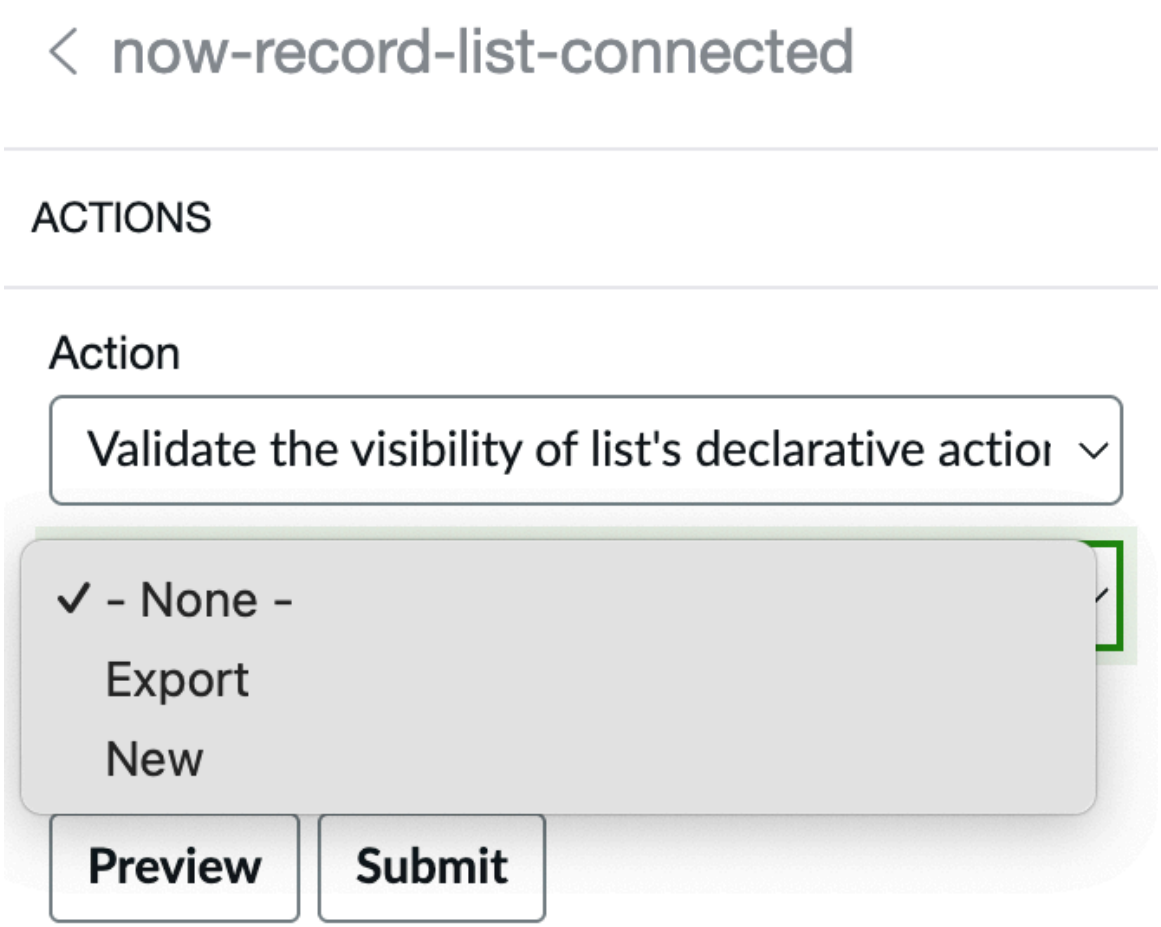
- Select the record from the list you want to validate.

Note: All the records in the list show up when you select the Validate record visibility in the list action.



- Select the required UI action to validate its visibility in the list.

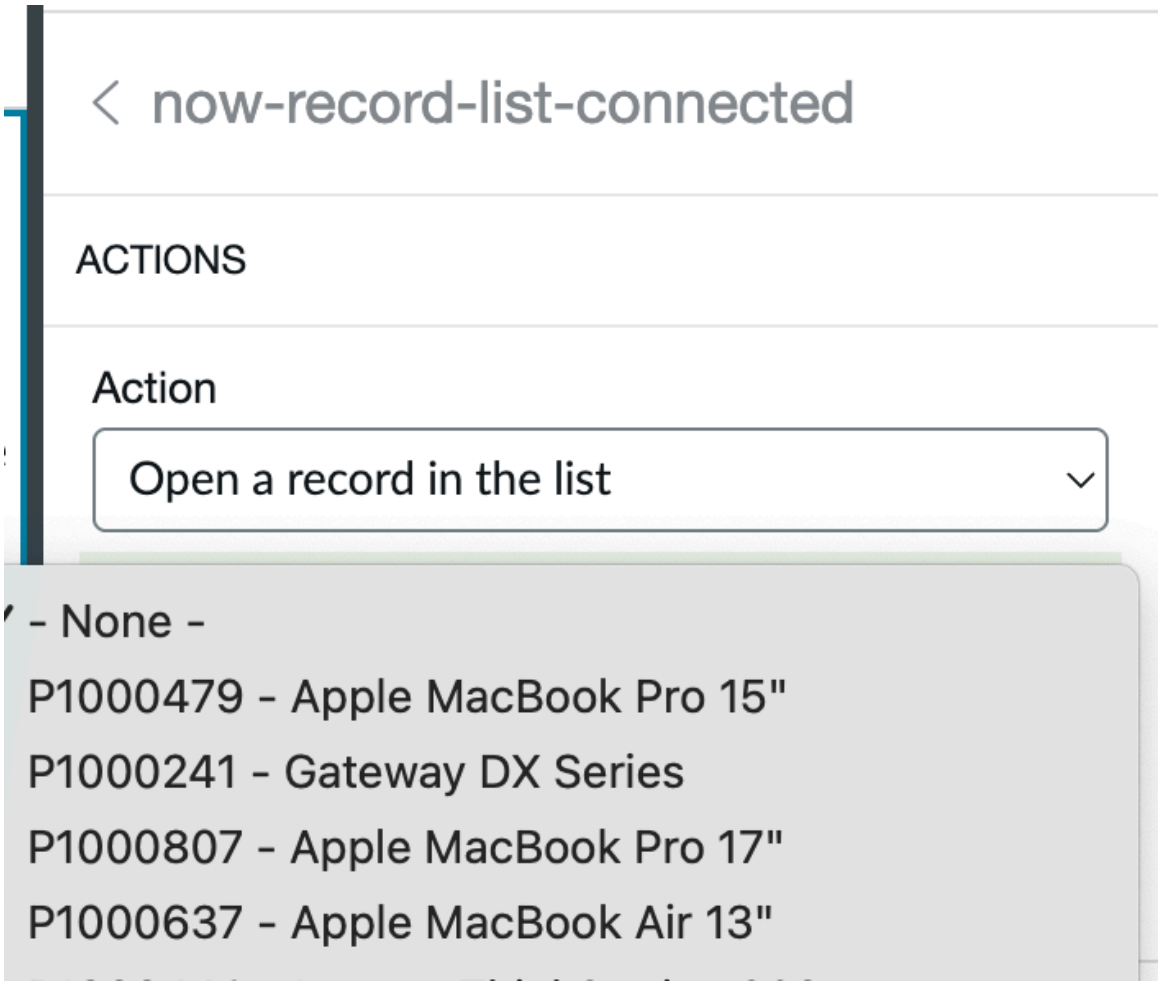
© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



- Select the required record to open from the list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: All the records in the list show up when you select the Open a record in the list action.



Automated Test Framework roles

Automated Test Framework is installed with these roles.

ATF test administrator [atf_test_admin]

To learn more about managing per-user subscriptions, see Managing per-user subscriptions in Subscription Management and contact your account representative.

Create or edit Automated Test Framework properties. Has permission to:

- View the tests page
- Create/edit/delete tests
- Create/edit/delete test steps
- View the step config page
- View the test runner page
- View the test suite results, test results and result items pages
- Execute user tests
- View, create, edit, delete and execute test suites

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Create/edit step config records
- Create/edit Automated Test Framework properties

Contains Roles

List of roles contained within the role.

- impersonator
- personalize_dictionary

Groups

List of groups this role is assigned to by default.

None

Elevated

Whether the role is an elevated role. Elevated roles aren't assigned to users or groups, and must be used by elevation. For details, see Elevate to a privileged role.

No.

Special considerations

None.

ATF test designer [atf_test_designer]

To learn more about managing per-user subscriptions, see Managing per-user subscriptions in Subscription Management and contact your account representative.

View Automated Test Framework properties only (can't create or edit properties). Has permission to:

- View the tests page
- Create/edit/delete tests
- Create/edit/delete test steps
- View the step config page
- View the test runner page
- View the test suite results, test results and result items pages
- Execute user tests
- View, create, edit, delete and execute test suites
- View Automated Test Framework properties

Contains Roles

List of roles contained within the role.

impersonator

Groups

List of groups this role is assigned to by default.

None

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Subscription

Whether the role is a chargeable user role that requires allocation of users with this role to subscriptions.

Elevated

Whether the role is an elevated role. Elevated roles aren't assigned to users or groups, and must be used by elevation. For details, see Elevate to a privileged role.

No.

Special considerations

None.

ATF workspace designer [atf_ws_designer]

To learn more about managing per-user subscriptions, see Managing per-user subscriptions in Subscription Management and contact your account representative.

View or set the basic or mutual authentication needed for REST endpoints that require authentication.

Contains Roles

List of roles contained within the role.

- atf_test_designer
- web_service_admin

Groups

List of groups this role is assigned to by default.

None

Subscription

Whether the role is a chargeable user role that requires allocation of users with this role to subscriptions.

No.

Elevated

Whether the role is an elevated role. Elevated roles aren't assigned to users or groups, and must be used by elevation. For details, see Elevate to a privileged role.

No.

Special considerations

None.

Administration

The Administration module contains forms for configuring and managing the automated test framework.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Properties

On the Properties form, you can set parameters that control how the system executes automated tests and test suites.

Test/Test Suite Properties

Field / Element	Description
Enable test/test suite execution *sn_atf.runner.enabled*	If checked, enables running tests and test suites on this instance. This setting is unchecked by default to prevent users from unintentionally running tests on production instances.
Enable scheduled test suite execution sn_atf.schedule.enabled	If checked, enables scheduling test suites on this instance.

Test Debugging Properties

Field / Element	Description
Enable additional debugging functionality *sn_atf.debug*	Enables additional debugging functionality, including adding a debugging tab on the client Test Runner UI page and saving UI Test Result JSON to the test result record.

Screenshot Properties

Field / Element	Description
Enable or disable screenshot capture during test execution. *sn_atf.screenshots.mode* **Note:** For additional information, see Set the system property to control when the Automated Test Framework captures screenshots	• To capture screenshots for all steps, select **Enable for all steps**. • To capture screenshots only for failed steps, select **Enable for all failed steps**. • To capture no screenshots, select **Disable for all steps**.
Enable the GlideScreenshot feature *sn_atf.screenshots.use_glide_screenshot*	Improves the fidelity of screenshots on Workspaces and other interfaces.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Screenshot Properties (continued)

Field / Element	Description
	Note: IE and Safari don't support this property. If this property is disabled or your current browser doesn't support it, html2canvas is used to take screenshots. If you are upgrading to a new release version, this property is set to false by default.
Enable capturing the full page when taking a screenshot *sn_atf.screenshots.capture_full_page*	Enables dynamic resizing of the test frame and capturing the full page while taking a screenshot. **Note:** By default, the property is set to false because it can significantly slow test execution time If the property is disabled, only the portion of the page that is visible in the test frame is captured.
Screenshot timeout *sn_atf.atf_test_runner.screenshot_timeout*	Skips a screenshot capture attempt in the Client Test Runner if it exceeds this value in seconds. Users should review performance settings and browser caches on affected client systems before increasing this value.
Number of pixels for screenshot height *sn_atf.atf_test_runner.testframe.min_height*	Numeric value representing the number of pixels for the screenshot height. Default value: 600
Number of pixels for screenshot width *sn_atf.atf_test_runner.testframe.min_width*	Numeric value representing the number of pixels for the screenshot width.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Screenshot Properties (continued)

Field / Element	Description
	Default value: 800

Custom UI Page Data Capture Properties

Field / Element	Description
Enable tests with Custom UI steps to capture page data each time they are run. *sn_atf.page_data_capture.enabled*	When true, Custom UI test steps retrieve page data every time the test runs. Set this property to true when developing custom UI pages to always run tests on the most recent page version. When false, Custom UI test steps do not retrieve page data unless the test designer manually selects **Retrieve Components** during test design. Set this property to false when UI development is complete to enable faster test runs.

Test Runner Properties

Field / Element	Description
Test runner timeout *sn_atf.runner.heartbeat.timeout*	If there is no heartbeat from the test runner within this period of time in seconds, the status changes from online to offline. The value of this property should be between 120 and 1800. Default value: 120
Test runner heartbeat interval *sn_atf.runner.heartbeat.interval*	Time interval in seconds for sending a heartbeat from the test runner to the server.
Offline test runner retention interval *sn_atf.runner.offline_retention.timeout*	If an offline test runner does not communicate with the system for this period of days, the system deletes that test runner.

Test Suite Report Properties

Field / Element	Description
Test suite report properties *sn_atf.schedule.reports.suite.aging_threshold*	The number of test suite results to display in the test suite aging report.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Email Properties

Field / Element	Description
Boolean value for results displayed in scheduled suite result emails. *sn_atf.schedule.suite_result_email.only_show_failed_results*	When true, the scheduled suite test result emails only show results that failed. When false, displays all results.
Maximum number of test results to be displayed in scheduled suite result emails. *sn_atf.schedule.suite_result_email.max_test_history*	Maximum number of test results to be displayed in scheduled suite result emails.
Maximum depth when printing suite results in suite result emails. *sn_atf.schedule.suite_result_email.max_depth*	Maximum depth when printing suite results in suite result emails.
Color to indicate an ATF test failed in scheduled suite result emails. *sn_atf.schedule.suite_result_email.fail_color*	Hexadecimal code for color to indicate an ATF test failed in scheduled suite result emails.
Color to indicate an ATF test errored in scheduled suite result emails. *sn_atf.schedule.suite_result_email.error_color*	Hexadecimal code for color to indicate an ATF test errored in scheduled suite result emails.
Color to indicate an ATF test passed in scheduled suite result emails. *sn_atf.schedule.suite_result_email.pass_color*	Hexadecimal code for color to indicate an ATF test passed in

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Email Properties (continued)

Field / Element	Description
	scheduled suite result emails.
Color to indicate an ATF test was skipped in scheduled suite result emails. *sn_atf.schedule.suite_result_email.skip_color*	Hexadecimal code for color to indicate an ATF test was skipped in scheduled suite result emails.
Color to indicate an ATF test was canceled in scheduled suite result emails. *sn_atf.schedule.suite_result_email.cancel_color*	Hexadecimal code for color to indicate an ATF test was canceled in scheduled suite result emails.

Step configurations

Step configuration records define how each type of step behaves.

Automated Test Framework Step Config record

The step config record controls how a test step of this type behaves.

Fields

Field / Element	Description
Name	The name for this test step configuration.
Active	**True** if this step configuration is active. Otherwise, **false**.
Step Environment	The step environment in which a step with this configuration can run.
Category	The category assigned to a step with this configuration.
Application	The application scope in which the system runs steps with this configuration.
Batch order constraint	Constrains where a step with this configuration can appear in a test: • **None**: A step based on this configuration can appear at any point in a test. • **Start Batch Execution**: If a test includes this a step based on this configuration, it must be the first step in the test.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Fields (continued)

Field / Element	Description
	• **Run in the middle of an execution**: If a test includes a step based on this configuration, it cannot be the first or last step. • **Stop Execution**: If a test includes a step based on this configuration, it must be the last step in the test.
Class type	For custom step configurations, this field is always **Script**.
Order	An integer specifying where steps with this configuration appear in the step list on the Add Test Step dialog. For more information, see the example using the Order field.
Template reminder	The instructions that appear when a step with this configuration is included in a test as part of a template. For more information, see theexample of using the Template reminder field.
HTML description	The text that appears when the cursor highlights a step with this configuration on the Create New Step dialog. For more information, see the example using the HTML description field.
Description generation script	Generates the text that describes a step when a test includes it. For more information, see the example using the description generation script.
Step execution script	Script (javascript) that runs when a step with this config runs.
Input Variables related list	The variables that act as inputs for a step with this config.
Output variables related list	The variables that act as outputs for a step with this config.

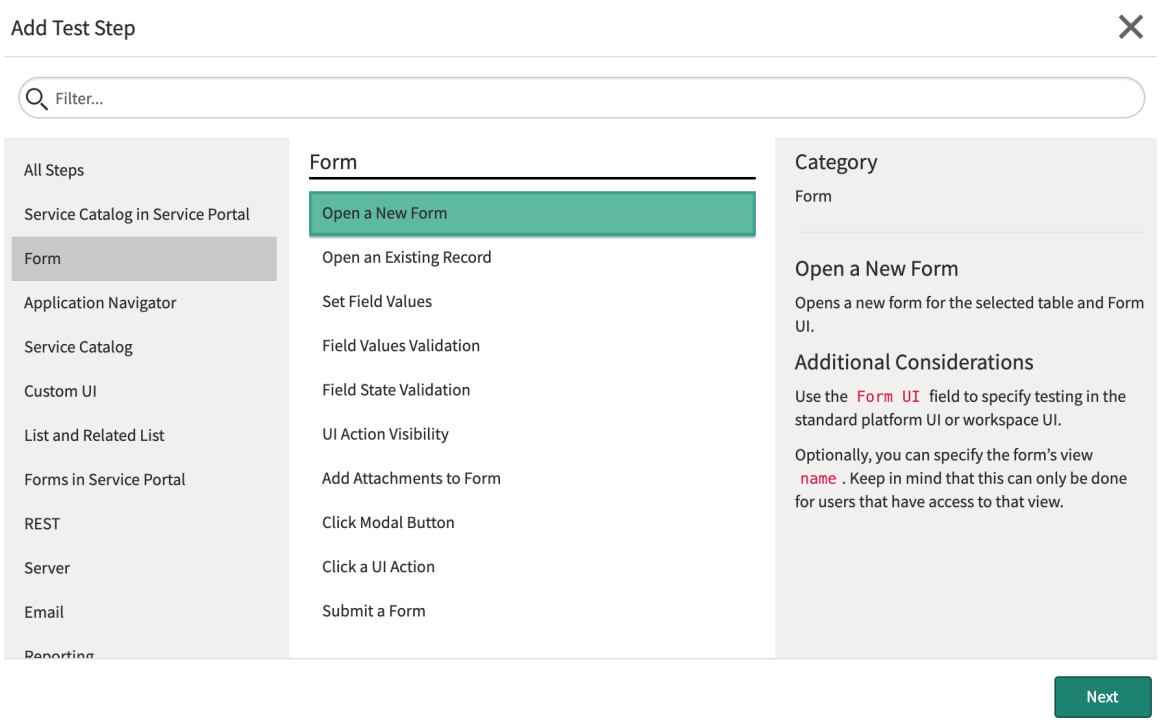
Examples of step config field values

Examples of where the system displays values assigned to some of the step config fields.

Order field example

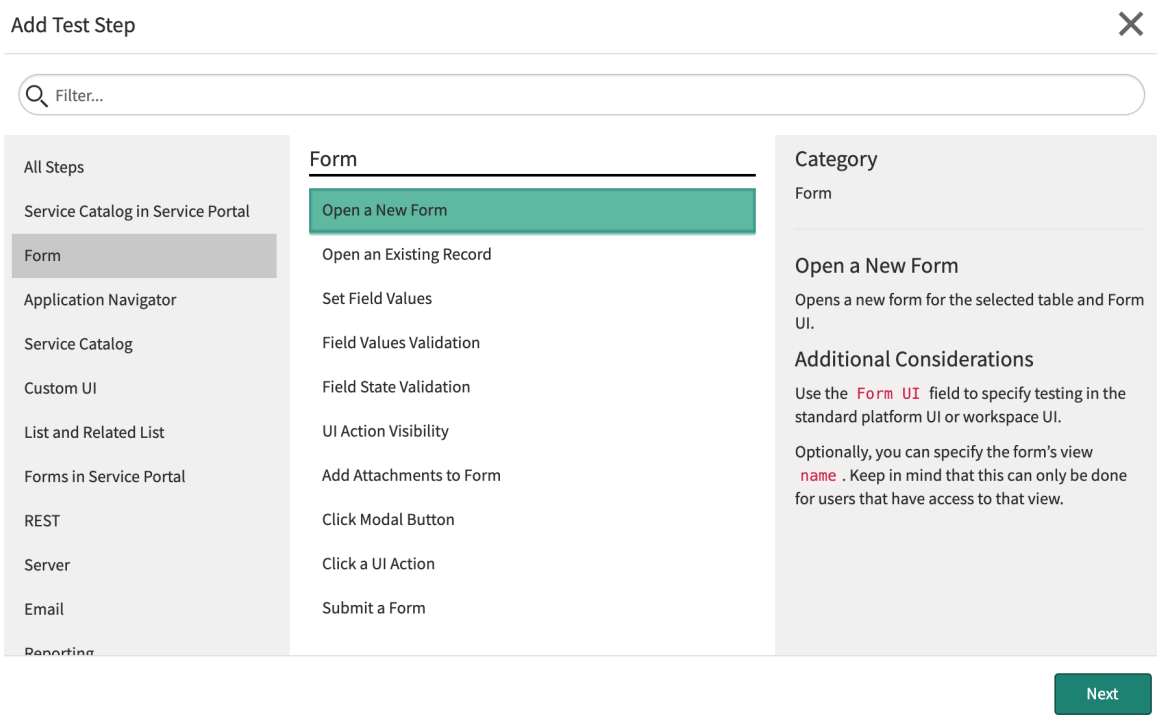
The steps in the middle column are sorted according to the values of the **Order** field.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Category field example

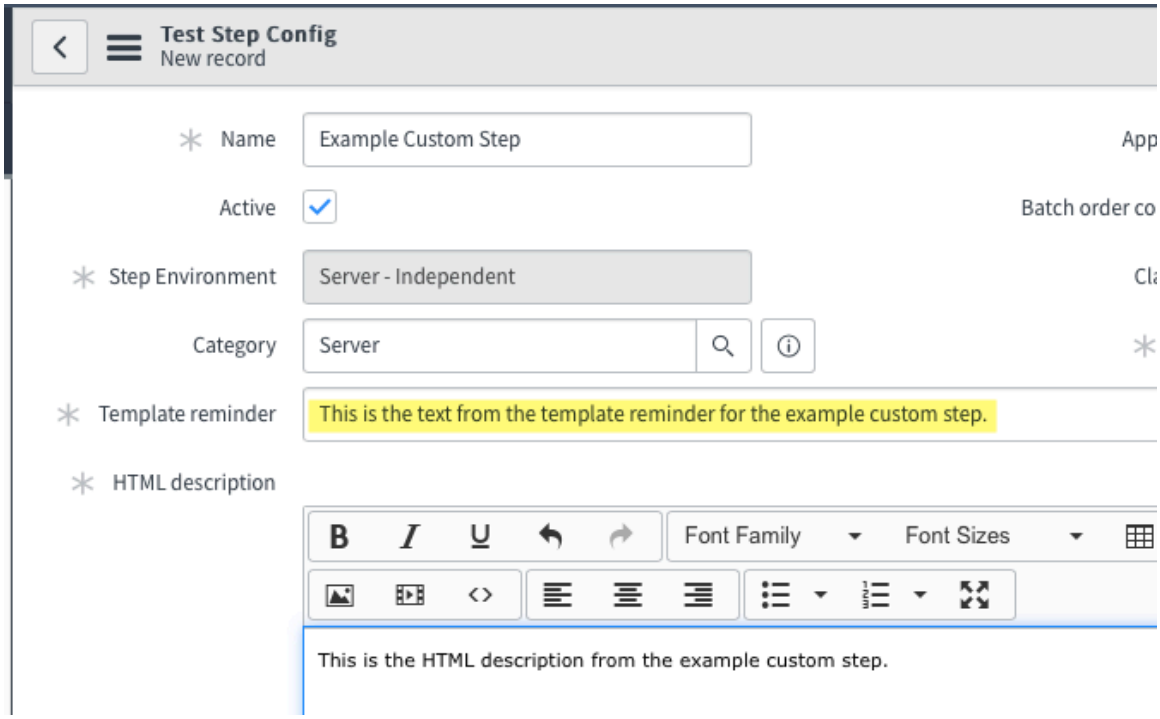
The labels in the left column show the categories for available steps. You can filter the list in the middle column, by selecting a category in the left column.



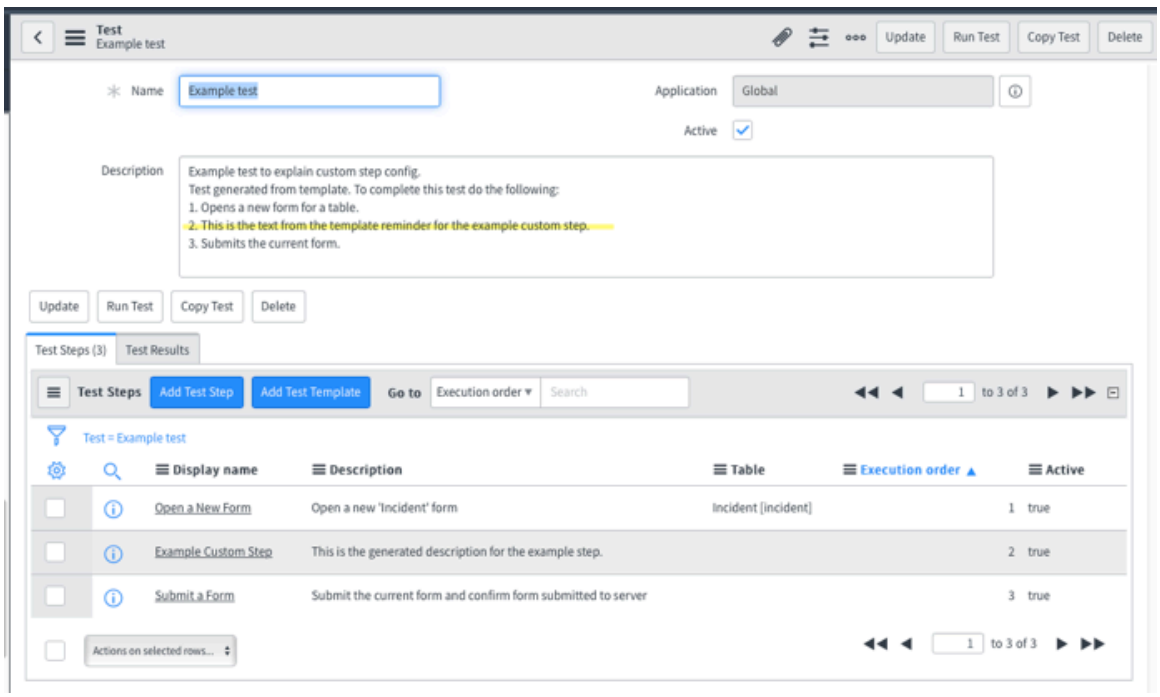
Template reminder example

Here is a portion of the new record form for a test step config for Example Custom Step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



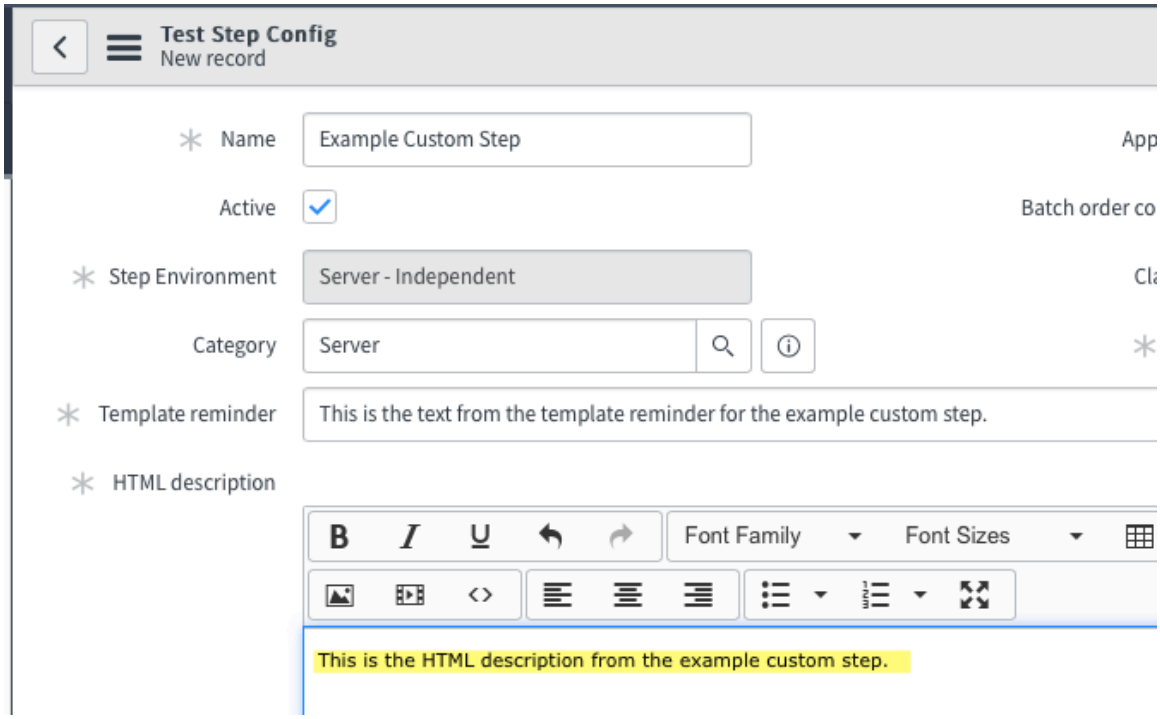
When you add a template to a test, the system generates a set of instructions for completing the template steps and saves them in the test **Description** field. The text in the step config's **Template reminder field** appears as the instructions for the corresponding step.



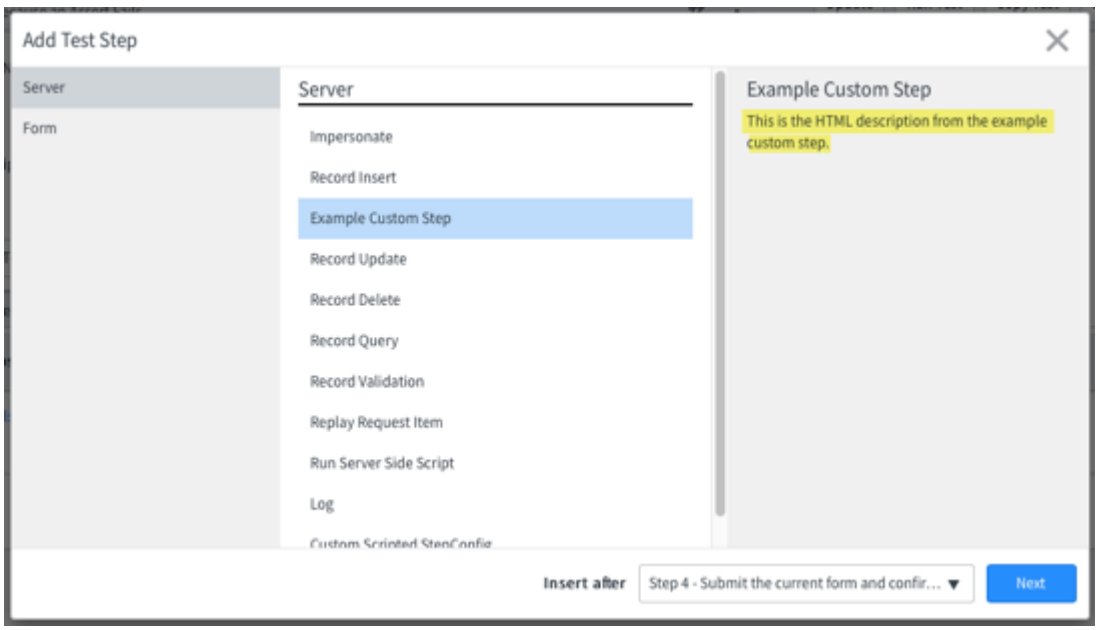
HTML description example

Here is a portion of the new record form for a custom step named Example Custom Step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



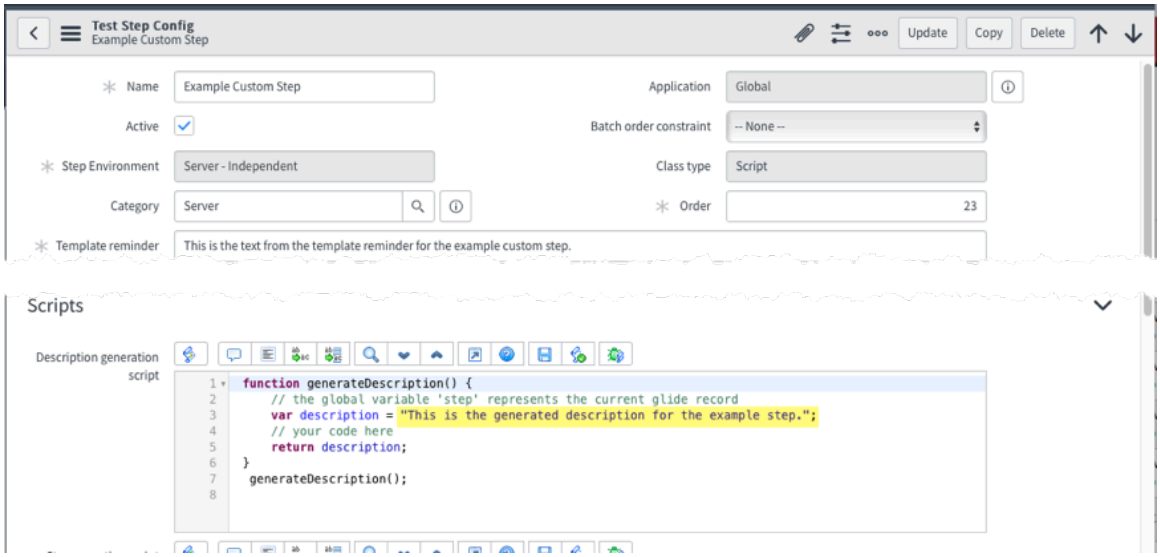
Here is how the step appears in the Add Test Step dialog.



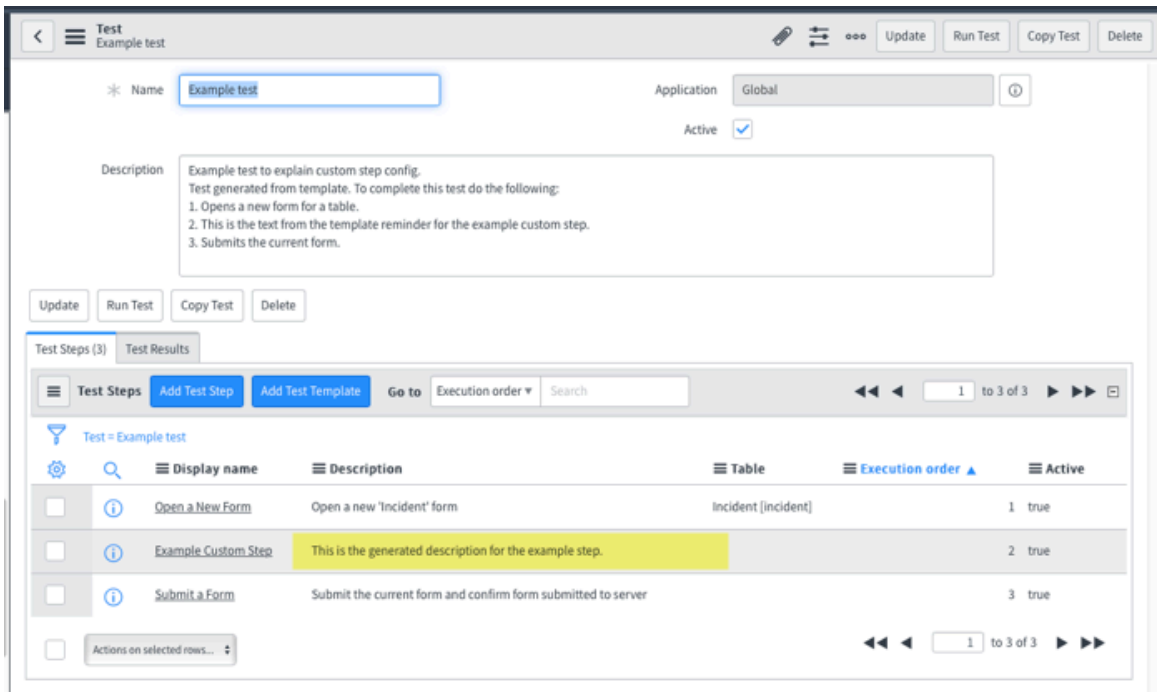
Description generation script example

Here is a portion of the new record form for a custom step named Example Custom Step.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



When a step of this type is included in a test, the generated description appears in the Test Steps related list.



Step execution scripts

In a step configuration record, the step execution script field determines what a step with this configuration does when it runs.

Step inputs

The input variables to a step are determined by the inputs related list in the step configuration record. The *`inputs`* parameter to `executeScript()` gives the script access to these variables. For example, if the inputs related list contains two records, *`var1`* and *`var2`*, the script can reference *`var1`* with the expression `inputs.var1` and can reference *`var2`* with `inputs.var2`.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Step outputs

The output variables to a step are determined by the outputs related list in the step configuration record. The *outputs* parameter to `executeScript()` gives the script access to these variables. For example, if the outputs related list contains two records, *out1* and *out2*, the script can reference *out1* with the expression `outputs.out1` and can reference *out2* with `outputs.out2`.

Step result

The *stepResult* parameter provides access to an API that controls whether the step passes or fails. It also determines the message the step writes to the log.

The method `stepResult.setSuccess()` causes the step to succeed. The method `stepResult.setFailed()` causes the step to fail.

The method `stepResult.setOutputMessage()` sets the message to write to the log when the step succeeds or fails. It takes one parameter: the string to write to the log. If the script calls `stepResult.setOutputMessage()` more than once, the most recent value set overwrites any previous value.

Example: Record Query step execution script

```
   (function executeStep(inputs, outputs, stepResult) {
        if (gs.nil(inputs.table)) {
           stepResult.setOutputMessage(gs.getMessage("The '{0}'
input variable was not specified",
              'table'));
        stepResult.setFailed();
         return;
   }
   var query = new GlideRecord(inputs.table);
   query.addEncodedQuery(inputs.field_values);
   query.query();
   if (!query.next()) {
        stepResult.setOutputMessage(gs.getMessage("No records
matching query:\n{0}",
           inputs.field_values));
         stepResult.setFailed();
   } else {
        stepResult.setSuccess();
        outputs.table = inputs.table;
        outputs.first_record = query.getUniqueValue();
        stepResult.setOutputMessage(gs.getMessage("Found {0}
{1} records matching query:\n{2}",

[query.getRowCount(),
                                                inputs.table,

inputs.field_values]));
   }
   }(inputs, outputs, stepResult));
```

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example: Custom scripted step configs

```
   ((function executeStep(inputs, outputs, stepResult, timeout)
{
// Waits up to the timeout for some asynchronous logic to
finish
// This script checks for completion once a second for up to 60
seconds
var counter = 1;
// Try for up to 60 seconds
while (counter <= timeout) {
 // If the asynchronous logic is finished, return "true" to
pass the step
 // isMyAsyncLogicFinished() can be replaced with any
asynchronous event that needs to be tested
 if (isMyAsyncLogicFinished()) {
  stepResult.setOutputMessage("Success!");
  stepResult.setSuccess();
  return;
 }
 // Wait one second, and log the total number of seconds waited
 gs.info("Waited " + counter + " seconds for asynchronous logic
to finish");
 sn_atf.AutomatedTestingFramework.waitOneSecond();
 counter++;
}
// If this point is reached, the retry loop ran out of tries;
return false to fail the step
stepResult.setOutputMessage("FAILURE: Timed out after waiting
for " + timeout + " seconds");
stepResult.setFailed();
}(inputs, outputs, stepResult, timeout));
```

Note: The above example can also be used for Run Server Side script by replacing `stepResult.setSuccess()"` and `stepResult.setFailed()` with `return true` and `return false`.

Step description generation script

In a step configuration record, the step description generation script field determines the step description that the system generates when a step of this type is added to a test.

For an example showing how where the description generated by this script appears, see Description generation script example.

Step

The *step* parameter to `generateDescription()` gives the script access to the step object, which it turn gives access to the input variables as defined in the step configuration record. (Input variables are defined in a related list.)

For example, if – in a step config record – the inputs related list contains two records: *var1* and *var2*, the script can reference *var1* with the expression `step.inputs.var1` and can reference *var2* with `step.inputs.var2`.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example: Record Query description generation script

```
   (function generateDescription(step) {
   var td = GlideTableDescriptor.get(step.inputs.table);
   if (!td) {
        gs.log("Invalid table name in test step: " +
step.inputs.table);
        return gs.getMessage("Set field values");
   }
   var descriptionGenerator = new
ATFStepDescriptionGenerator();
   var description = gs.getMessage("There should be at least
one record in '{0}' matching " +
                    "a query of\n{1}",
                    [step.inputs.table.getDisplayValue(),

descriptionGenerator.getConditionDescription(step.inputs.table,
step.inputs.field_values)]);

   return description;
   })(step);
```

Add output variables to scripted steps

Execute the following steps to add additional outputs in Run Server Side Script and Custom Scripted StepConfig test steps.

Adding outputs in Run Server Side Script test step

Modify the test scripts of Run Server Side Script test step to create additional outputs of your choice.

Before you begin

Role required: admin or atf_test_admin

Procedure

1. Navigate to **All** > **Automated Test Framework (ATF)** > **Administration** > **Step Configurations**.
2. Search and select Run Server Side Script.
   The read-only Test Step Config form shows up.

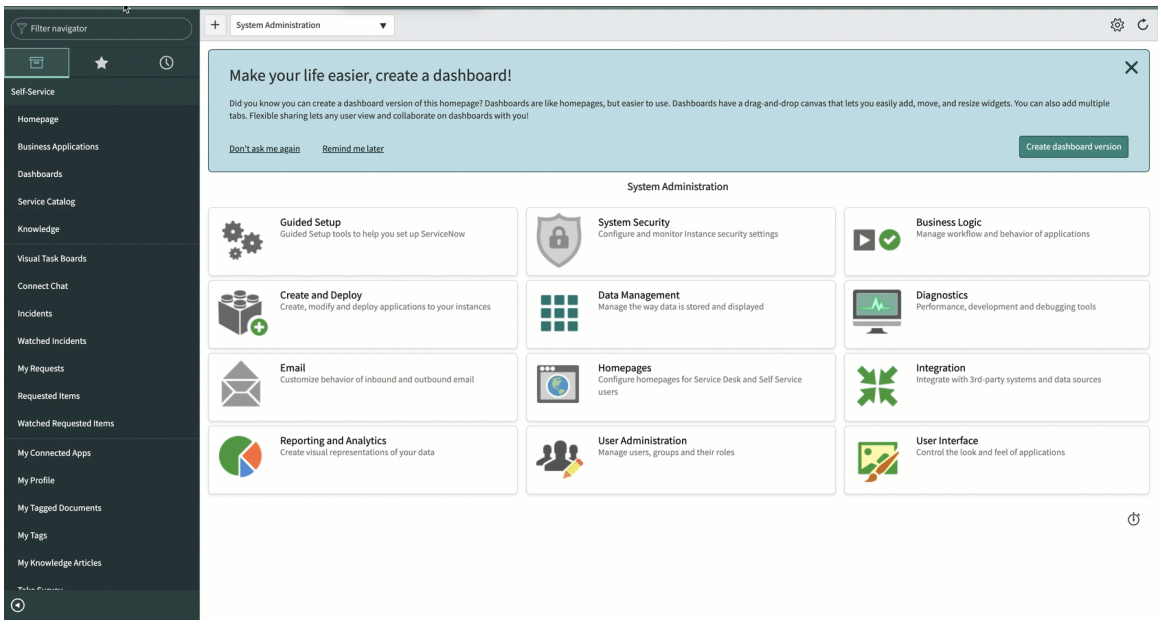
   Note: Although the form is read-only, new output variables can be created.

3. Scroll down to the Output Variables related list.
   The list of default output variables shows up.
4. Click **New** to create a new output variable.

Field	Input Value
Type	Type of output variable
Application	Scope of the user
Label	User-facing name of the output variable
Column name	Variable name used in the script

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Input Value
Max length	Length of the data type. **Note:** This field doesn't show up for all data types.



The new output variable gets added to the Output Variable related list.

5. Navigate to **Automated Test Framework (ATF)** > **Tests**.
6. Select the test where you want to implement the new output variable.
7. Click **Add Test Step**.
8. Search and select Run Server Side Script test step.
9. Follow the instructions to change outputs mentioned in the **Test script**.

```
outputs.<column_name> = "<desired value>";
```

10. Click **Submit**.
    The newly created output variables are now ready to be used in any subsequent steps of the test.

Adding outputs in Custom Scripted StepConfig test step

Copy the Custom Scripted StepConfig test step and customize the copied version by adding additional outputs.

Before you begin

Role required: admin or atf_test_admin

Procedure

1. Navigate to **All** > **Automated Test Framework (ATF)** > **Administration** > **Step Configurations**.
2. Search and select Custom Scripted StepConfig.
   The read-only Test Step Config form shows up.

Note: Since the form is read-only, you must copy the test step to customize it to add more output variables.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

3. Select **Copy** to have a copied version of the test step.

   Note: The copied version is no longer read-only.

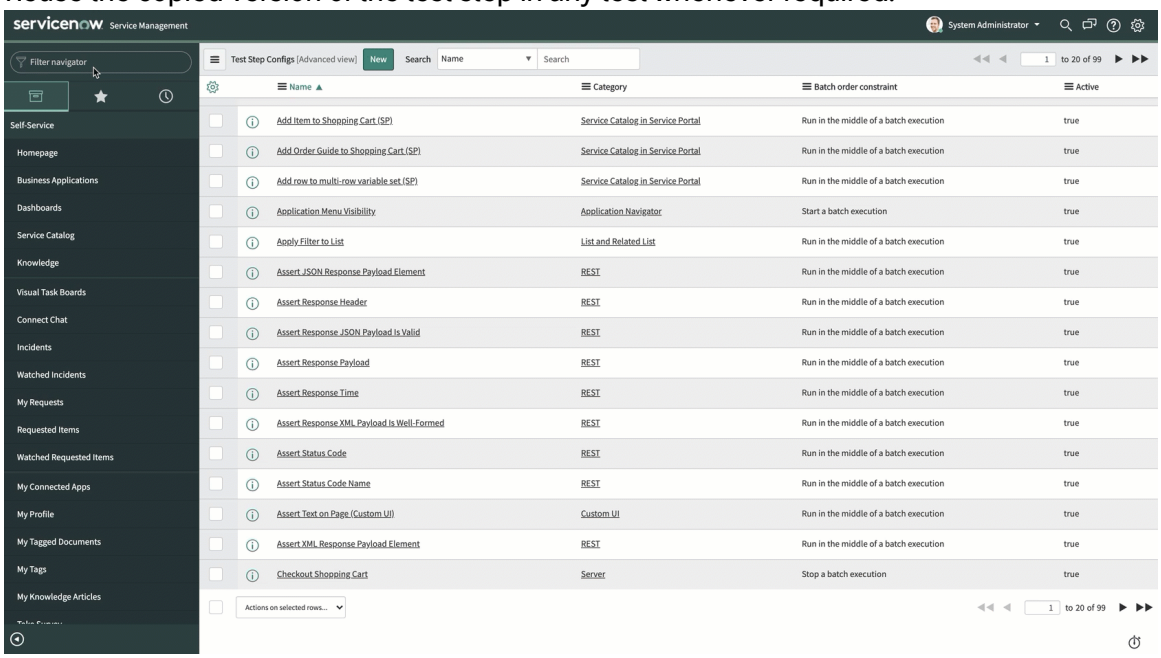
4. Add additional output variables in the copied version by implementing the following steps:

   a. Select **New** under Output Variables related list.

   b. Modify **Step execution script** to add more output variables.
   See Adding outputs in Run Server Side Script test step to add more output variables.

   Note: You can use these steps to customize the test step only in the copied version.

5. Reuse the copied version of the test step in any test whenever required.



Test templates

The Test Templates module opens a list of available templates. From this module, you create, view, and edit test templates.

To view or edit a test template, click the row for the template you want. For more information, see Edit automated test steps template.

For information how to add a new template, see Create an automated test steps template. For information on how to use a template when creating a test, see Add a predefined list of steps (template) to an automated test.

Automated Test Template record

The Test Template record contains information about one test template.

Fields

Field / Element	Description
Name	The name of this test template.
Application	Application scope in which the system runs this test or test suite.
Test Template	The tests to include in this template.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Fields (continued)

Field / Element	Description
Description	(Optional) Enter a description to identify the purpose of this test template.

Step configuration categories

The Step Configuration Categories module opens a list of records specifying the step categories on the Add Step dialog. From this module, you can add, delete, and edit these categories.

Categories are used for filtering the step list in the Add Step dialog. For more information, see Category field example.

Test step config category form

On the Test Step Config Category form, you specify a retention policy for a set of records on a given test results table.

Fields

Field / Element	Description
Name	The name for this step config category.
Step Environment	The step environment for this step config category.
Display name	The category name that appears in the middle column of the Add Test Step dialog when this category is selected. See an example in Step Config Category Display name.
Application scope	The application scope to which this policy applies.

Table cleanup

The Table Cleanup module opens a list of records specifying the retention policies for test result and test suite result tables and the records within them. From this module, you can view and modify these policies.

By default, the system deletes system records related to test results and test suite results 30 days after creation. To modify the retention policies for a table and its records, click the table for which you want to modify policies. For more information, see Modify data retention policy for ATF test results.

Note: Table cleanup policies are platform-specific policies. See Table cleaner for more information.

Autoflush form

On the Auto Flush form, you specify a retention policy for a set of records on a given test results table.

Fields

Field / Element	Description
Tablename	The table containing the records to which the policy applies.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Fields (continued)

Field / Element	Description
Matchfield	The field for which the system monitors duration.
Age in seconds	The amount of time (in seconds) the system wait before deleting the records.
Active	**True** if this policy is active. Otherwise, **false**.
Application scope	The application scope to which this policy applies.
Cascade delete	If checked, the system deletes all matching records plus any records referring to them. If unchecked, the system deletes matching records, but not records referring to them.
Conditions	The filter conditions defining the records in this table to which the policy applies.

Note: Table cleanup policies are platform-specific policies. See Table cleaner for more information.

Step environments

A test step environment specifies where the step executes (for example, server versus browser). In this release, custom step configs can use only the Server-Independent environment.

In this release, you cannot add custom test step environments.

Performance Analyzer

The ServiceNow® Performance Analyzer application enables you to access page load time data, trends, and waterfall visualizations directly from your instance. It also helps you to identify page performance issues and bottlenecks in UX framework pages independently.

Get started

Explore



Learn about Performance Analyzer concepts and features.

Configure



Configure Performance Analyzer environment and access.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Get started (continued)

Test	Reference
	
Test application performance with Performance Analyzer.	Get additional information about Performance Analyzer.

Exploring Performance Analyzer

Learn about Performance Analyzer concepts and features.

Performance Analyzer overview

The Performance Analyzer application enables you to access page load time data, trends, and waterfall visualizations directly from your instance. It also helps you to identify page performance issues and bottlenecks in UX framework pages independently.

- View a waterfall or request flows displayed as executions sequences in a structured tree-like timeline.
- Receive page load time data directly on the instance.
- Access aggregated metrics by application and routes.

Performance Analyzer features

Filter by duration

Select a duration from the last fifteen minutes to the last seven days to view the aggregate data from that time period for each of your applications. The aggregate data includes total UI time, download time, trends, and a comparison to previous performance over that duration.

Application metrics

Select an application to filter metrics by total UI time and download time, as well as view page route performance.

Page route metrics

Select a page route within an application to view interaction metrics including timestamps, interaction types, total UI time, download time, and record.

Interaction metrics

Select an interaction within a page route to view a waterfall with details for each resource on the page during that interaction.

Configuring Performance Analyzer

Configure Performance Analyzer environment and access.

Starting with the Zurich release, Performance Analyzer is available on your instance automatically.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

For access to Performance Analyzer on earlier instances, install Performance Analyzer from the ServiceNow® Store. Visit the ServiceNow Store website to view the application and for information about submitting requests to the store.

Testing applications with Performance Analyzer

Get information on how to test applications with Performance Analyzer.

Get metrics by duration

Filter metrics in the Performance Analyzer by duration from the last fifteen minutes to the last seven days.

Get application metrics

Filter metrics by application in the Performance Analyzer.

Get page route metrics

Filter metrics for an application by page route in the Performance Analyzer.

Get interaction metrics

Filter metrics for an application's page route by interaction in the Performance Analyzer.

Get metrics by duration

Filter metrics in the Performance Analyzer by duration from the last fifteen minutes to the last seven days.

Before you begin

Role required: admin

Procedure

1. Navigate to **All** > **Next Experience Developer Tools** > **Performance Analyzer**.
2. In the Duration field, select a duration from the list.

Result

The Performance Analyzer displays aggregate metrics for the selected duration.

Get application metrics

Filter metrics by application in the Performance Analyzer.

Before you begin

Role required: admin

Procedure

1. Navigate to **All** > **Next Experience Developer Tools** > **Performance Analyzer**.
2. From the Application field, select an application.

Result

The Performance Analyzer displays page-route metrics for the selected application including total UI time, download time, and trends.

Get page route metrics

Filter metrics for an application by page route in the Performance Analyzer.

Before you begin

Role required: admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **Next Experience Developer Tools** > **Performance Analyzer**.
2. From the Application field, select an application.
3. From the Page route field, select a page route.

Result

The Performance Analyzer displays interaction metrics for the selected page route including total UI time and download time.

Get interaction metrics

Filter metrics for an application's page route by interaction in the Performance Analyzer.

Before you begin

Role required: admin

Procedure

1. Navigate to **All** > **Next Experience Developer Tools** > **Performance Analyzer**.
2. From the Application field, select an application.
3. From the Page route field, select a page route.
4. From the Timestamp field, select a timestamp.

Result

The Performance Analyzer displays a waterfall with details for each resource on the page during that interaction.

Performance Analyzer reference

Get additional information about Performance Analyzer.

Get help with Performance Analyzer

To get help with Performance Analyzer, your ServiceNow instance, plugins, permissions, and more, watch a short video to contact the ServiceNow admin who works in your company.

Get help with Performance Analyzer

To get help with Performance Analyzer, your ServiceNow instance, plugins, permissions, and more, watch a short video to contact the ServiceNow admin who works in your company.

Get help now from your Now Support administrator

The ServiceNow admin in your company can help you with things like activating plugins, kicking off upgrades, unlocking accounts, updating user permissions, and much more.

https://player.vimeo.com/video/1056112088?h=1a5cd3b245&badge=0&autopause=0&player_id=0&app_id=58479

Deploying applications

Deliver your application to users by publishing it to a live production environment.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Deploying applications on the Next Experience

Manage deployments using pipelines in AEMC



Use App Engine Management Center (AEMC) to manage, review, and publish applications built in App Engine Studio, Creator Studio, and ServiceNow Studio.

Manage deployments using pipelines in AEMC

The App Engine Management Center (AEMC) provides a unified experience for deploying custom applications built in App Engine Studio, Creator Studio, and ServiceNow Studio. AEMC uses Pipelines and Deployments to move applications between instances as they are prepared for release. In AEMC, you can customize your pipeline, review, approve, and reject deployment requests, and schedule deployments all in one centralized location.

Deploying applications in the Core UI

ReleaseOps



ReleaseOps enables you to create custom pipelines and automate changes across your pipeline instances using update sets. ReleaseOps can help to improve the predictability and reliability of deploying changes and reduce the risk of unwanted changes releasing to production instances.

ServiceNow application repository



The ServiceNow Application Repository is a central location for all scoped applications that are published by all ServiceNow customers. After you develop and test a custom application, you can make the application available to company instances by publishing it to this repository.

System update sets



An update set is a group of configuration changes that can be moved from one instance to another. This feature enables administrators to group a series of changes into a named set and then move them as a unit to other systems for testing or deployment.

ReleaseOps

ReleaseOps automates the deployment process using update sets and customizable pipelines. ReleaseOps orchestrates the entire process of deployment, including moving changes between instances, running scans and tests, and releasing changes, while providing transparency into the process for release managers and teams.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

ServiceNow application repository

The ServiceNow Application Repository is a central location for all scoped applications that are published by all ServiceNow customers. After you develop and test a custom application, you can make the application available to company instances by publishing it to this repository.

System update sets

An update set is a group of configuration changes that can be moved from one instance to another. This feature enables administrators to group a series of changes into a named set and then move them as a unit to other systems for testing or deployment.

System update sets

An update set is a group of configuration changes that can be moved from one instance to another. This feature allows administrators to group a series of changes into a named set and then move them as a unit to other systems for testing or deployment.

Get started

Explore  Learn about System Update Sets features and concepts.	Configure  Configure System Update Sets in your instance.	Work with update sets  Working with System Update Sets in your instance.
Batched Update Sets  Working with batched System Update Sets in your instance.	Reference  Learn about System Update Sets general guidelines and system properties.	

Exploring System Update Sets

An update set is a group of configuration changes that can be moved from one instance to another. Update sets enable developers to create functionality on a non-production instance, and promote the changes to another instance for testing or deployment.

System update sets overview

An update set is an XML file that contains:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- A set of record details that uniquely identify the update set.
- A list of configuration changes.
- A state that determines whether another instance can retrieve and apply configuration changes.

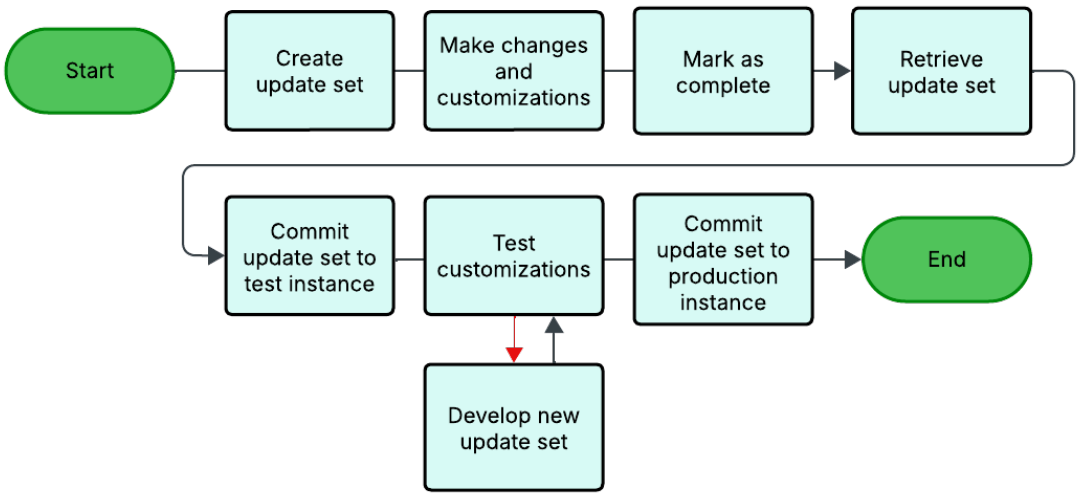
Users

User	Description
Admin	Creates, compares, and merges update sets while also managing how update sets store, retrieve, preview, and apply configuration changes between instances.
Developer	Creates update sets for specific versions of an application and specifies which application tables to track.

Workflows

A common process for developing customizations with update sets involves moving changes from development to test and production instances.

Three-step import process



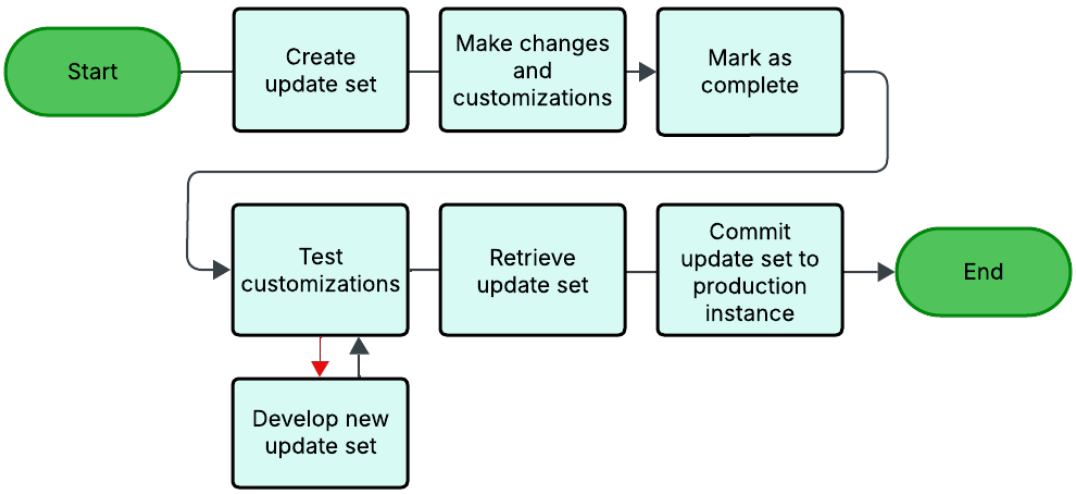
1. Create an update set on the development instance.
2. Make customizations and changes on the development instance.
3. Mark the update set as complete.
4. Log in to the test instance and retrieve the completed update set from the development instance.
5. Commit the update set on the test instance, and test customizations thoroughly.
6. If the update set has problems in the test instance, repeat steps 1–5 to develop the fix on the development instance with another update set.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

7. Identify a common path for update sets to move from instance to instance and maintain that model. Never migrate the same update set from multiple sources. Move update sets from dev to test and then from test to production.
8. Commit the update set on production. If the update set required a fix, commit both update sets in the order they were made.

If your development environment consists of only two instances, you can combine your development and testing instances into a single staging instance.

Two-step import process



1. Create an update set on the staging instance.
2. Make customizations and changes on the staging instance.
3. Mark the update set as complete.
4. Test customizations thoroughly on the staging instance.
5. If the update set has problems, repeat steps 1–4 to develop the fix on the staging instance with another update set.
6. Log in to the production instance and retrieve the completed update set from the staging instance. If the update set required a fix, retrieve both update sets.
7. Commit the update set on production. If the update set required a fix, commit both update sets in the order they were made.

Benefits

Benefit	Feature	Users
Create an update set to store local changes.	Create and select an update set as the current set	Developer
Select the current update set to store local changes.	Select the current update set in Unified Navigation	Admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Benefit	Feature	Users
Commit an update set to prepare it for distribution.	Commit an update set	Admin
Compare update sets to determine what differences they contain.	Compare local update sets	Admin
Create an external file from an update set.	Save an update set as a local XML file	Admin
Retrieve update sets from remote instances.	Retrieve an update set	Admin
Back out changes applied from an update set.	Back out an update set	Admin
Set system properties related to update sets.	Update sets properties	Admin
Track customizations to application tables, fields, and records.	Customizations tracked by update sets	Admin
Batch update sets together so you can preview and commit them in bulk.	Working with batched update sets	Admin

Use cases

Choose update sets or the application repository depending on the result that you want to achieve. Don't use both Update Sets and the Application Repository for scoped app development. Combining the usage of both results in skipped changes and commit errors. After installing an app from the Application Repository, continue its usage for development and publishing. Update sets can only transfer limited application files. For larger transfers, export and import data using import sets or web services.

Update sets

- Storing changes to a base system or installed application.
- Storing and applying a particular version of an application.
- Producing a file for export.

Application repository

- Installing and updating applications on all company instances.
- Managing application update sets.
- Restricting access to applications from the same company.
- Deploying completed applications to end users.

⚠ **Warning:** Update sets track changes to applications and system platform features. Update sets from older releases can be loaded onto newer instances without problems. Moving update sets from newer to older releases, or between different ServiceNow Store app versions, should be avoided due to potential compatibility issues and risks of data loss.

What to explore next

To learn more about configuring, using, and managing system update sets see:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Configuring System Update Sets
- Working with update sets
- Working with batched update sets
- Update set transfers

Configuring System Update Sets

Configure your instance with tracked app installations or merge update sets into a single update set.

Enable application installation tracking through an update set

You can add specific applications or versions as an app installation and configure to track them in an update set.

Before you begin

Role required: admin

This topic is applicable only to global update sets and is enabled by default.

Procedure

1. Select **All** and enter **sys_store_app.list**.
2. Select the update set to configure in the store apps list.
3. Select **Add app install to current update set** in the related links section.
   A message appears notifying you to confirm tracking app installs in the update set.
4. To configure app installs using plugins, select **All** and enter **v_plugin.list**.
5. Select the plugin to configure from the system plugin list.
6. Select **Add app install to current update set** in the related links section.
7. Select **Update**.
   A message appears notifying you to confirm the app installs for the update set.

Overwrite customizations during an upgrade

Specify which customized objects you want to replace during the next upgrade. By default, the upgrade process skips changes to customized objects.

Before you begin

Role required: admin

Configuration changes you make such as modifying the dictionary record for a table or field are tracked. Each configuration change has a corresponding record in the customer update table, where **Replace on upgrade** is set to `false`.

About this task

When updating to the latest software version, you can replace your custom changes. For instance, if you modified a script as a temporary fix, you'll want to remove that workaround during the upgrade so you can benefit from new improvements and confirm compatibility with future updates.

Procedure

1. Navigate to **All** > **System update sets** > **Local update sets**
2. Select an update set.
3. Select a customized object in the customer updates list.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

4. Select and hold (or right-click) the header and select **Show Latest Update**.
5. Configure the form to add the **Replace on upgrade** field, if necessary.
   a. Hold (or select and hold (or right-click)) the header and select **Configure**.
   b. Select **Replace on upgrade** from the available list and add it to the form.
   c. Select **Save**.
6. Select the **Replace on upgrade** check box and select **Update**.
   The customized object will be replaced on the next upgrade.

Merge update sets

Merge multiple update sets into a single update set.

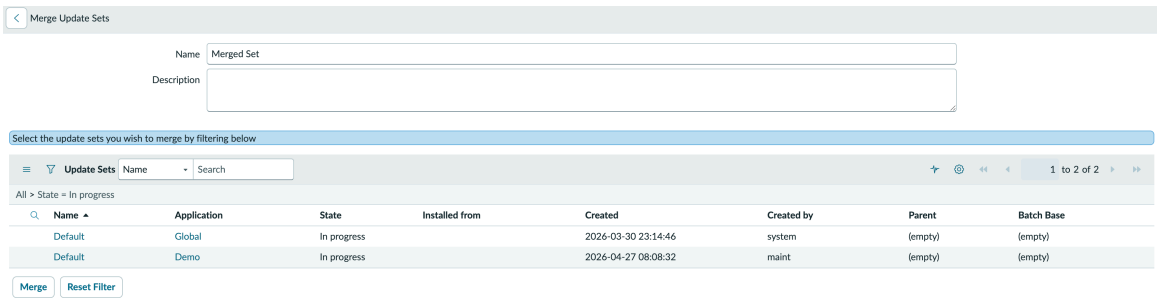
Before you begin

The batch update sets feature accomplishes the same outcome with a more predictable and robust solution. See Update set batching.

Role required: admin

Procedure

1. Navigate to **All** > **System Update Sets** > **Merge Update Sets or Merge Completed Sets**.
2. Filter the list to show only the update set that you want to merge.
   You can only merge update sets that belong to the same application.



3. Enter a **Name** for the new update set.
4. **Optional:** Enter a **Description** for the update set.
5. Select **Merge**.
6. In the confirmation dialog box, select **OK**.
   The most recent change for each object is moved from the original sets to the new set. Only changes that aren't merged into the new set remain in the original sets. A message indicates how many updates were moved and how many were skipped.

   Note: The system determines which record is the most recent by comparing the **Updated** field for the records, NOT the **sys_updated_on** value in the payload.

7. Open the old update set records in the *Merged Update Sets* related list and verify that the correct changes were moved.
8. Delete or empty the original update sets to avoid committing an older change by mistake.

Set up the source instance for an update set

Set up the source instance for an update set.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin
Role required: admin

Procedure
1. If IP address access control is enabled on the source instance, set up the target instance as an exception.
2. On the target instance, navigate to **System Update Sets** > **Update Sources** and select **New**.
3. Specify the connection settings as described in the table.

Field	Description
Name	Enter a unique name for the instance.
Type	Specify whether the remote instance is a development, test, or UAT instance.
Active	Specify whether the local instance can transfer update sets to the remote instance. Update sets can only be transferred to active remote instances. **Note:** Use the Active field to deactivate unwanted remote instances.
URL	Specify the URL of the remote instance using the appropriate transfer protocol. Each remote instance record should have a unique URL. **Note:** Creating duplicate records with the same URL can cause errors. The remote instance must be on the same release family as the local instance. You can't change the URL after the system verifies the connection.
Username	Enter the user on the remote instance who authorizes transferring update sets to the instance. This user account must have the admin user role on the remote instance.
Password	Enter the password of the authorizing user.
Short description	Enter any other relevant information about the remote instance.

4. Select **Test Connection**.
   - If the connection is successful, a confirmation message appears.
   - If the connection fails, a warning message identifies the cause of the failure.
5. If the connection fails, modify the settings to establish connectivity.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- You must establish connectivity before you can save the connection settings.
- You must modify the source instance (for example, change the password).

6. Select and hold (or right-click) the form header and select **Save**.

Save an update set as a local XML file

Administrators can export an update set as an XML file to save a specific version of an application or set of changes.

Before you begin
Role required: admin

About this task
Create an XML file of an update set when:

- The two instances lack network connectivity, preventing the retrieval of update sets or the creation of a data source for import from the source instance.
- You don't want to give administrator credentials for the source instance.
- You want to back up important customizations locally.

Procedure

1. Navigate to **System Update Sets** and select either **Local Update Sets** or **Retrieved Update Sets**.

   Note: **Retrieved Update Sets** must be in a loaded state to be exported.

2. Select an update set that is in the **Complete** state.
3. On the Update Set form, select the **Export to XML** Related Link.
4. Save the XML file.

   When a file is uploaded to a different instance, it’s recognized as a retrieved update set.

Load customizations from a single XML file

Administrators can load an update set XML file to apply a specific version of an application or set of changes.

Before you begin
Roles required: admin

Procedure

1. Elevate privileges to the security_admin role.
2. Navigate to **System Update Sets** > **Retrieved Update Sets**.
3. Select **Import Update Set from XML**.
4. Select **Choose File** and select an XML file.
5. Select **Upload**.
   The customization is available as a retrieved update set with the state **Loaded**.
6. To commit the update set, see Commit an update set.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Update set transfers

When an update set is completed, you can transfer it between instances to move customizations from development, through testing, and into production.

Transferring via ReleaseOps

ReleaseOps automates deployment of changes across your pipeline, increases predictability and reliability of deployments, and reduces the risk of releasing changes to production. See ReleaseOps.

Transferring via IP access control

If IP address access control is active on the source instance or a different datacenter from the target, complete these steps before transferring an update set.

1. Contact Customer Service and Support to find out the IP addresses of all application nodes supporting your instance.
2. On the source instance, navigate to System SecurityIP Address Access Control, and add the IP address as an exception.

Transferring via an update source

If IP address access control is enabled on the source instance, set up the target instance as an exception. See Set up the source instance for an update set.

Transferring via an XML file

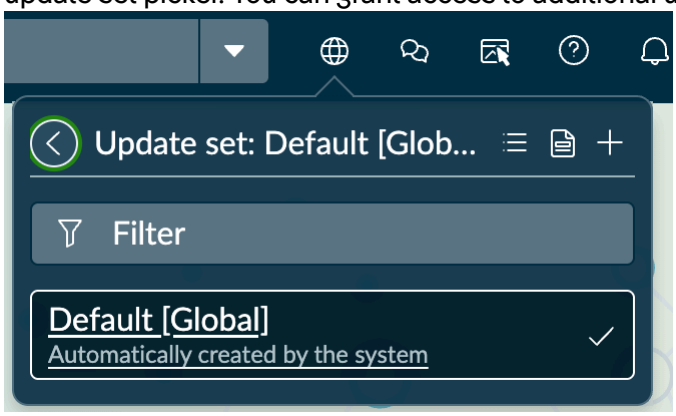
You can unload an update set as an XML file and then transfer it to another instance, see Save an update set as a local XML file.

Grant access to the update set picker

Enable specific roles to access the update set picker.

Before you begin
Role required: admin

About this task
The update set picker appears when selecting the globe icon. The picker enables you to choose an update set for making and tracking customizations. By default, only administrators can use the update set picker. You can grant access to additional users.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Grant the user role read access to the sys_update_set table.
   For more information, see Grant the user role read access.
2. Enable access to the update set picker when selecting the globe icon.
   a. Add glide.ui.update_set_picker.role to the System Properties table.
   b. Set the value of glide.ui.update_set_picker.role to the role that you want to grant access to.
   See Add a system property and Update sets properties for more information.

Working with update sets

Manage your customizations and resolve potential collisions before you move them to another instance.

To create an update set see Create and select an update set as the current set.

When the update set is completed, you can transfer the update set to another instance according to your test process. See Update set transfers.

If a customization is made in the wrong update set

1. Switch to the desired update set.
2. Modify the object (record) that was originally changed. You can make a trivial change, such as adding a field.
3. Save the record.
4. Back out the change just performed, and then save the record again.

   This action ensures that the latest version of the object is included in the desired update set and prevents duplicate updates for the same object in a single update set.

Don't change the **Update Set** field value in a Customer Update record.

Create and select an update set as the current set

Create an update set to store customization changes and select it as the current set.

Before you begin

Role required: admin

Procedure

1. Navigate to **All** > **System Update Sets** > **Local Update Sets** and select **New**.
2. Fill in the form.

Update Set Fields

Field	Description
Name	Enter a unique name for the update set. You can use naming conventions to organize update sets.
State	○ In progress: The update set tracks customizations and displays in progress update sets in the picker. ○ Completed: The update set is complete. Use completed update sets to transfer changes from one instance to another.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
	○ Ignore: The work on an update set is stopped, but you don't want it to be transferred to another instance. Set completed update sets on the production instance to ignore to help prevent additional commits when cloning the instance.
Release date	Enter the date on which you plan to release the update set.
Application	Populates the application scope that is currently selected in the Application picker. All changes in the update set apply only to the current scope.
Description	Enter a description of the update set.

3. Select **Submit** to create the update set or select **Submit and Make Current**.

Select the current update set in Unified Navigation

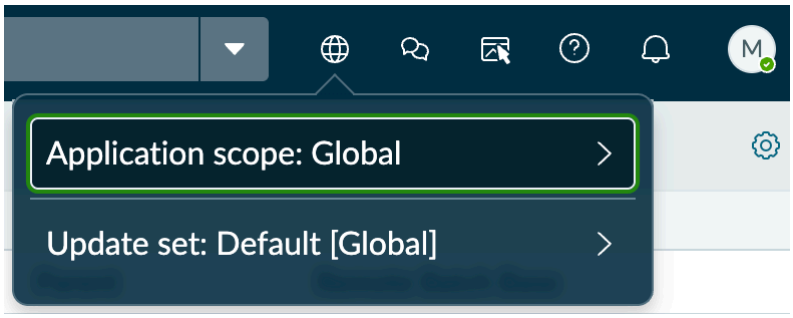
You can change the current update set at any time using the update set picker in Unified Navigation.

Before you begin
Role required: admin

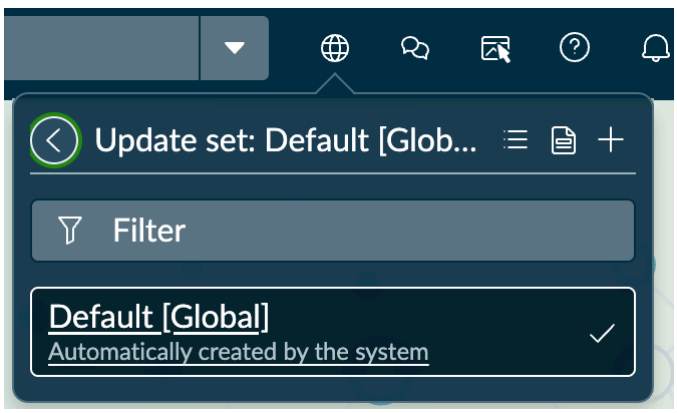
About this task
When you change your application scope, the system automatically switches the current update set to be the application's default update set. However, you can choose a different update set.

Procedure

1. Select the Application scope icon ( ) in Unified Navigation.



2. Select the update set you want to make current from the list available in the **Update set** field, using the **Filter** if needed.



© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The update set you selected is confirmed with a check mark icon.

3. **Optional:** If the update set you need isn't in the list, you can create an update set by selecting the plus icon (+).

What to do next

The update set picker is part of the Next Experience picker. For more information, see Exploring Next Experience pickers.

For more information about application scoping, see Application scope.

View customizations and compare with current version

View the customizations that make up an update set and compare the update to the current version.

Before you begin

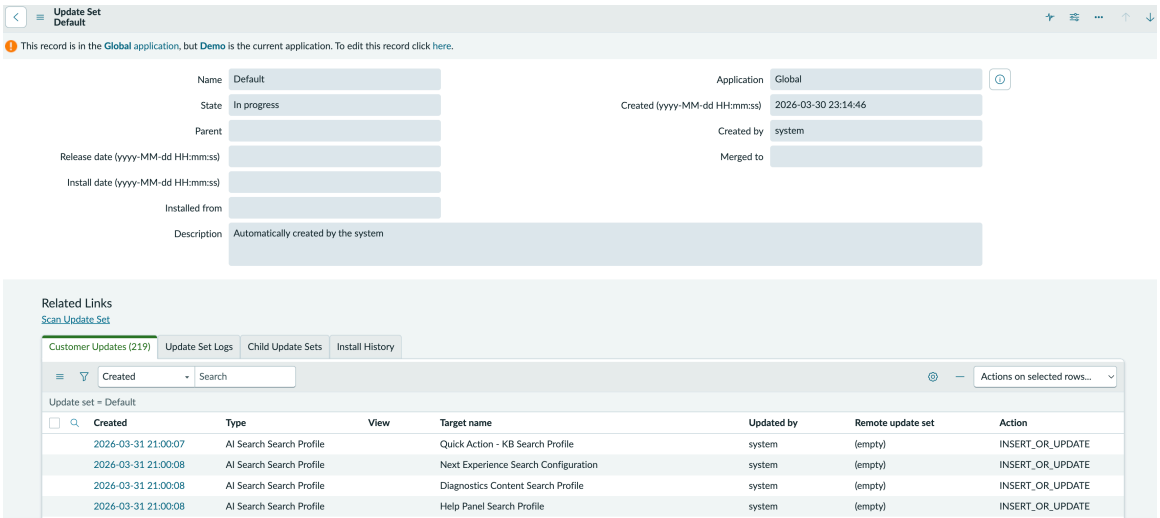
Role required: admin

About this task

The customer updates table contains one record per customized object.

Procedure

1. Navigate to **All** > **System Update Sets** > **Local Update Sets**.
2. Select an update set.
3. View the customer updates related list.



4. Select and hold (or right-click) the update record and select **Compare to Current** to compare to the current version.

Navigation between records

Navigate between a customer update record and the customized object or the application file for the object.

Before you begin

Role required: admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Navigate to **All** > **System update sets** > **Local update sets**.
2. Select an update set.
3. To navigate to a customized object, such as the application menu record, select the **Show Related Record** related link.
4. To navigate to the application file record for the object, select the **Show Application File** related link.
5. To navigate to the current customer update record from a customized object or an app file, select and hold (or right-click) the form header and select **Show latest Update**.

Compare local update sets

Administrators can preview local and remote (retrieved) update sets and compare the sets with one another to resolve conflicting changes.

Before you begin

Role required: admin

About this task

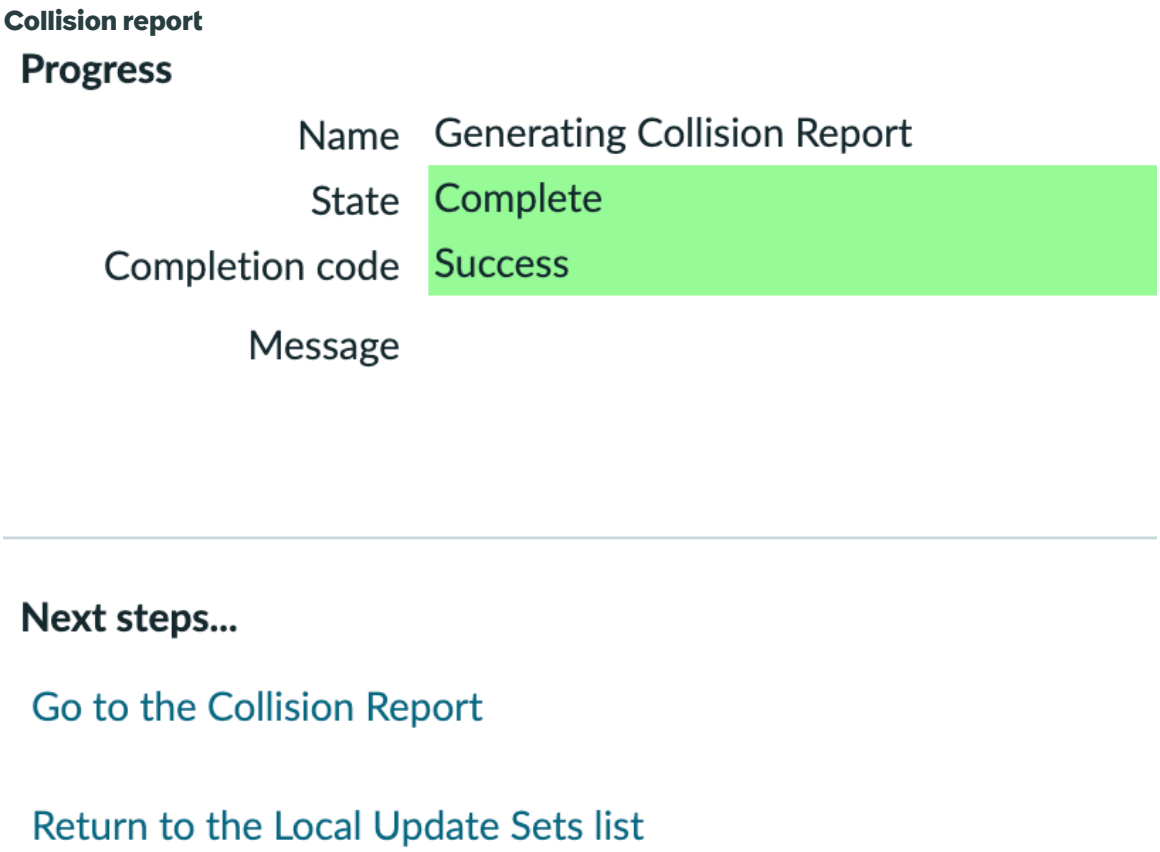
Compare local update sets to identify collisions and confirm that the proper changes are being committed. Resolve all conflicts before moving an update set between instances.

Procedure

1. Navigate to **All** > **System Update Sets** > **Local Update Sets**.
2. Select the check boxes beside the update sets to compare.
3. In the **Actions on selected rows...** dropdown, select **Compare Update Sets**.

   The progress screen displays as ServiceNow generates the collision report.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



4. Select **Go to the Collision Report** when the report is complete.
5. Inspect the list for collisions by locating duplicate collision numbers that show the same change in separate update sets.
6. Resolve the collision by deleting the unwanted update record from one of the update sets.

   Note: To delete a record, open the update record directly. You can't delete it using the **Delete** action from the Update Set Collisions list. Deleting the update record only removes the record of the customization, it doesn't undo the customization.

   a. Select the link in the sys update column for the unwanted update.
   b. Select **Delete**.
7. Run the comparison again to make sure that all collisions have been resolved.

Update set collision resolution

A collision is an update that has a newer local update.

The platform detects collisions by comparing the values in the Name and Updated fields of the customer update record from each update set. If the name matches but there are different update date values, then there's a collision.

When a customer update is moved from one instance to another, it may be rewritten to match the target instance. The rewrite can involve changing the update name of the customer update and one or more sys_id within the update. The rewrites are done when the record or the reference field is for a table that uses a coalesce strategy. This verifies that the customer update will be applied to the correct record. For example, if the sys_dictionary record for tablename.fieldname has sys_id 123 on instance A and sys_id 456 on instance B, when a customer update that refers

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

to that record is retrieved from instance A and recorded in the sys_update_xml table on instance B, references to 123 are updated to read 456.

Coalesce strategies

Coalescing refers to the process of identifying a record based on one or more fields that are used as unique identifiers during the update process. Update sets can detect collisions between identical records that you independently create on separate instances. To detect such collisions, the record must have a coalesce strategy based on coalescing columns. Because collision detection depends on the uniqueness of tables, the tables must be unique when the coalescing columns are combined. Records that aren't listed here won't collide if the same record is created separately on different instances.

Type	Coalescing Columns
atf_input_variable	name, element
atf_output_variable	name, element
dp_data_pattern	source_sys_id
dynamic_attribute	namespaced_name
dynamic_category	namespaced_name
dynamic_category_member	category, attribute
dynamic_choice_override	choice, category, attribute
dynamic_namespace	name
sys_analytics_bucket	sys_scope, bucket_document_id, bucket_table_name
sys_attachment	(uses custom matching logic)
sys_choice_set	name, element
sys_collection	collection, name, join_field
sys_db_object	name
sys_df_data_dictionary	name, element
sys_dictionary	name, element
sys_documentation	name, element, language
sys_index	logical_table_name, col_name_string
sys_module	path
sys_notification_category	name
sys_package	source
sys_package_dependency_m2m	dependency, sys_package
sys_properties	name
sys_report_chart_color	name, element, value
sys_scope_script_access	source_scope, target_scope, script_name
sys_scope_table_access	source_scope, target_scope, table_name
sys_script_validator	internal_type, ui_type

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Type	Coalescing Columns
sys_translated	name, element, value, language
sys_translated_text	tablename, fieldname, documentkey, language
sys_ui_form	name, view, sys_domain
sys_ui_list	name, view, sys_domain, element, relationship, parent
sys_ui_message	key, language, code
sys_ui_related_list	name, view, related_list, sys_domain
sys_ui_section	name, view, caption, sys_domain
sys_ui_view	name
sys_user_group	name
sys_user_role	name
sys_wizard	name
ua_table_licensing_config	name

How customer update record names affect collisions

To understand coalescing, it helps to understand how records that don't coalesce work. For most record types, when a customer update is moved to a new instance, the system doesn't detect collisions for the following reason:

- When you create a record, it receives a unique sys_id. For most record types, the sys_id becomes part of the customer update record name. For example: `sysevent_email_template_9e1998c078b71100a92ecacd80df1d39`.
- Creating an identical record in the same table on another instance produces a customer update record name with a different sys_id. For example: `sysevent_email_template_10b958c8653311005840134572f8e020`

As a result, even though the records might be otherwise identical, the records have different names so the system doesn't detect the collision.

Coalescing records, in contrast, use the following approach to naming records and determining collisions: The following customer update record types use some or all of their coalescing columns instead of the sys_id in their names.

- sys_dictionary
- sys_documentation
- sys_choice_set
- sys_ui_list
- sys_ui_related_list

The resulting identical record name in each instance helps the system to identify collisions even if the records have different sys_ids.

When a customer update is moved from one instance to another, it may be rewritten to match the target instance. The rewrite can involve changing the update name of the customer update and one or more sys_ids within the update. The rewrites are done when the record or the reference field is for a table that uses a coalesce strategy. This confirms that the customer update will be applied to the correct record. For example, if the sys_dictionary record for tablename.fieldname has sys_id "123456789" on instance A and sys_id "987654321" on instance B, when a customer

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

update that refers to that record is retrieved from instance A and recorded in the sys_update_xml table on instance B, references to "123456789" are updated to read "987654321".

Preventing duplicate records

- Transfer data with update sets rather than recreating it on separate instances to verify the records have the same sys_id.
- Export and import records as XML files to verify the records have the same sys_id. See Export and import XML files.
- Enable a unique index for the table from the system dictionary. See Table administration.

Note: The default records included in the baseline system will always have the same sys_id because the instance imports the records as XML files during instance provisioning.

Preview a remote update set

Previewing checks a retrieved update set against the local instance to find issues. Resolve all problems before committing the update set.

Before you begin

Role required: admin

If the system property glide.update_set.auto_preview is set to `true`, the system automatically starts the preview process after the update set is retrieved. If this property is `false`, the preview process must be started manually.

Procedure

1. Navigate to **System Update Sets** > **Retrieved Update Sets**
2. Select **Preview Update Set**.
   For large update sets, the preview process may require a significant amount of time. If necessary, you can cancel the preview process by selecting the **Cancel** button on the progress dialog box.
   The Update Set Preview page shows results and lists problems. Read the information and select **Close**.
3. **Optional:** Preview the records.
   a. Open the update set record and select **Show All Preview Records** to make sure that the correct updates are being committed.
   b. Open the update set record and select **Run Preview Again** to run the comparisons again.
   c. Review the **Update Set Preview Problems** related list to verify that the correct updates are being committed.
4. On the Retrieved Update Set form, select **Commit Update Set** to commit all changes on the instance without reviewing the preview results.
   If problems were detected, address each problem in the Update Set Preview Problems related list.

Review a preview record for an update set

The process of previewing an update set creates a preview record for each update. You can review the preview records to make sure that the correct updates are being committed.

Before you begin

Role required: admin

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Procedure

1. Open the Update Set record and preview the update set.
2. Select the **Show All Preview Records** related link.
3. Select the **Disposition** to open a preview record.
4. Fill in the form.

Field	Description
Disposition	Indicates when a collision is detected. If the change predates a local modification to the same object. ○ Collision ○ Update ○ Insert ○ Delete
File differences	Compares the latest local object version with the update set, marking differences with colors. ○ Additions are marked green. ○ Changes are marked yellow. ○ Deletions are marked red.
Proposed action	Specifies the procedure for managing the change once the update set has been committed. **Commit** Accept the change in the remote update. The default proposed action for every preview record is to commit, even if a newer update exists on the instance. **Skip** Reject the change.

5. **Optional:** Resolve any problems listed in the **Update Problems** related list.
6. In the proposed action field, select the action to take when committing the update set.
7. Select **Update** to save the action.

Resolve a preview problem in an update set

Preview an update set to detect and resolve problems that may occur if you commit the updates on the local instance.

Before you begin

Role required: admin

Procedure

1. Navigate to **All** > **System Update Sets** > **Retrieved Update Sets**.
2. Open the update set record and scroll to the **Update Set Preview Problems** related list.
3. Review each problem description to determine the cause and resolve the problem.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Problem	Description
Missing Object	The object or a referenced object doesn't exist on the target instance. Create another update set on the source instance to transfer the missing object to the local instance, or create the object on the local instance. ○ Open a new window with the **Find missing field** or **Find missing record** and search the source instance for the missing field or record. ○ Open a new window with **Find missing update** and search the source instance for the update record that corresponds to the missing field or record.
Collision	A change in the update set is older than a change to the same object on the local instance. Compare the two updates and determine which version to use. ○ To use the version on the local instance, select **Skip remote update**. ○ To use the version in the update set, select **Accept remote update** ○ To use the version in the update set, select Accept remote update. ○ Open the preview record with **Compare with local** to view a comparison of the differences between the local version and the version in the update set. ○ Show local field ○ Show local record ○ Show local update
Uncommitted update	The object exists in another remote update set that hasn't been committed. Commit the other remote update set first or move this update to the other update set. ○ **Show uncommitted update** opens the update record in the other remote update set. ○ **Show uncommitted Update Set** opens the other remote update set record.
Table to be deleted has data	Deleting a table removes its data, unlike other metadata deletions. Skip or ignore related issues before committing updates. Restoring the table doesn't recover its data.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Problem	Description
	Note: You can't delete system tables or tables outside of your application scope.
Application scope validation issue	The previewer identifies the following combination of states as a problem: ○ The scope for the update set isn't Global. ○ The application isn't found on the target instance. ○ The application isn't included with the update set. ○ The application isn't found on the ServiceNow Store. Transfer the update set only to instances that include the application scope or confirm that the update set includes the application.
Conflict within a single batch	Two or more update sets within the same batch have conflicting changes. The **Update Set Preview Problems** list contains a record for each update set with a conflicting change. ○ Compare the conflicting update sets and determine which version to use. ○ Select **Compare Collisions** to compare the conflicting update sets. ○ Select **Accept this collision**.

Commit an update set

After resolving any issues from previewing, commit the update set to apply all changes and generate a local copy with records of each update.

Before you begin

Role required: admin

Procedure

1. Navigate to **All** > **System Update Sets** > **Retrieved Update Sets** and open the update set.
2. **Note:** You can't commit an update set until all problems are resolved.

   Resolve any problems with the update set.
3. Select **Commit Update Set**.
   a. Select **Cancel** to return to the preview and reevaluate the change.
   b. Select **OK** to skip the change and continue committing the changes that are marked as **Commit**.

   If the update set contains one or more deletes, the system displays a warning. The warning lists up to five updates that may contain problems. If more than five updates have potential problems, the system provides a link.
4. When the system successfully commits an update set, it displays a completion page.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

5. Select **Commit log** on the confirmation page.

   a. **Optional:** Navigate to **System Update Sets** > **Update log** and filter for the update set name.

   b. Look for unsafe edit warnings.
   The system helps prevent changes that could cause data loss, like changing a field type with existing data. Make changes manually if needed, and use caution when updating production data. Look for errors that indicate which records failed to commit and why. Create an update set to address those failures, if necessary.

6. Navigate to **System Update Sets** > **Local Update Sets** and open the local update set record to review the changes.
7. Change the state to **Ignore** when you're no longer working on the update set, but don't want it transferred to another instance.

Retrieve an update set

Retrieve completed update sets from another instance.

Before you begin

Role required: admin or teamdev_user

Warning: Update sets let you transfer changes across instances with different family releases and features. You can load update sets from older releases into newer ones, but loading updates from newer releases into older instances may cause errors or data loss and should be avoided if possible. The same cautions apply when moving updates between different versions of store apps.

Procedure

1. If IP address access control is enabled on the source instance, set up the target instance as an exception.
2. On the target instance, navigate to **System Update Sets** > **Update Sources** and select **New**.
3. Specify the connection settings as described in the table.

Field	Description
Name	Enter a unique name for the instance.
Type	Specify whether the remote instance is a development, test, or UAT instance.
Active	Specify whether the local instance can transfer update sets to the remote instance. You can transfer update sets only to active remote instances.
URL	Specify the URL of the remote instance using the appropriate transfer protocol. Each remote instance record should have a unique URL. Creating duplicate records with the same URL can cause errors. The remote instance must be on the same release family as the local instance. **Note:** You can't change the URL after the system verifies the connection. Use the *Active* field to deactivate unwanted remote instances.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Username	Enter the user on the remote instance who authorizes transferring update sets to the instance. This user account must have the admin user role on the remote instance.
Password	Enter the password of the authorizing user.
Short description	Enter any other relevant information about the remote instance.

4. Select **Test Connection**.
   If the connection is successful, a confirmation message appears. If the connection fails, a warning message identifies the cause of the failure.
5. **Optional:** If the connection fails, modify the settings to establish connectivity.

   (Optional) Connectivity must be established prior to saving the connection settings. Additionally, it may be necessary to modify the source instance configuration, such as updating the password.
6. Select and hold (or right-click) the form header and select **Save**.
7. Under **Related Links**, select **Retrieve Completed Update Sets**.
   Update sets marked as completed are moved from the source to the target instance, while existing update sets on the target are skipped. The confirmation page details how many sets were transferred and skipped.
8. To view retrieved update set, navigate to **System Update Sets** > **Retrieved Update Sets**.

Back out an update set

You can back out changes to existing records for any committed update set.

Before you begin

Role required: admin

About this task

Backing out an update set adds delete updates to your current set. If you commit, back out, and reapply a remote update set, preview errors occur due to recent deletions causing collisions. The back-out process reverses both record-level updates and changes to the dictionary. Some changes caused by a back-out can result in data loss.

Warning: Don't back out the **Default** update set. This action can damage the configuration of the instance. Backing out an update set that belongs to an update set batch backs out the selected update set, plus any children of the backed-out update set.

Procedure

1. Navigate to **All** > **System Update Sets** > **Local Update Sets**.
2. Open the update set record to back out.
   Make sure it's complete and a date is set on the **Release date** field.

   Note: The application you select determines the available update set options. Ensure your chosen application matches the update set contents, for example Global.

3. Review the update set and determine if backing it out can cause issues.

   If backing out an update set can cause problems, create and commit a new update set to undo the changes instead.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

4. Select **Back Out**.
   The page displays actions, progress, and problems. Problems are changes in more recent update sets that affect the update set that is being backed out.
5. Resolve each problem before proceeding with the back out.
   The back-out preview process generates a warning for each problem.
6. To keep the latest version, select **Decide to Keep Current**.
7. To back out to the previous version, select **Decide to Use Previous**.
   The update set and all associated update records are deleted. If needed, you can still navigate to the retrieved update set, preview it, and commit it again.

Mark an update set complete

When you have completed the customizations and compared local update sets to resolve conflicts, mark the update set as *Complete*.

Before you begin
Role required: admin

About this task
Mark an update set as *Complete* only when it is ready to transfer. Once an update set is complete, do not change it back to *In progress*. Instead create another update set for the rest of the changes, and commit them in the order that they were created.

Procedure

1. Open the update set record.
2. Change the *State* of the update set from *In progress* to *Complete*.
   The update set is available for other instances to retrieve and additional customizations aren't tracked in the update set.

Working with batched update sets

Batched update sets enable you to group update sets together so you can preview and commit them in bulk.

Dealing with multiple update sets can lead to problems, including committing update sets in the wrong order or inadvertently leaving out one or more sets. You can avoid these problems by grouping completed update sets into a batch.

Note: To preserve the update set batch hierarchy while cloning, set only the parent update set to Ignore status and leave all other update sets as Complete. This will prevent batch sets from cloning and preserves the batch update set hierarchy.

The system organizes update set batches into a hierarchy. One update set can act as the parent for multiple child update sets. A given set can be both a child and parent, enabling multiple-level hierarchies. One update set at the top level of the hierarchy acts as the base update set.

When you preview or commit the base update set, you preview or commit the entire batch. The system decides which changes to process first and checks for any conflicts by looking at when each change was made and how they are connected in a sequence. This sequence, or ancestry, refers to the specific moments when each change in the update sets occurred.

Note: For more details, see Compare local update sets and View customizations and compare with current version.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Example of batched update sets

The list of update set records reflects the batch hierarchy in the **Parent** and**Batch Base** columns.

List of batched update sets

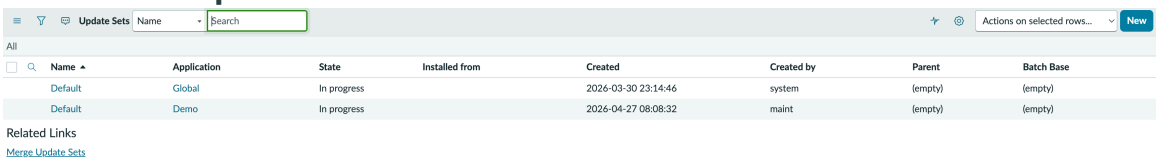
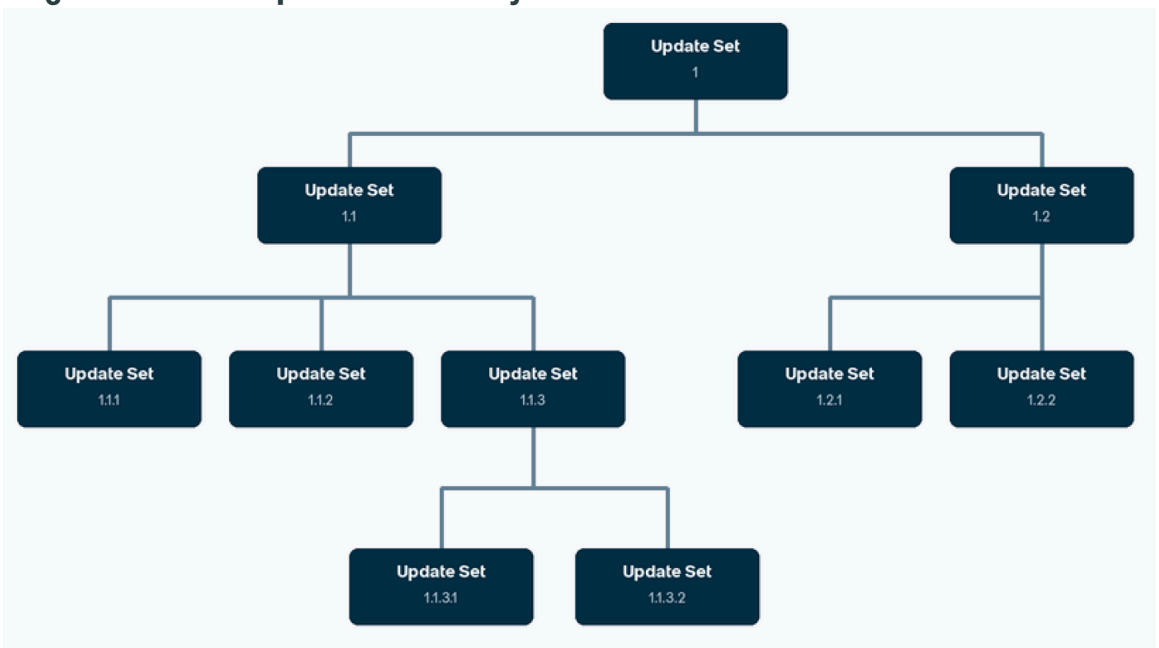


Diagram of batched update set hierarchy



Create a batch update set

You include an update set in a batch by specifying another update set as its parent.

Before you begin

Role required: admin

When you add an in-progress update to a batch that is marked as complete, the status of the batch changes back to in progress.

Procedure

1. Navigate to **All** > **System Update Sets** > **Local Update Sets**.
2. Select the record for an update set that you want to include as a child in the batch.
3. Navigate to the parent field on the update set record.
4. Select the update set to act as the parent.
5. Select **Update**.
   The system navigates back to the Update Sets list. When the parent column is displayed, it identifies the parent associated with the newly created child.

Preview a batch of update sets

You can preview at once all the update sets belonging to a batch.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

Role required: admin

You must have retrieved the update sets from the source instance.

Procedure

1. Navigate to **All** > **System Update Sets** > **Retrieved Update Sets**
2. From the list of retrieved update sets, select the batch base for the batch you want to preview. You can't separately preview an update set that is a child in a batch. You must preview the entire batch by previewing the batch base. If necessary, you can remove the child update set from the batch by editing its record's **parent** field.
3. Select **Preview Update Set Batch**.
4. If the system found problems, preview the problems.
   a. Select the **Preview Problems for Batch** and resolve the problems as you normally would for any update set.
   b. When you have resolved all the problems, select **Run Preview Again for Batch**.

Commit a batch of update sets

You can commit at once all the update sets belonging to a batch.

Before you begin

Role required: admin

Before committing, you must have previewed the update sets from the source instance and resolved any collisions.

Procedure

1. Navigate to **All** > **System Update Sets** > **Retrieved Update Sets**
2. From the list of retrieved update sets, select the batch base for the batch you want to preview. You cannot separately commit an update set that is a child in a batch. You must commit the entire batch by committing the batch base. If necessary, you can remove the child update set from the batch by editing its record's **parent** field.
3. Click **Commit All Update Sets**.

Reorganize a batch of update sets

You can remove an individual update set from the batch or change its parent.

Before you begin

Role required: admin

Procedure

1. Navigate to **All** > **System Update Sets** > **Local Update Sets**.
2. Select the record for an update set that you want to move or remove as a child in the batch.
3. On the update set record, navigate to the parent field and select the new update set to act as the parent.
4. To remove the update set from the batch, delete any text from the parent field and leave it empty.
5. Select **Update**.
   The system returns to the list of updated sets. If the batch base column is visible, it shows the parent for the newly created child.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Update sets reference

Reference topics such as general guidelines and system properties for system update sets.

Default update set

Only one update set can be the default set for any application scope.

Global default set

Use the global default update set to change an instance without adding the changes to any user-created update sets. The global default update set is the set where the default set is `true` and the application scope is global. The global default set provides system functionality and shouldn't be changed, deleted, or moved between systems. Use the global update set to change an instance without adding the changes to any user-created update sets.

Note: The system sets the default set to `false` for all other update sets with the same scope. This ensures that there's only one default update set for each scope.

Auto-generated default set

Common cases where the system auto-generates a default update set are as follows.

- The first time that an admin logs in, the system sets the system's global default update set as the administrator's update set. In addition, the application picker sets the administrator's application scope to global.

  If a global default update set is marked **Ignored** or **Completed**, the system creates an update set for the global application scope and performs the following actions:

  - The system sets the default set to `true` for the new set.
  - The system sets the name of the new set to start with the name of the former default set and appends the next numeral.
  - The system sets the newly created set as the administrator's update set.
- When a user marks the default set for a scope as **Ignored** or **Completed**, the system immediately auto-generates a new default set for the scope.
- The system auto-generates a new default update set for a scope when all the following conditions occur:
  - You change the application scope.
  - Your preferred update set is **Complete** or **Ignored**.
  - There's no In-Progress default update set for the new scope.

Customizations tracked by update sets

Update sets track customizations to application tables, fields, and records.

Update sets capture configuration details but exclude task or process data. For example, they record catalog item definitions, variables, and variable choices, but not submitted orders or catalog tasks.

update_synch attribute

To see the list of tables where customizations are tracked, navigate to **System Definition** > **Dictionary** and filter on attributes contains update_synch.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Warning: Don't add the update_synch attribute to a dictionary record. When improperly used, this attribute can cause major performance issues or cause the instance to become unavailable. The update_synch attribute isn't accessible to customers.

A default rule blocks the use of the update_synch attribute on a table for which it isn't predefined to avoid the following issues:

- Some core tables require special update handling because they represent information on multiple tables. When the update_synch attribute is added to these tables, duplicate update records are created, causing major conflicts that are difficult to troubleshoot and repair.
- Using the update_synch attribute to migrate data records between instances can cause performance issues. To migrate data see Import sets key concepts to use an instance-to-instance import.

Special handlers

Special handlers represent information on multiple tables. These changes are packaged into one update set entry so that all records are properly updated when the customization is committed.

- Workflows
- Form sections
- Lists
- Related lists
- Choice lists
- System dictionary entries
- Field labels

Warning: Form sections, lists, related lists, choice lists, and field label handlers can delete and reinsert records, which may lead to unexpected results or data loss if fields reference those tables.

Choice lists

Update sets store both new and updated choice options as separate records in the sys_update_version and sys_update_xml tables. For example, if you extend the Task table with a u_activity table, add a state field choice visible only in your extended table.

When you publish these changes as an update set, the update only contains update and version records for the choice you added to the u_activity table. The choice options in the task table are unaffected.

Warning: Don't use large choice lists in update sets. Doing so leads to excessively long update set commits.

Dictionary changes

Using update sets help to prevent you from applying dictionary changes that result in data loss. For example, blocked dictionary changes include removing tables or changing a column data type.

Update sets don't track the removal of tables from the system dictionary. You must manually remove tables from the target instance. While update sets track data type changes, the target instance skips any change that results in data loss and instead adds a log message about the action. You can use the log to make data type changes on the target instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: Update set previews don't detect type mismatches when changes could cause data loss. If you delete a table column via update sets, any existing data in that column is lost when the update is committed. A warning appears noting which deletions result in data loss.

Home pages and content pages

Home pages and content pages aren’t added to update sets by default. Add pages to the current update set by unloading them.

Important:

The functionality found in homepages, arranging information from your instance to tell a story about your data, is found in dashboards on new instances. On upgraded instances with Next Experience enabled, users can view existing homepages if they have a direct URL, but they can't create or edit them. Responsive dashboards and Analytics Overview dashboards take over homepage functionality.

Use the Homepage deprecation help tool to convert the homepages on your instance to responsive dashboards.

For more information, see:

- Dashboards in the Analytics Center.
- Working with responsive dashboards.

Application changes

The system creates a separate update set for each application that only contains changes associated with the application. This separation verifies that access settings for each application are properly evaluated and applied when committing update set changes.

General guidelines for planning the update process

A reference topic that contains general guidelines to create a standard process for moving customizations from instance to instance.

- Use a standard path for update sets: migrate from development to test, then from test to production. Avoid moving the same update set from more than one source.
- Check that both instances are on the same version. While it is possible to move an update set from an instance on an older version to an instance on a newer version, it increases the likelihood of problems. Customizations may not work if they rely on code that has changed between versions.
- Change a single update set for each small or medium task. Large update sets with schema changes or major workflow revisions are tougher to review, slower to process, more conflict-prone, and take longer to preview or commit.
- Ensure that all base system records have consistent sys_id fields. Certain base system records may be generated on an instance post-provisioning, resulting in differences between instances and potential complications with update sets.
- Schedule update set commits outside business hours to avoid slowing down the production instance. Any reduced performance is brief.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- Use clear update set names and establish a naming convention to coordinate developer changes and streamline referencing during commits.
  - If update sets are being generated as fixes for problems, consider including the problem ticket in the name. For example **PR10005 - Duplicate Email Issues Fix**.
  - If you need more than one update set to address a problem, include a sequence number in the naming convention. Sequenced naming conventions confirm that update sets are applied in the order that they were created. For example, **PR10005 - Duplicate Email Issues Fix** and **PR10005.2 - Duplicate Email Issues Fix**.

Update sets properties

A reference topic that contains information on the system properties used with update sets.

To open the System Properties [sys_properties] table, enter `sys_properties.list` in the navigation filter.

Properties for Update sets

Property	Description
glide.update_set.auto_preview	Automatically preview retrieved update sets: By default, the system automatically previews update sets as soon as it has retrieved them. • Type: true I false • Default value: true • Other possible values: false • Location: System Property [sys_properties] table
glide.ui.update_set_picker.role	Grant access to the update set picker: By default, only administrators can use the update set picker. You can grant access to additional users. • Type: true I false • Default value: true • Other possible values: false • Location: System Property [sys_properties] table

Important considerations before deleting update sets

Revert a customization, back out the update set rather than deleting it.

Administrators can delete an update set only when it isn't the current update set and contains no sys_update_xml entries. For example, after merging update sets, you might want to remove the original sets. This capability is governed by an access control rule (ACL) applied to the sys_update_set table.

Don't delete sys_update_xml entries, because this action:

- Doesn't undo the updates.
- Removes any record of who applied the customizations.
- Removes the sys_update_xml entries associated with the update set, so customizations are overwritten when the instance is upgraded.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

When you try to delete an update entry, a warning message appears. Select **OK** to confirm the deletion.

Administer your apps

The system offers several ways to manage applications. You must have the admin role to perform these procedures.

Application Manager

A new Application Manager experience is now available. The new experience simplifies the process of managing your licensed applications by consolidating them in one convenient location, displaying all your licensed applications, plugins, installed applications, and available updates. For more information, see Application Manager.

Application sharing

Administrators can share applications that are complete and are ready for use on other instances by publishing to the application repository, publishing to the ServiceNow Store, publishing to an Update Set, or pushing to team development instances. For more information, see Application sharing.

Application sharing

Administrators can share applications that are complete and are ready for use on other instances.

Application developers can share applications using one of the following methods.

Application sharing methods

Sharing method	Makes available to	Typical use case
Publish to the application repository	All instances assigned to the same company	Transfer an application to a test or production environment.
Publish to the ServiceNow Store	All ServiceNow customers	Share or sell applications to other companies.
Publish to an Update Set	Any instance with access to the Update Set file	Save a version of an application for compliance or backup reasons.
Push to team development instances	Other instances in the team development environment	Push developer changes to the parent instance.

Note:

"Tracking schema": Deleting a table or a column in a scoped application is enabled by default for freshly zBooted instances. This is done by having the system property `com.glide.apps.include_my_schema` set to "true".

For upgraded instances, if you have no custom applications installed or in development, "tracking schema deletes" is enabled by default. Otherwise the property is set to "false" so that customers get the same experience for schema deletes in their applications as in previous releases before Paris. To learn more see the New York and Scoped Applications New Features article on the ServiceNow Community site.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Custom licensing for ISV applications

For applications that you are sharing, you can create a definition to track usage metrics on your application. For more information, see Custom licensing for ISV applications.

ServiceNow application repository

After you develop and test a custom application, you can make the application available to company instances by publishing it to the ServiceNow application repository.

The ServiceNow application repository is a central repository for all scoped applications that are published by all ServiceNow customers.

Note: The ServiceNow application repository isn't available for self-hosted (on premise) customers.

The application repository enables ServiceNow customers to upload and distribute applications between their instances. When you access the application repository, you can see and manage only the applications that are published by your own organization. You can't see or manage applications that are published by other organizations.

After you have designed, developed, and successfully tested a custom application, you can publish your application to the ServiceNow application repository to share it to other instances in your company.

Entitlements

An entitlement refers to permission given to an instance to install a scoped application from the application repository. An instance must be entitled to an application in order for you to be able to install the application on the instance.

By default, after you publish an application to the application repository, all your company instances are entitled to the application automatically. To limit which company instances are entitled to the application, access the application repository by going to https://apprepo.service-now.com/sn_appstore_store.do, and then change the entitlement type for the application. You can also entitle an instance again if the application entitlement has already been removed. For more information, see Manage application entitlements from the application repository.

Using the application repository

You can access the application repository by going to https://apprepo.service-now.com.

Note: Customers using the Federal Store must log in to HIWave and then select the Federal Store option at the bottom of the page. Then, under your company name, select My Repository to proceed.

After you publish an application to the application repository, you can:

- Install an application from the application repository
- Manage application entitlements from the application repository
- Delete an application from the application repository
- Release a scope from the application repository

Manage applications

Learn how to manage the applications you publish to the ServiceNow application repository.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

After you have designed, developed, and successfully tested a custom application, you can publish your application to the ServiceNow application repository to share it to other instances in your company. You can see, manage and customize only the applications that are published by your own organization or scoped applications that are installed via plugins using the ServiceNow Application Repository. You can't see or manage, or customize applications that are published by other organizations.

Note: Do not combine the usage of both Update Sets and the Application Repository for scoped app development. This will result in numerous issues, including skipped changes, commit errors, and more. Once you have installed an application from the Application Repository, you must continue to develop and publish to the Application Repository for all future development. If you decide to develop an application using update sets, you must continue to use that method exclusively.

Overview of managing applications

The Application Repository is part of the ServiceNow Store. ServiceNow customer instances are connected to a particular Store instance (for example, Commercial or Regulated Market). Customer applications published to the Application Repository are stored in a multi-tenant artifact storage, but the storage is designed so that customers cannot see or access other customers' published applications.

Application Repository has a limit of 20 versions of a given application. If your team is generating more builds or versions of your application than 20 at a time, the oldest is deleted.

Application Repository has a file size limit for applications at 1GB. Applications over this file size are not accepted during the publishing process.

Manage application entitlements from the application repository

Add or remove application entitlements to limit which instances the application can be installed on.

Before you begin

You can manage only the applications that you've published to the application repository. For more information, see Publish an application to the application repository.

Role required: none

About this task

By default, after you publish an application to the application repository, all your company instances are entitled to the application automatically. To limit which company instances are entitled to the application, access the application repository by going to https://apprepo.service-now.com/sn_appstore_store.do, and then change the entitlement type for the application. You can also entitle an instance again if the application entitlement has already been removed.

Procedure

1. Go to https://apprepo.service-now.com/sn_appstore_store.do.
2. Log in using your Now Support credentials.

Note: Customers using the Federal Store must log in to HIWave and then select the Federal Store option at the bottom of the page. Then, under your company name, select My Repository to proceed.

3. Next to the application listing, click **Select Action** and then click **Manage Entitlements**.
4. Choose an entitlement type for your application.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Option	Description
Remove all existing entitlements	None of your company instances can install the application.
Entitle all instances	Any of your company instances can install the application. This option is selected by de fault when you publish an application to the application repository.
Entitle selected instances	Only instances that you select can install the application. Pick which instances to entitle by moving instances from the **Available In stances** list to the **Selected Instances** list.

5. Select **OK**.

Release a scope from the application repository

Release a scope from the application repository so that the scope can be used to create new scoped applications.

Before you begin

You can release an application scope only if the scope isn't being used by any application. If your scope is being used by an application, follow the steps in Delete an application from the application repository before releasing the scope.

Role required: Primary customer admin of the account

About this task

The application repository stores the scopes of all your custom applications. You cannot create a new application using a scope that is being stored in the application repository. To release a scope means to remove the scope from the application repository so that you can create new applications using the scope.

Procedure

1. Go to https://apprepo.service-now.com/sn_appstore_store.do.
2. Log in using your Now Support credentials.

> **Note:** Customers using the Federal Store must log in to HIWave and then select the Federal Store option at the bottom of the page. Then, under your company name, select My Repository to proceed.

3. Open the Scopes tab.
4. Next to the scope listing, click the trash icon ( ).

Publish an application to the application repository

Publish a custom application to the application repository so that it can be installed on other instances in your organization.

Before you begin

To enable a developer to publish an application to the application repository, delegate the Publish to App Repo permission to the developer. For more information, see Delegate development and deployment permissions to personnel.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

If you use an on-prem (self-hosted) instance, see additional details at Application Repository for self-hosted, air-gapped customers.

Role required: admin or delegated_developer with Publish To App Repo permission enabled

Procedure

1. Navigate to **All** > **System Applications** > **My Company Applications**.
2. Open the In Development tab.
3. Open the application record that you want to publish to the application repository.
4. Select the **Publish to My Application Repository** related link.
5. Select **Submit**.

What to do next

Install the application on company instances so that your organization can start using it. For more information, see Install an application from the application repository.

By default, after you publish an application to the application repository, all your company instances are entitled to the application automatically. To limit which company instances are entitled to the application, access the application repository by going to https://apprepo.service-now.com/sn_appstore_store.do, and then change the entitlement type for the application.

Note:

The **Can Edit Application in Studio** property defaults to true for new applications, but you can set it false before publishing.

For more information, see Manage application entitlements from the application repository and Access ServiceNow Studio.

Install an application from the application repository

Install your application on an instance so that employees can start using the application that you developed.

Before you begin

- Publish an application to the application repository.
- Check the entitlement type of the application to ensure that your instance is entitled to the application. For more information, see Manage application entitlements from the application repository.

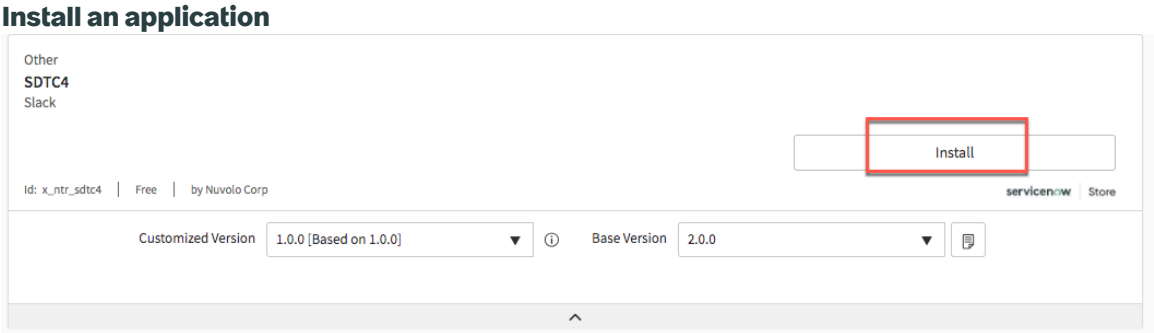
Role required: admin

Procedure

1. Navigate to **All** > **System Applications** > **My Company Applications**.
2. Find the application.
3. Next to the application listing, select a version to install.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

4. Click **Install**.



Delete an application from the application repository

Delete an application from the application repository so that it's no longer available to your company instances.

Before you begin

- Publish an application to the application repository
- You can delete an application only if the application is not installed on any of your company instances. Uninstall the application on all your instances before deleting it from the application repository.

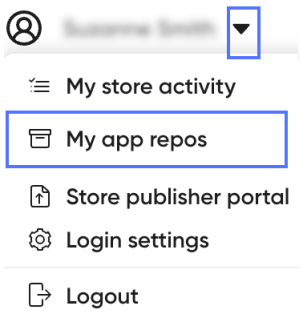
Role required: Primary customer admin of the account

Procedure

1. Go to https://apprepo.service-now.com/sn_appstore_store.do.
2. Log in using your Now Support credentials.

> **Note:** Customers using the Federal Store must log in to HIWave and then select the Federal Store option at the bottom of the page. Then, under your company name, select My Repository to proceed.

3. Select the expand arrow next to your name and select **My app repos**.



4. Next to the application listing, click **Select Action** and then click **Flag for Deletion**.
5. On the confirmation window, click **Yes**.
   After you flag an application for deletion, the application is deleted automatically after 90 days.
6. To delete the application immediately:
   a. Open the Flagged Apps tab.
   b. Next to the application listing, click **Select Action** and then click **Delete Immediately**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Manage customizations to applications

You can manage your company's customizations for applications that belong to other organizations or a scoped ServiceNow plug-in.

Before you begin

Role required: admin

The entitlements of your company's application-customizations are controlled solely by the entitlements of the respective store or vendor-released application. There are no separate entitlements for application-customizations, so you can't customize instances that don't have an entitlement to a base store application.

The development instance that you are using to create application-customizations should have its respective store or plugin application entitled and installed.

Note: ISV partners who want to customize their own applications must first publish to the ServiceNow Store and install on an instance before they can use application-customizations via the Application Repository. This is the case even though all of the records are in the same scope. Installing on the instance enables ISV partners to publish a different set of capabilities for ServiceNow Store applications versus extensions used internally for their own use.

About this task

You cannot change the actual applications that belong to other organizations (purchased from the Store) or the global plugins provided in the ServiceNow base system. But you can create and manage your company's customizations for store applications or scoped applications that are installed via plugins, using the ServiceNow Application Repository.

There are two ways to create application-customizations.

- Create from ServiceNow Studio
- Create from the navigation pane

Important: An App Customization package is intended to be authoritative and a full representation of the desired changes or customizations of the application that you're building on top of. Local customizations (typically tied to the sys_update_xml) are not honored because of this. If a specific field must be persisted (for example a MID Server) – consider using the Loader exempt attribute, see Altering tables and fields using dictionary attributes. If a specific property must be set – consider making it 'private', see Available system properties.

Procedure

1. Use these steps to create application-customizations from ServiceNow Studio.

   a. Open the Store application or the application installed via plugin from ServiceNow Studio.
   To learn more, see Access ServiceNow Studio.
   The application opens in Studio.

   b. Make application changes from Studio.

2. Use these steps to create application-customizations from the navigation pane.

   a. Navigate to **System Applications >All**.

   b. Search for the installed application you want to customize.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note: You can open the application in either Studio or select it as the current working application in the Store application record.

c. To open this application in Studio, select the stack menu, and then select **Edit in Studio**.
The application opens in Studio. You can make application changes from Studio.

d. To edit the Store application record, select the name of the application.

e. In the Related Links section of the Store application record, select **Switch to this Application**.
The application becomes the current working application in the Application picker. You can make application changes from the Store application record.

Related topics

Manage application entitlements from the application repository

Publish customizations to an application repository

After you have designed, developed, and successfully tested your customizations to a store application or a scoped plugin, you can publish it to the ServiceNow application repository to share it to other instances in your company.

Before you begin

Role required: admin, or delegated_developer with Publish To App Repo permission enabled

Only one customization can be created per vendor or customer code registered on the store and app repo.

To enable a developer to publish an application to the application repository, delegate the Publish to App Repo permission to the developer. For more information, see Delegate development and deployment permissions to personnel.

Procedure

1. Navigate to **System Applications** > **All Available Applications** > **All**.
2. Find the application by typing its name in the Filter Navigation box.



3. Open the application record for which you want to publish the customization to the application repository.
4. Click the **Publish Customizations to My App Repo** related link or in ServiceNow Studio, use the **File Publish** menu item.
5. Click **Submit**.

Convert custom applications to upgrade from the application repository

When your applications are placed in the Custom Applications table [sys_app], you can't upgrade them directly through the Application Repository. This procedure helps you do a one-time conversion when you want to migrate deploying your applications using the Application Repository.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

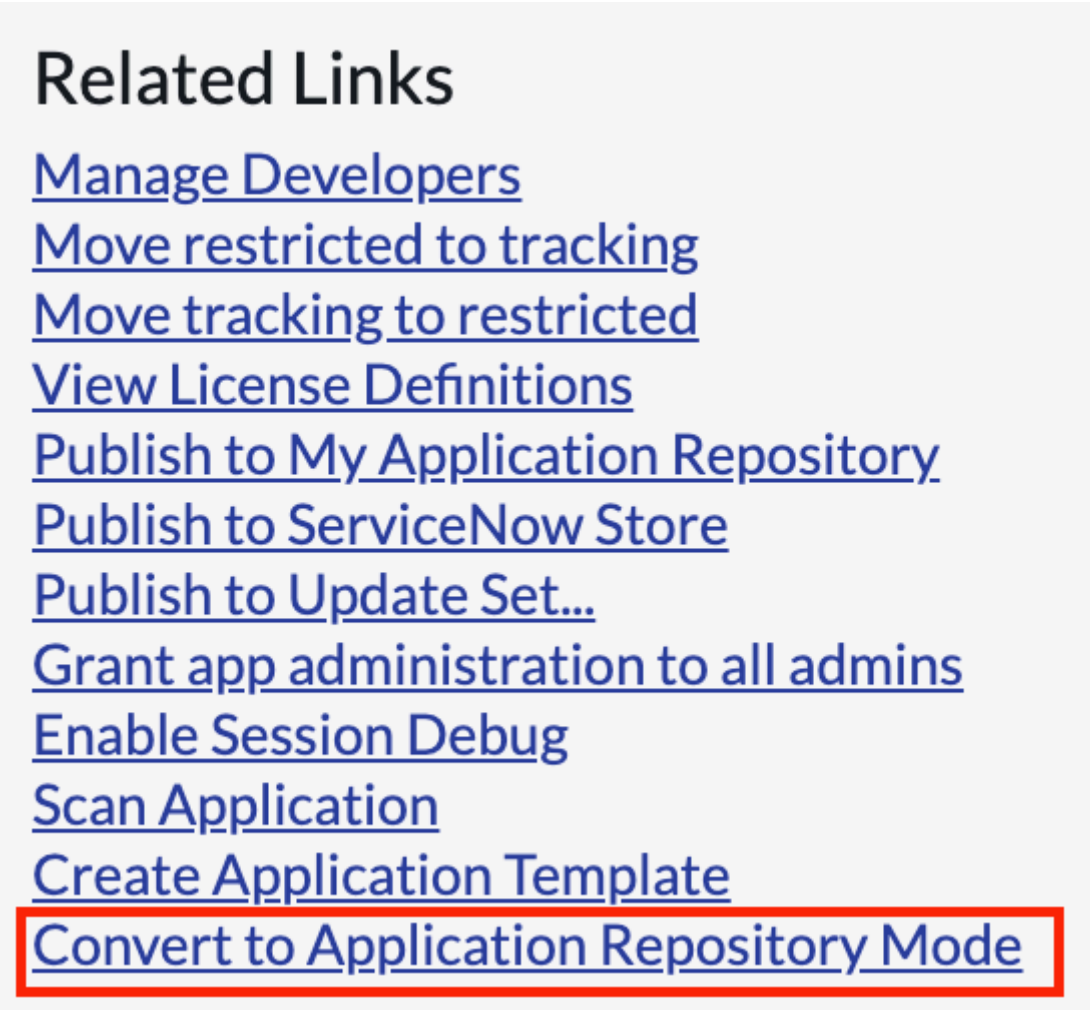
Role required: admin

When you start with Update Sets, all your applications in all your instances are placed in the Custom Applications table [sys_app]. However, if you decide to use the recommended ServiceNow® Continuous Integration/Continuous Delivery API/Spoke guidance for using the Application Repository in your CI/CD pipelines, the system doesn't allow the upgrades because the applications already exist in Custom Applications table. They are moving to the ServiceNow Store table [sys_store_app] so they can be installed from the application repository, which is why they can no longer be developed on that instance.

After you convert the application, it is no longer enabled for development on the instance where the conversion is performed. For a scoped custom application, all associated records in the Customer Updates table are deleted for this application.

Procedure

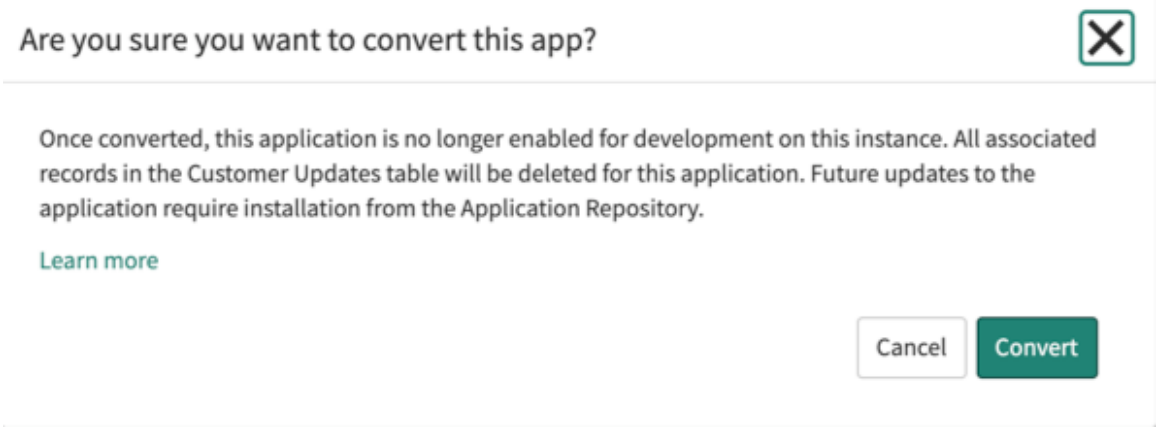
1. Navigate to **All** > **System Applications** > **My Company Applications**.
2. Open the **In Development** tab.
3. Click the name of the application record that you want to convert.
4. Select the **Convert to Application Repository Mode** related link.



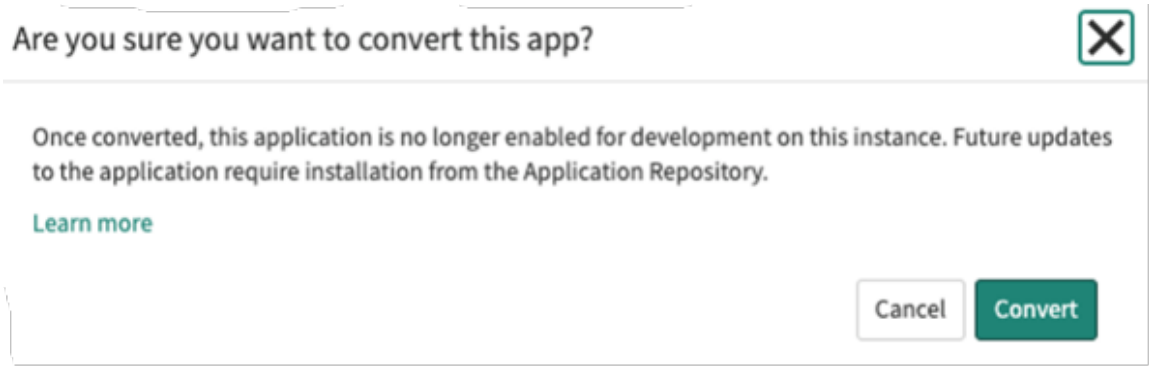
5. Select **Convert**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

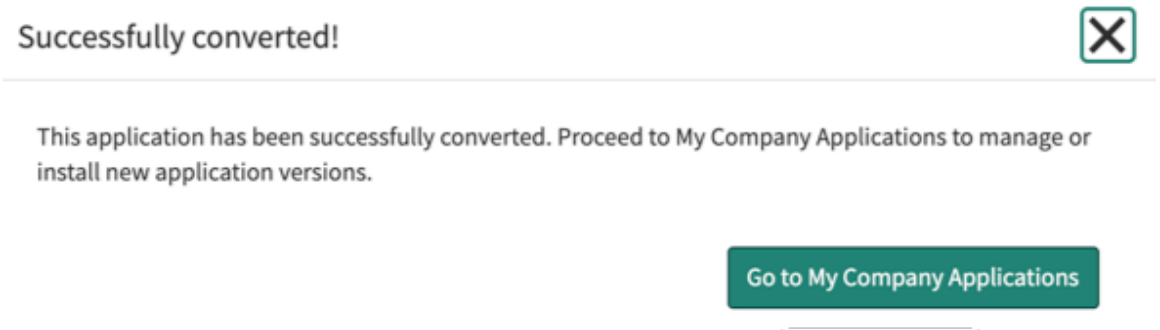
- In scoped custom applications, the following message displays.



- In global custom applications, the following message displays.



- After the successful conversion, the following message displays.



What to do next

Go to **System Applications->My Company Applications** and update the application using the software from the application repository. See Install an application from the application repository to learn more.

Convert your installed applications to development mode

Convert the installed applications that your company owns to development mode after you install an application onto a non-production instance to use for development or clone a production instance into a non-production instance for development. With this conversion, you enable newer versions of the application to be created, committed, and published.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin

You must set the instance scope to the desired application scope. For example, if converting Agent Workspace, the application scope is set to Agent Workspace.

Role required: admin

About this task

The system properties feature [sn_appclient.store_app_convert_enabled] controls this action, which is enabled by default. You can disable this feature by setting sn_appclient.store_app_convert_enabled to false.

You can convert only the applications that your company owns. The application scope must match a key that is set in sn_appauthor.all_company_keys.

Note: Installed global applications are eligible for conversion.

After you convert the application, it can't receive updates from the application repository on this instance. Ensure that the version of the application that is installed on the instance is the version that you intend to develop as a new version.

Note: Applications that are installed with demo data lose their reference to this data on installation. If an application is converted, you must re-create any sys_metadata_link records to the demo data before you link the application to the source control or publish a new version.

Procedure

1. Navigate to **All** > **System Applications** > **My Company Applications**.
2. In the **Installed** tab, click the name of the application record that you want to convert.
3. Select the Convert to Development Mode related link.
4. Select **Convert**.
   - If the installed version of the application is equal to the latest version that is published to the Application Repository, you see the following dialog:



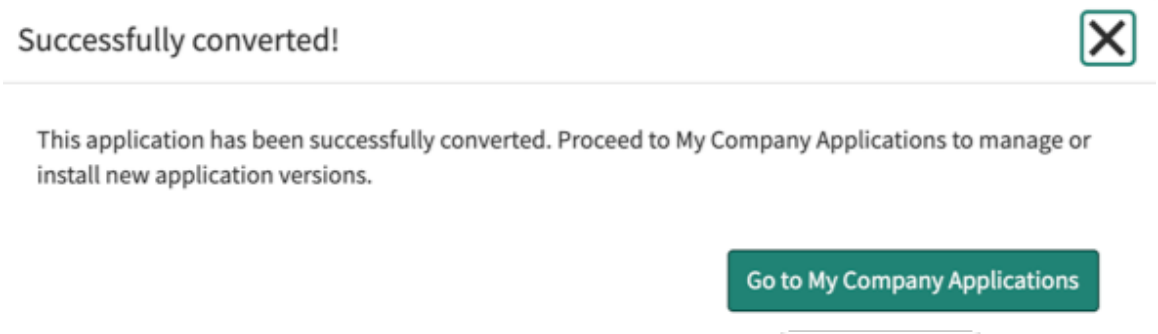
   - If the installed version of the application is lower than the latest version that is published to the Application Repository, you see the following warning:

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- The latest version of this application is not installed. Please ensure your application is updated to the intended version to use for development prior to conversion

  - If an instance is connected to a self-hosted application repository, you may see this warning regardless of the availability of a higher version in the repository. Check your repository to ensure that the development version that you require is installed in this case.

- After the successful conversion, the following message is displayed:



  - Close the window to be redirected to the development application form.

Install customizations from an application repository

Install a customization of a store application from the Application Repository to your instance.

Before you begin

Role required: admin

Publish a customization of a store application to the application repository. Check the entitlement type of the application to verify that your instance is entitled to the application.

Procedure

1. Navigate to **System Applications** > **All Available Applications** > **All**.
2. Find and select the application.
3. Select **Proceed to update**.
4. Select a customization version to install.

   Note: For scoped applications installed via plugins, the base system version is the ServiceNow release version that you're currently using (for example, Brazil).

5. Select **Install**.

Preserve applications and customizations in development during a system clone

Manually preserve a copy of each application and customization that you currently have in development before you can clone the application version to the target (development) instance.

Before you begin

Ensure that you have write access to the application record.

Ensure that you have access to a source control repository.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Role required: admin

About this task

The cloning process does not preserve version differences for applications and app customizations in development. Instead, the system clones only the copies of the application and app customization versions that are installed on the source instance onto the target instance. If the target instance had a development version of the same application, the application is editable after the clone, but is at the version that was installed on the source instance. If the application was missing from the source instance, the cloning process deletes the application from the target instance.

Procedure

1. To preserve the application on the clone target instance, do one of these actions:

Version differences between instances

Application version state	Action to take
The application version on the clone target instance is different than the source instance version.	Export each application from the clone target instance. The choices include: ○ Link each application to a source control repository. **Note:** If the application is already linked to a source control repository, commit the latest version to it. ○ Publish each application to an update set.
The application is available only on the clone target instance.	
The application version on the clone target instance is the same as the source instance.	None. The system clone process copies this application version onto the target instance during the clone.

2. Request a system clone of the source instance to the target instance.

   Example
   For example, clone your production instance over your development instance.

3. After the clone process finishes, log in to the clone target instance.

4. **Note:** If source control is linked, then post clone, the platform will automatically retrieve applications and customized applications. If this is disabled via `glide.source_control.post_clone_import_enabled` you will need to manually retrieve by doing the following.

   If you saved each application to a source control repository, use one of these actions to retrieve them from the source control repository:

   Note: For what to expect after application customization post clone, see Results post cloning for application customizations.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Retrieve applications from a source control repository

Application installation state	Action to take on clone target
The application and customization were previously installed on the source instance.	Apply remote changes from the source control repository.
The application was never installed on the source instance.	Delete the repository configuration (sys_repo_config) and import the customization from the source control repository.

Remote changes after clone

Field	Description
glide.source_control.post_clone_import_enabled	To disable the apply remote changes automation, set to **False**. The default is **True**.
glide.source_control.post_clone_import_delay_time_sec	To provide a delay time, which will delay processing of the queue, provide a value. The default is zero.
glide.source_control.post_clone_import_pause_refresh_time_sec	To provide a time interval in which the refresh repository job will not run, provide a value. The default is three hours (10800).

5. If you saved each application to an update set, do one of these actions to retrieve them from the update set:

Retrieve applications from an update set

Application installation state	Action to take on clone target
The application was previously installed on the source instance.	**a.** Delete the application version that was cloned from the source instance. **b.** Load the update set that contains the current application version.
The application was never installed on the source instance.	Load the update set that contains the current application version.

Result
The applications previously in development are available for further development on the clone target instance.

Example: Preserve the Marketing Events application

Let's say that your company previously created version 1.0 of a custom application called Marketing Events. You have already published version 1.0 of the Marketing Events application to the application repository and installed it on your production instance.

Over time, users have submitted enhancement requests for the application, and you decide to develop version 2.0 of the Marketing Events application on a non-production instance to address these requests. As development nears completion, you want to update your non-production instance to the latest copy of production for some comprehensive testing.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Because you previously used a source control integration to develop version 1.0 of the Marketing Events application, you have already linked the Marketing Events application to a source control repository. You commit version 2.0 of the Marketing Events application to the source control repository.

You schedule a clone of the production instance over the development instance. After completion, you log in to the development instance and see that it has version 1.0 of the Marketing Events application, because that was the version installed on the source instance.

Because the application was already installed on the source instance, you apply the remote changes from the source control repository to receive the latest application version. The development instance now has version 2.0 of the Marketing Events application and is available for further development and testing.

Related topics

- Link an app to source control in ServiceNow Studio
- Commit changes to a repository
- Publish an application to the application repository
- Install a ServiceNow Store application
- Import an app from source control in ServiceNow Studio
- Load customizations from a single XML file

Results post cloning for application customizations

The results to expect post cloning for application customization display the expected behaviors based on the state of the application, and the actions to recover your application customizations.

The following are the expectations based on the state of the application customizations, and the actions to recover your customizations.

Application Customization expectations post cloning

State of the App Customization on Source Instance	State of the App Customization in the Target (Development) instance pre-clone	State of the App Customization in Target (Development) instance post clone	Action to take to recover App Customization in the Target (development) instance post clone
Base application is not installed	Base application version 1.0 and App Customization version 1.0	Base application is not installed	Reinstall Base application and App Customization
Base application not installed	Base application version 1.0 and App Customization in Source Control	Base application is not installed, and the sys_repo_config has an empty app field.	Remove the repository configuration (sys_repo_config) record (the app field has been emptied) and import again from Source Control and install Base application.
Base application version 1.0	Base application version 1.0 App	Base application version 1.0	Reinstall App Customization

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Application Customization expectations post cloning (continued)

State of the App Customization on Source Instance	State of the App Customization in the Target (Development) instance pre-clone	State of the App Customization in Target (Development) instance post clone	Action to take to recover App Customization in the Target (development) instance post clone
	Customization version 1.0		
Base application version 1.0	Base application version 1.0 and App Customization in Source Control	Base application version 1.0, repo_config is retained and the App Customization version displays as none in the all apps page.	Apply the remote changes from the Git repository.
Base application version 1.0 and App Customization version 1.0	Base application version 2.0 and App Customization version 2.0	Base application version 1.0 and App Customization version 1.0	Reinstall App Customization version 2.0 and Base application version 2.0
Base application version 1.0 and App Customization version 1.0	Base application version 2.0 and App Customization version 2.0 in Source Control	Base application version 1.0 and App Customization version 1.0, sys_reconfig is retained	Apply the remote changes from the Git repository.
Base application version 2.0 and App Customization version 2.0	Base application version 1.0 and App Customization version 1.0	Base application version 2.0 and App Customization version 2.0 is installed.	Both Base and App Customization are at the latest versions.
Base application version 2.0 and App Customization version 2.0	Base application version 1.0 and App Customization version 1.0 in Source Control	Base application version 2.0 and App Customization version 2.0 is installed with repo_config retained, so the branch version is on 1.0	Both Base and App Customization are at the latest versions.

Application Repository for self-hosted, air-gapped customers

This application feature enables self-hosted customers to use their own internal application repository just as cloud customers do. These customers don't install the Application Repository via the ServiceNow® Store; they use an alternative method.

Overview of Application Repository for self-hosted

Customers whose instances are hosted on the ServiceNow® cloud have access to an application repository by default via the public ServiceNow® Store. However, self-hosted and air-gapped customers don't have such access, and thus can't take advantage of advanced application development features.

⚠ **Warning:** Most customers install applications through their networks from the ServiceNow® Store. This topic covers customers who aren't using networked sources.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The term "air-gapped" refers to the condition when a computer or network has no network interfaces, either wired or wireless, connected to any outside networks. When you want to move data from one source to another, you must use an alternative method. To move data between sources that aren't on a network, you must use an alternative such as the one described in this topic.

The ServiceNow® Application Repository is a scoped application that can be installed on a customer instance, and enables usage of the latest application development features inside their own private network without the need to communicate with external networks.

Recommended installation

It is recommended to install the Application Repository on a dedicated instance used only for that purpose. Installing in this way reduces the load on other instances from other transactions and improves the experience of installing other Store applications. After you have successfully installed the application, other networked client instances can publish and install application artifacts to and from this instance.

Acquiring the application

Before attempting to install the application, ServiceNow self-hosted "air-gapped" customers should reach out to their account representative. They will work with Support and your team to set up the installation process. To learn more, see the Knowledge Base article, How On-Prem customers can request a Store App.

How the setup process works

Instances are authenticated with the repository via a "pairing" process that stores instance credentials on the server-side, and needs to be done only once per instance. After an instance is "paired" with the repository, it can start interacting with the repository via its various application development features.

Installing the Application Repository store app on an air-gapped instance

Because self-hosted, air-gapped customers don't allow their instances to communicate with external networks, they don't install the Application Repository directly from the ServiceNow® Store. These customers should use the following procedure instead.

Before you begin

Note: Customization of the Application Repository is not supported or recommended.

Role required: You will need the maint role to install and configure, and then just the admin role once the configuration is complete.

Warning: If your instances are already connected to the public ServiceNow® Store, you should not install the Application Repository. Doing so causes these instances to lose access. This product is intended for customers who can't connect to the public store.

Note: Install on a dedicated ServiceNow® instance separate from your other instances.

Procedure

1. Install the application repository using the instruction in the following Knowledge Base article: How On-Prem Customers Can Request a Store App.
2. After installing, complete the steps in Configure the application repository on an air-gapped instance.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Configure the application repository on an air-gapped instance

After installing the application repository, you must configure it using the following procedure.

Before you begin

Role required: You'll need the maint role to install and configure, and then only the admin role after the configuration is complete.

Procedure

1. Pair an instance with the application repository.

   a. Log in to the instance you want to connect to the application repository.

   b. Set the system property *sn_appclient.repository_base_url* to the URL of your application repository instance.
   For example, `http://localhost:8080/`.

   c. Clear the values of the *sn_appclient.upload_base_url* and *sn_appauthor.upload_base_url* properties.

   d. Set the value of the *sn_apprepo.credential* property in a Global scope to any non-null value, such as "secret."

      i. SSH into the instance.

      ii. Change the directory to `/root/instance/instance_<portno>/conf/overrides.d` using `cd /root/instance/instance_<portno>/conf/overrides.d`.

      iii. Open / Create the `glide.properties` file.

      iv. Add the credential property to the end and save the file `[sn_apprepo.credential=<value>]`.

      v. (Shutdown.sh/Startup.sh) Restart the glide, or run `Packages.com.glide.util.GlideProperties.loadPropertyFile(new Packages.java.io.File(gs.getProperty("glide.home.dist")+"/conf/overrides.d/glide.properties"));` in background scripts to dynamically load the properties file at runtime without restarting the instance.

      vi. Verify by printing the property in a background script `gs.info(gs.getProperty("sn_apprepo.credential"));`.

   e. Add *sn_appclient.repo_auth_name* with **sn_repo.AppRepo** as its value.

   f. Set the *glide.test_instance* property to **False** on both the application repository instance and the client instance.

   g. Set the *sn_appclient.client_calls_allowed* property to **True**.

      Note: A scheduled job might set this property to **False** when not connected.

   h. Set the *sn_appclient.app.install.offline* property to **False** on the client instance.

   i. Select **Submit**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

2. Log in to the instance where the application repository is installed and complete the following steps.

   a. Navigate to the **core_company.list** table and ensure there is a record with the **Primary** field set to **True** or create one with any user-defined name.

      Note: The details of this record are not important.

   b. Access the **sn_repo_instance.do** screen and create a new instance record for the client instance that you want to connect.

      i. Ensure that **State** is set to Pairing.

      ii. Enter the name of the instance that you want to connect (on the **stats.do** screen on that instance) in the **Name** field.

      iii. Leave all other fields empty. They're populated automatically.

   c. Repeat the previous step for any additional instances that you want to connect.

3. Log back in to the instance you used in Step 1 (the instance that you want to pair with the application repository) and navigate to the **Scripts - Background** module.

   a. Select **sn_appauthor scope** from the drop-down list.

   b. Run the following script: `new ConfigChecker().checkForChanges()`.

4. **Optional:** To remove the instance, navigate to the Instance record (sn_repo_instance table) and change the State to **Blocked**, which temporarily restricts access to the instance, or delete the instance.
   If you need the instance again, you can change the State to **Paired** again.

   Warning: If the instance name, instance ID, or credential of a paired instance changes, it must be paired again. Manually updating any of these values on the instance record isn't recommended.

What to do next

After an instance is paired, it's fully set up to use the application repository. You can test your configuration by publishing a scoped application as described in Publish an application to the application repository. After publishing, you can verify that the app was successfully published by locating it in **All** > **Application Repository** > **Artifacts** > **Internal Apps**.

Publish an application to the ServiceNow Store

Publishing an application to the ServiceNow Store makes it available to everyone.

Before you begin

To publish an application to the ServiceNow Store:

- Create an application within a private application scope.
- Join the Technology Partner Program.
- Have the application certified.

Note: Applications in the global scope cannot be published to the ServiceNow Store.

Role required: admin, or delegated_developer with Publish To App Store permission enabled

About this task

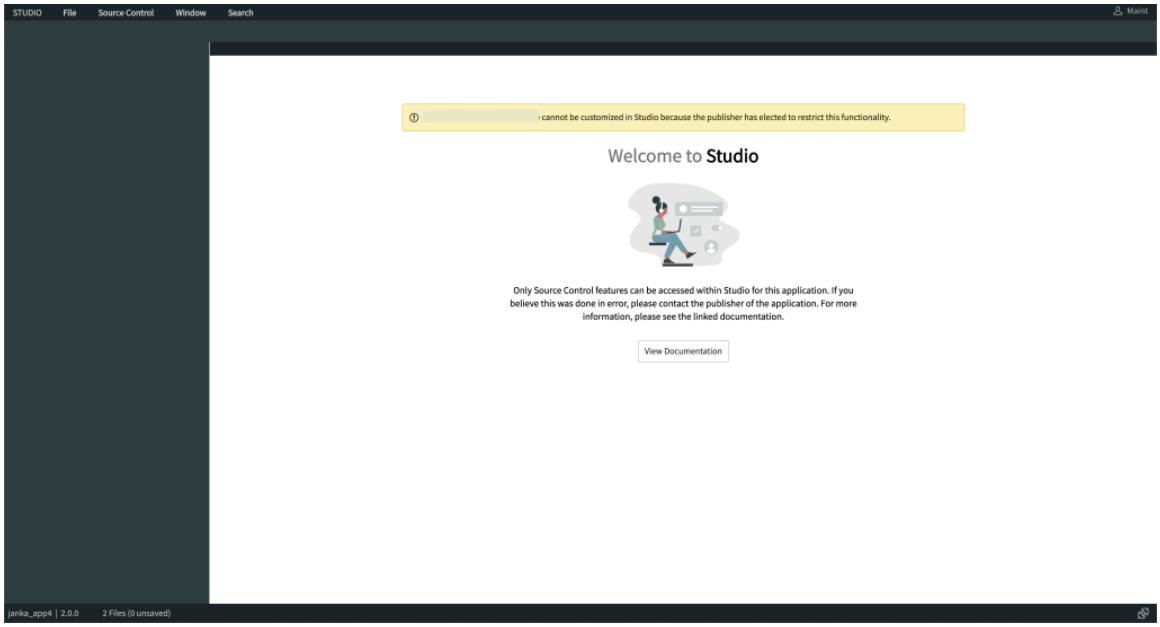
After you have met the prerequisites, you can publish the application to the ServiceNow Store

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Note:

The **Can Edit Application in Studio** property defaults to true for new applications, but you can set it false before publishing.

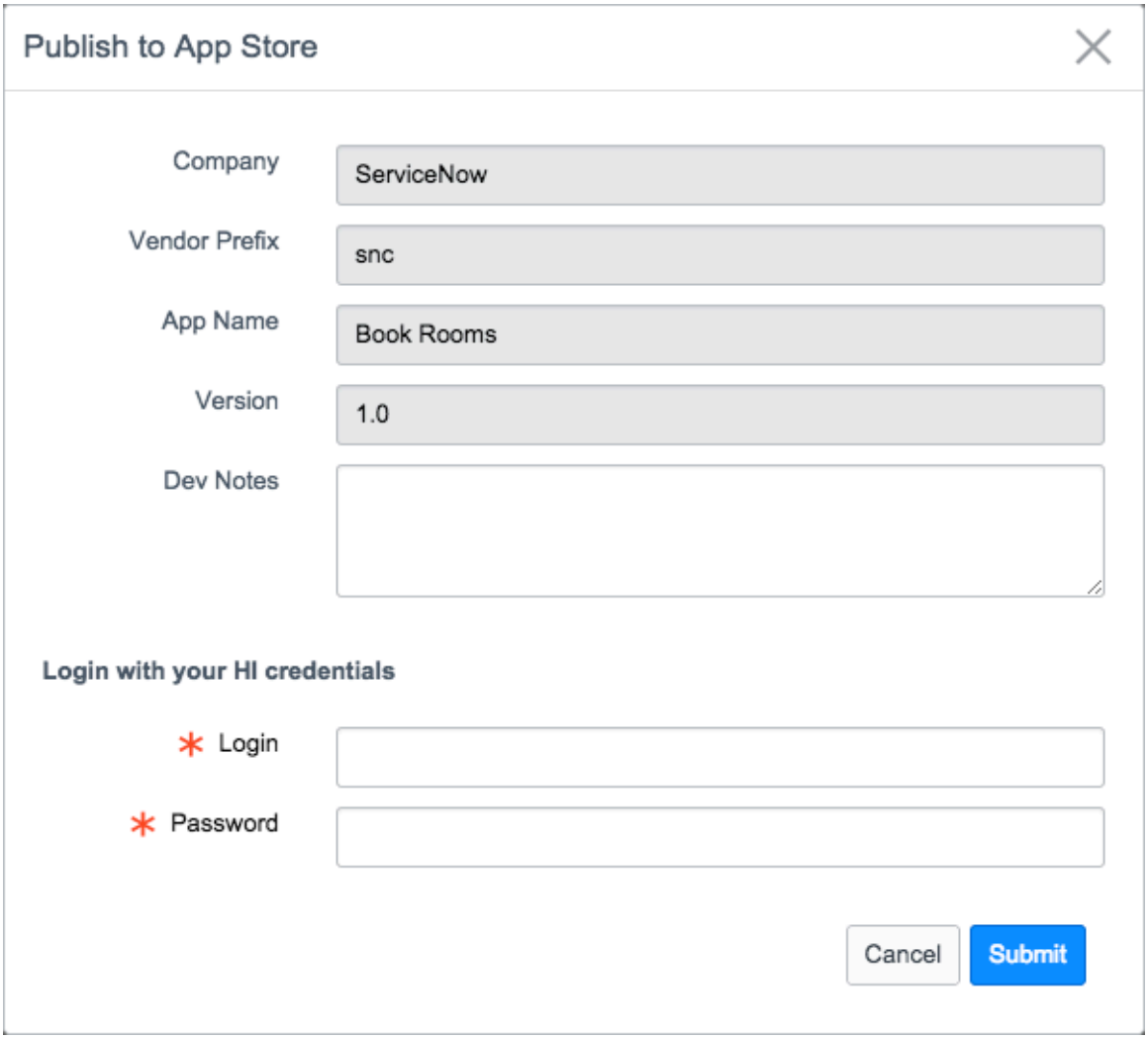
In Studio, when an application customization has the **Can Edit Application in Studio** property set to false, the user sees this warning:



Procedure

1. Navigate to **All** > **System Applications** > **Applications**.
2. Click the **In Development** tab.
3. Open the application record you want to publish to the ServiceNow Store.
4. Click the **Publish to Store** related link.
5. **Optional:** Fill in the fields, as appropriate (see table).

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



6. Enter your HI credentials.
7. Click **Submit**.
   The system uploads the current version of the application to the ServiceNow Store allowing other users to download it.

Related topics

Delegate development and deployment permissions to personnel

Create application files to include sample data

Include sample records from an application data table when sharing a custom application.

Before you begin

Role required: admin, or delegated_developer

About this task

The system can export selected records as application files that are included as part of the application update set when you share it. Including application files in an update set is not intended for the export and import of large numbers of records between instances. If you're trying to move data between instances, see Import from another instance instead.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

The application data only includes the version of the records that existed when the records were shared. The system does not update this snapshot of the application data when the records change. Application designers can include data on a table by table basis.

Procedure

1. In your application, open the list view of the data table you want to use.

   Note: To navigate directly to the list view of any table, use the `<table name>.list` shortcut.

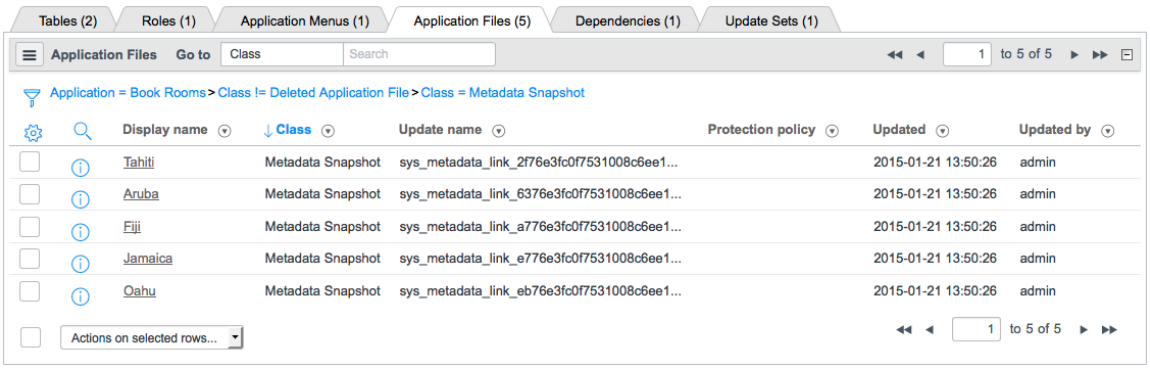
2. Filter the list to display the records you want to include.
3. Perform the appropriate action for the list version.

Version	Action
List v2	Open any column context menu and select **Create Application Files**.
List v3	Open the list title menu and select **Create Application Files**.

4. For **Load When**, select when the application record includes application data.

Option	Description
New Install and Upgrades	Includes application data whenever the application is installed or upgraded.
New Install	Includes application data only when the application is installed.
New Install with Demo Data	Includes application data only when the application is installed with demo data.

5. Click **OK**.
   The system adds the records to the application files related list.



6. Repeat steps 1–5 for each application data table you want to include.

Publish an application to an Update Set

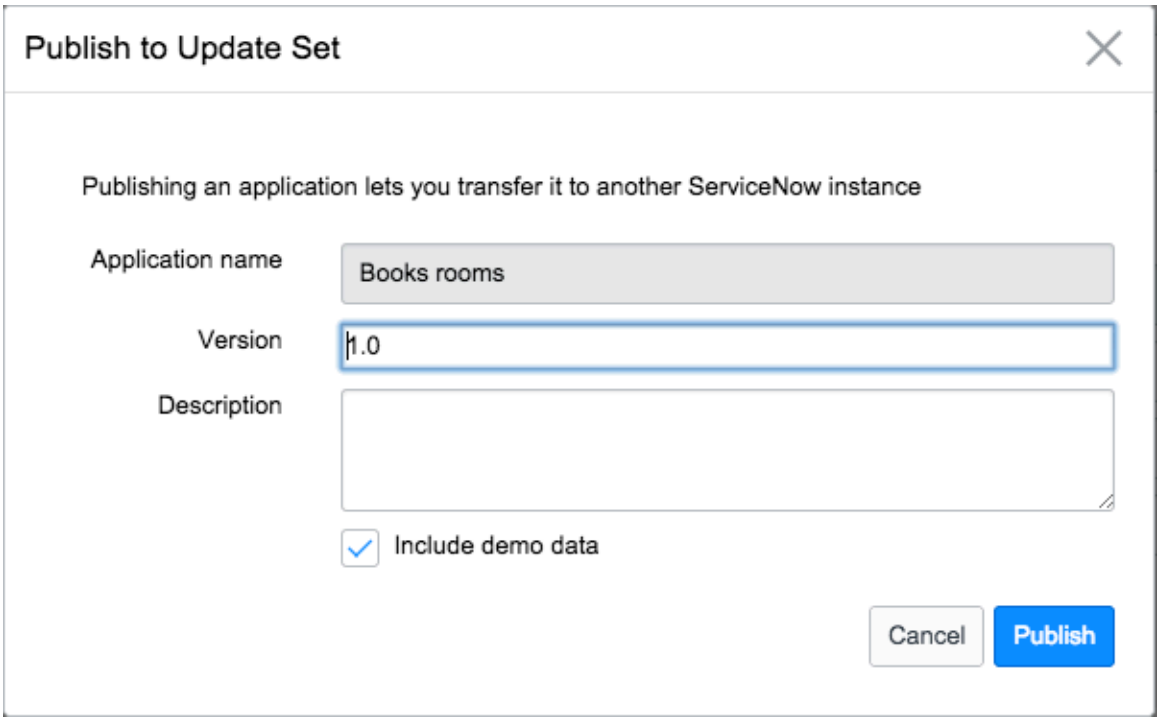
Publishing an application creates an update set containing the current version of all application configuration records.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Before you begin
Role required: admin, or delegated_developer with Publish To Update Set permission enabled

About this task
You can use this update set as a backup file for auditing purposes or to transfer the application to another instance.

Procedure
1. Navigate to **All** > **System Applications** > **My Company Applications**.
2. Select the **In Development** tab.
3. Open the application record you want to create an update set for.
4. Select the **Publish to Update Set** related link.
5. **Optional:** On the form, fill in the fields.



Publish to Update Set

Field	Description
Application name	[Read-only] Displays the name of the application that you are publishing.
Version	(Optional) Enter version information to append to the Update Set name in dot-notation such as 1.2.3. The platform saves the information you enter here in the application **Version** field. The Update Set has the name <Application name> - <Version>. If you leave this field blank, the initial Update Set has the name <Application name> and subsequent Update Sets have the name <Application name> - <Sequential number>.
Description	Enter a description for the Update Set. By default, this field contains the short description of the application.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Field	Description
Include data	(Optional) Select the check box to include a limited number of data records from each table in the application. Use this feature to package sample data with your applications. **⚠ Warning:** Using this feature to migrate large quantities of data records between instances can cause performance issues, as it is not intended for this purpose. To migrate data, use an instance-to-instance import. You can adjust the maximum number of data records to include with an application. See Import sets. ○ If your sample data includes tables with record numbering, the current counter value is also transferred. When the update set is applied on another instance, the counter is set to the larger of the sample data or the target instance counter. ○ For translated fields, only records in English are transferred.

6. Select **Publish**.
   A new update set is created and the latest update of each application file in the application is copied into it. The update set is marked as complete.
7. Transfer the update set to another instance according to your test process.
   - Retrieve the update set from the source instance.
   - Save the update set as an XML file.
8. Run any fix scripts that are included in the application.

Related topics

Retrieve an update set

Save an update set as a local XML file

Run fix scripts

Delegate development and deployment permissions to personnel

Queued Application Operations

CICD APIs that must obtain the update **instance wide lock / mutex** to perform the requested operations are queued instead of being rejected when the update **instance wide lock / mutex** is occupied by the other operations.

Beginning in Tokyo, the CICD APIs that require to obtain the update **instance wide lock / mutex** to perform the requested operations will be queued instead of being rejected when the update **instance wide lock / mutex** is occupied by the other operations. When a CICD request is received, the corresponding CICD service constructs an application operation NowMQ (Now Message Queue) message and insert the message to the queue using NowMQ APIs. The queued messages are then polled by the scheduled job and handled one by one or in parallel if parallel processing is enabled and the operation satisfies necessary criteria.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

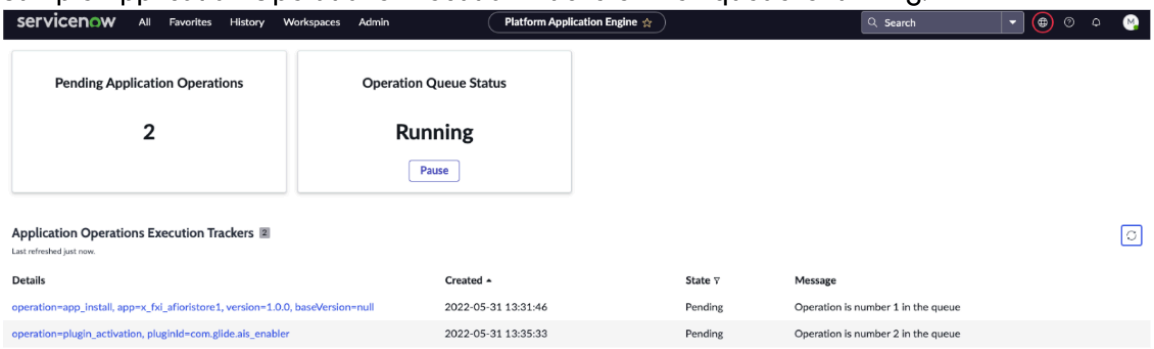
Application Operation NowMQ Message

The application operation NowMQ messages have a common subject that is "sys.applifecycle.operation". The message body of the application operation NowMQ message contains a JSON object that includes the sys_id of the Execution Tracker (also refer to as the progress id returned in the CICD API response), the operation type that can be one of the following: app_install, plugin_activation, batch_install, rollback, import_app and apply_changes. It also contains the information like plugin id for plugin activation, app id or scope for application install.

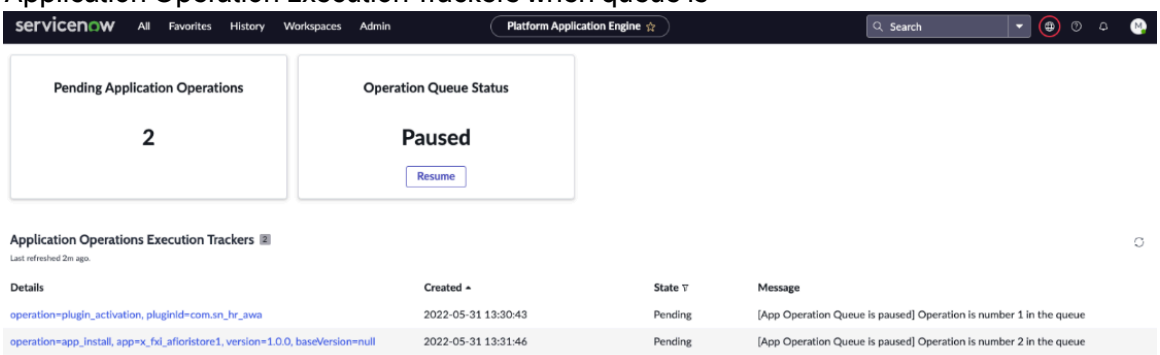
Execution Tracker for Application Operation

When the Application Operation NowMQ message is constructed and inserted, the Execution Tracker record for the corresponding CICD request is created and its sys_id is added into the body of the NowMQ message. The Execution Tracker is in Pending state initially. The Execution Tracker's "Details" column contains the information about the type of operation, and the important input parameters for the CICD request. Its "Message" column contains the information about the queue position. When the queue is paused, the message is prefixed with "[App Operation Queue is paused]".

Sample Application Operations Execution Trackers when queue is running.



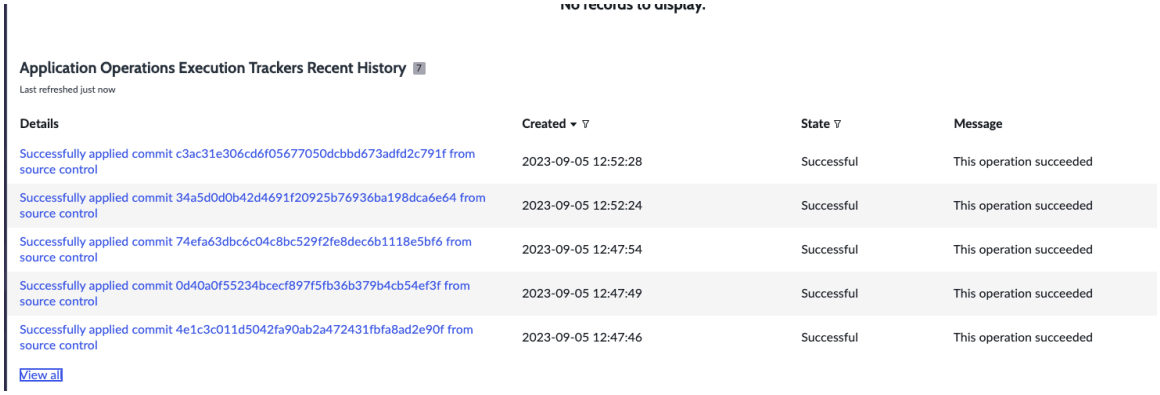
Sample Application Operation Execution Trackers when queue is paused.



Note: Operations that are in a pending state will not display in recent history.

Sample Applications Operations Execution Trackers recent history. Recent history displays historical operations that were queued and processed in the past 24hrs.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.



Manage Application Operation Queue

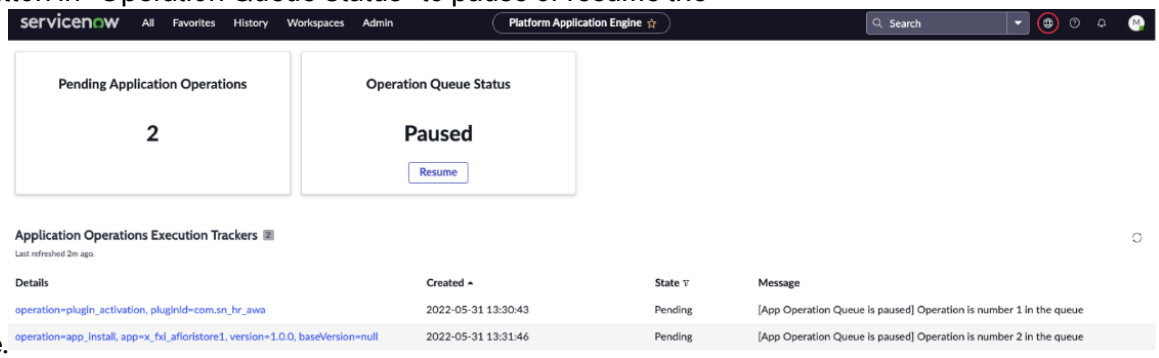
While manually installing products from All Applications is queued, manually installing an application from UI like All Applications or Application Manager isn't queued.

Sometimes, the instance may be receiving, queueing, and handling many CICD requests which may cause the manual install from UI starving for update **instance wide lock / mutex**. When this happens, admin can temporarily pause the Application Operation Queue.

Admin can manage the Application Operation Queue through System Diagnostics->Application Operation Queue UI page. On "Operation Queue Status" panel, the admin can pause or resume the queue. Admin can also cancel the pending execution tracker, this will eventually remove the corresponding queued message from NowMQ by the App Operation Queue Health Monitor job.

Application Operation Queue UI page

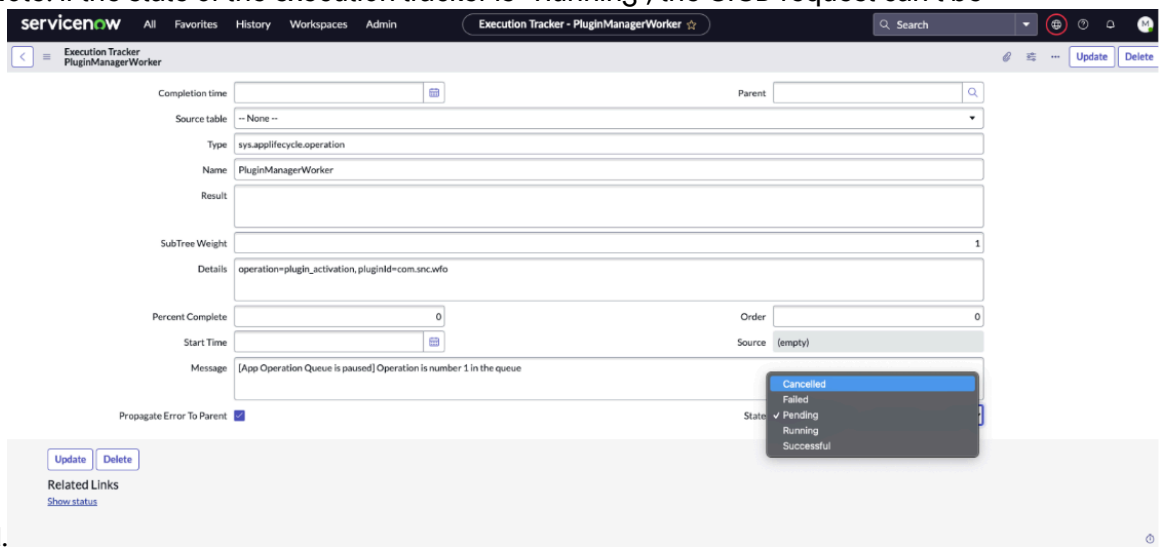
Sample Application Operation Queue UI page. Admin can click the button in "Operation Queue Status" to pause or resume the queue.



Click the "Application Operations Execution Trackers" list item, it opens the Execution Tracker form. If the queued message is pending in the queue, update the execution tracker state to "Cancelled" and save the change will cancel the corresponding queued CICD

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

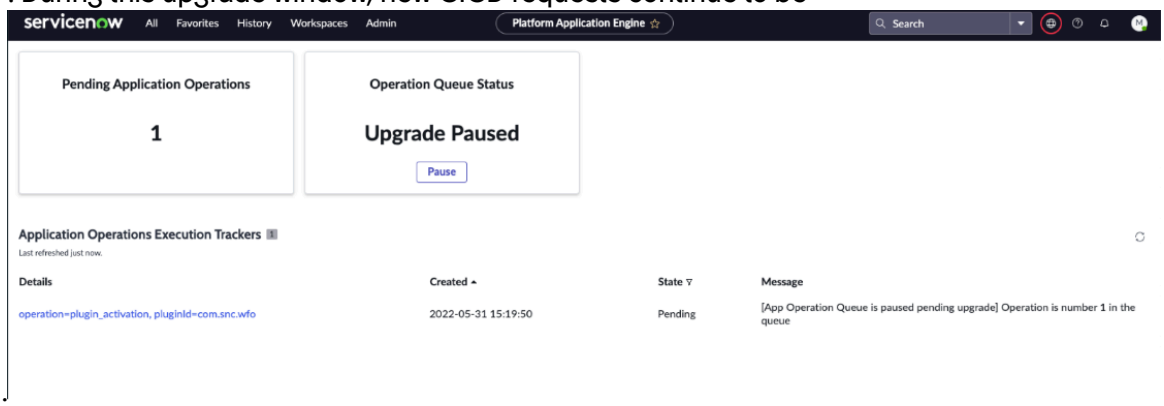
request. Note: if the state of the execution tracker is “Running”, the CICD request can’t be cancelled.



Application Operation Queue and Upgrade Window

By default, 2 hours (can be customized through sys property “com.glide.update_operation.queue_upgrade_window”) before the scheduled upgrade, Application Operation Queue stop processing queued messages.

The Application Operation Queue status is changed to “Upgrade Paused”. During this upgrade window, new CICD requests continue to be queued.



When upgrade completes, the Application Operation Queue resumes processing the queued messages automatically.

Impact to CICD Pipeline

The existing request/response contracts for the CICD APIs aren’t changed. The operation failure due to update **instance wide lock / mutex** conflicts that are observed in a pre-Tokyo release won’t be seen. The request is queued and served one by one or in parallel depending on they job type.

CICD APIs supporting queueing

The following CICD APIs are queued for processing.

- api/sn_cicd/app_repo/install
- api/sn_cicd/v1/app_repo/install
- api/sn_cicd/v2/app_repo/install

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

- api/sn_cicd/app_repo/rollback
- api/sn_cicd/v1/app_repo/install
- api/sn_cicd/v2/app_repo/rollback
- api/sn_cicd/sc/apply_changes
- api/sn_cicd/v1/sc/apply_changes
- api/sn_cicd/v2/sc/apply_changes
- api/sn_cicd/app/batch/install
- api/sn_cicd/sc/import
- api/sn_cicd/plugin/{plugin_id}/activate
- api/sn_cicd/plugin/{plugin_id}/rollback

Parallelize Application Installs and Plugin Activations

Starting in Tokyo, the following operations can execute in parallel with other operations.

- api/sn_cicd/app_repo/install
- api/sn_cicd/v1/app_repo/install
- api/sn_cicd/v2/app_repo/install
- api/sn_cicd/plugin/{plugin_id}/activate

All queue processing takes an **instance wide lock / mutex** and holds this mutex until any queued operation is completed. This lock is called **UpdateMutex**, and its status can be viewed in the **sys_mutex** table. During this time, operations that take this same lock (app installs, plugin activations, source control operations) are not performable via the UI. The queue can still be paused via the **Application Operation Queue** page to release the lock after currently running jobs have been completed.

Parallelization is enabled by default. It can be turned off by using the property `com.glide.update_operation.parallel_operation_enabled`, and all operations will run sequentially from the queue as in previous releases.

Limits on Parallelization

The queue processor determines if an enqueued job can run. If a job can run, it is scheduled to be picked up by an instance node at first availability. If not, it is returned to the queue, and the processor evaluates the next job in the queue for processing.

There is a limit to the maximum number of jobs that can be executed in parallel, which defaults to 2. This property can be toggled via `glide.update.app_operation_queue.parallel.max` but keep in mind that there is an upper limit of available threads to perform the install and increased parallelization takes additional memory that can slow the instance for active users.

Obtaining Resource Locks

The following plugins and applications can't be installed in parallel.

- Any two operations that share the same scope including customizations.
- Any two operations that cause schema changes.
- Any two operations containing fix scripts.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Queue processing determines if these criteria are met for enqueued operations and defers not processable jobs if necessary. The Progress ID of the job is updated to reflect if the operation is waiting for appropriate resource locks to be obtained. An executing operation maintains a list of the resources (scopes, schema, fix scripts present) in the sys_padlock table, and the insert and release of these locks can be observed in this table by the Progress ID. If a queued job is deferred due to inability to obtain necessary locks on resources, it is put on a cool-down to allow other messages to process. The cool-down period can be modified with property `com.glide.update_operation.job_cancel_timeout_minutes`. The job is still in the queue and is visible on the **Application Operation Queue** page.

If the queue is unable to download an application package or find a plugin to check for necessary resources to obtain, it logs an error against that queued operation. If the operation fails to preview its resources a determined number of times, the tracker fails, and the job is removed from the queue. The maximum number of times defaults to 3 and can be modified by the `com.glide.update_operation.max_failure_count` property.

Note: Failing to obtain necessary locks doesn't count as a failure attempt. Only errors encountered such as failing to download an application package from AppRepo count.

Custom licensing for ISV applications

Monitor the usage of ISV applications with Subscription Management. Create a definition for your store application with the metadata you want collected. After publishing the application with the definition to the store, Usage Analytics runs and aggregates your defined metrics.

Overview of custom licensing

Applications that you create with the can be published to the ServiceNow Store. Monitor the usage of your applications with custom licensing.

You can use custom licensing on ISV applications that you publish to the ServiceNow Store if they meet the following criteria:

- The application is licensable.
- The application is not scoped in ServiceNow or Global.
- The application has a capacity subscription model.

Custom licensing functions through the usage analytics (com.glide.usageanalytics) and scoped app author (com.sn_appauthorr) plugins. These plugins are active by default in the App Engine. Additionally, your instance must be connected to an application repository or application store instance.

Custom license definitions

Custom licensing is achieved by creating a definition. A definition is a set of licensing metrics that you define based on what usage information you want to collect for your application. See Create a definition for your store application.

Create a definition for your store application

Create a definition for your store app to define the licensing metrics you want collected.

Before you begin

Role required: admin or developer

About this task

Create a definition for an application in development or for an application you have already published. However, if you assign a definition to a published application, you will have to

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

re-publish it with the new definition. Additionally, you can only assign one definition to an application.

Note: You can only create a definition for capacity subscriptions. See Types of subscriptions in Subscription Management.

Procedure

1. Navigate to **All** > **System Applications** > **My Company Applications**.
2. Select the application.
3. Under the **Subscription Management** section, set **Subscription Model** to **Capacity**.
4. Select the search icon next to **License Definition**.
   The **Application License Definitions** pop-up loads.
5. Select **New**.
6. Fill in the **Application License Definition** record.

Field	Description
Name	Name of the definition.
Description	Brief summary of the definition.
Metric Type	Type of collected metric.
Frequency	How often metrics are collected.
Performance Validated	Check the box if you have tested the performance of the definition.
Table	Table used to determine usage. **Note:** Only lists tables associated with the current application.
Query	Criteria to use on the table.
Aggregation	Type of aggregation. ○ Set to Count to count all records. ○ Set to Count (Distinct) to count unique records only.
Aggregation Column	Table column used for aggregation. **Note:** Column must be indexed to be aggregated.
Group By	Columns in the table to group the data by. Max of 3. **Warning:** Case sensitive.

7. Select **Submit**.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Your new definition appears in the **Application License Definitions** pop-up.

8. Select the definition to apply it to the application.

What to do next

Create a new version of the application and publish it to the ServiceNow Store. While your application is in review, any changes to the definition may cause subsequent versions of your application to be rejected. Once the application is approved and published to the ServiceNow Store, the definition becomes read-only.

To change the definition of an application already on the ServiceNow Store, you must create a new definition, submit a new version of the application, and go through the review process.

Domain separation and Creator Workflow apps

Domain separation enables you to separate data, processes, and administrative tasks into logical groupings called domains. You can control several aspects of this separation, including which users can see and access data.

Domain separation is not supported for App development and low-code products.

Support level: No support

- The domain field may exist on data tables but there is no business logic to manage the data.
- This level is not considered domain-separated.

For more information on support levels, see Application support for domain separation.

Maintaining your application

Use your app builder tool to update and modify your application. Use your testing tool to verify that your application still functions properly.

Analytics and Dashboards

Performance Analytics Reporting	Analytics Hub
	
Integrate data visualization and analytics functionality into the workspace experience.	Use this exploratory view of indicators for more detailed analysis. You can see trends, predictions, breakdowns, and associated records for a specific indicator.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Analytics Center



Use the Platform Analytics Workspace with a single page for all the dashboards, visualizations, analytics answers, and insights on your instance.

Performance Analytics Dashboards



Continually visualize historical and real-time process health statistics in role-based dashboards that enable individual stakeholders to make informed decisions.

Security

Software Asset Management



Systematically track, evaluate, and manage software licenses, compliance, and optimization.

Security Operations



Bring incident data from your security tools into a structured response engine that uses intelligent workflows, automation, and a deep connection with IT to prioritize and resolve threats based on the impact they pose to your organization.

Service Mapping



Discover all application services in your organization and build a comprehensive map of all devices, applications, and configuration profiles used in these application services.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Monitoring

Monitor application health



Use the Application Service Dashboard to monitor the health state of application services. Then you can reduce the number of incomplete application services by populating them and adding any missing details.

For more information

You can either create a new workspace with an Analytics Center page in App Engine Studio or create a page from the Analytics Center page template. Either way, you have the same functions as the Analytics Center in the Platform Analytics Workspace.

Analytics Hub

In the Analytics Hub, analyze indicator scores by aggregating data, comparing scores, or viewing changes over time, and filter scores by breakdown. Enhance the Analytics Hub by adding targets, thresholds, trend lines, and useful comments for significant changes.

Monitor application health

Monitor counts indicating the health state of application services using the Application Service Dashboard. Next, reduce the number of incomplete application services by populating them and adding any missing details. To use application services effectively, ensure that application services are fully configured and populated.

Performance Analytics Dashboards

Your associates can get relevant, personalized insight by clicking a button instead of exporting data from databases and spreadsheets. You no longer manually create reports that quickly become stale and outdated.

Performance Analytics Reporting

Explore KPIs, and get answers and insights on analytics. Within the Platform Analytics Workspace, you can ask a natural language question and be shown the relevant analytics in an on-the-fly visualization, view any Next Experience dashboard that you own or has been shared with you, and create dashboards and data visualizations from your workspace runtime.

Security Operations

The ServiceNow® Security Incident Response application tracks the progress of security incidents from discovery and initial analysis, through containment, eradication, and recovery, and into the final post incident review, knowledge base article creation, and closure.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries. Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.

Service Mapping

Service Mapping enables IT departments of companies, organizations, and cloud companies providing platform as a service to create a service-aware view of infrastructure.

Software Asset Management

The ServiceNow® Software Asset Management (SAM) application systematically tracks, evaluates, and manages software licenses, compliance, and optimization. You can reclaim unused software rights, purchase new software rights, and manage allocations for entitlements.

© 2026 ServiceNow, Inc. All rights reserved.
ServiceNow, the ServiceNow logo, Now, and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc., in the United States and/or other countries.
Other company names, product names, and logos may be trademarks of the respective companies with which they are associated.