Yokohama Telecommunications Service Operations Management

Last updated: 06/13/2026

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.

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

servicenow

Table of Contents

Telecommunications Service Operations Management

Proactively monitor the health of your networks and services that helps to prevent potential downtime. Streamline your response with ServiceNow® Event Management and Metric Intelligence.

ServiceNow® Telecommunications Service Operations Management (TSOM) integrates with monitoring tools such as Event Management and Metric Intelligence to simplify operations and provides an end-to-end service view across telecommunications technology domains. The TSOM uses the TM Forum Alarm Management API to automate the collection, correlation, and analysis of vast network event data across disparate domains. It provides front and back-office teams with a single end-to-end service health view.

Handling the external events using Telecommunications API notification

Use the Telecommunications API notification to receive the external events that occurring in the customer network system so that you can promptly respond to them in the ServiceNow AI Platform.

Introduction to Telecommunications API notification

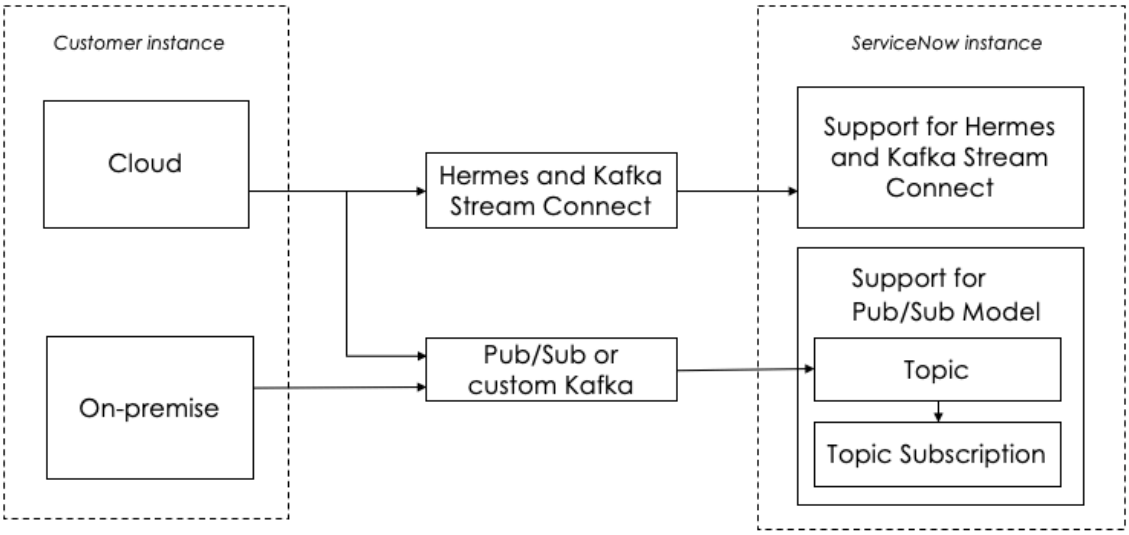
Telecommunications API notification is a feature available in the Telecommunications Alarm Management Open API application. The Telecommunications API notification enables ServiceNow to receive the incoming notifications that occurring in the external network system and responds to them in a timely manner. It enables the broadcasting of events to the external systems through platform capabilities by eliminating the need for point to point connections.

Telecommunications API notification receives incoming notifications from the external systems that are subscribed on your network. When the notifications are received from the external system, you can create the events for the responses by using the Event Management application. Based on the collected information, the Event Management provides dashboards showing a consolidated view of all service-impact events.

Telecommunications API notification data model

The following diagram shows the components in the data model for the Telecommunications API notification.

API notification data model



The Telecommunications API notification enables ServiceNow to receive incoming notifications through the event-driven architectures such as the Publisher/Subscriber (Pub/Sub) subscription model, Hermes, and Kafka Stream Connect. While cloud customers have the flexibility to select between both architectures, on-premise customers are limited to using their own Kafka or Pub/Sub subscription model.

- To learn more about Stream connect for Apache Kafka Stream, see Using Stream Connect for Apache Kafka.
- To learn more about Hermes Messaging Service, see Hermes Messaging Service.

In the Pub/Sub model, incoming notifications are categorized into topics. You use ServiceNow to publish the incoming notifications to these topics, and subscribers (customers) have the flexibility to select the topics to which they want to subscribe. This process enables subscribers to select only those messages that align with their interests. For example, if there are 10 topics for incoming messages from the external system, a customer can opt to subscribe to two of them based on their requirement. Consequently, when notifications are received from the external system, events are generated specifically for the two topics to which the customer has subscribed.

Modeling the Telecommunications API notification workflow

The following steps help to configure the Telecommunications API notification in the ServiceNow instance.

1. Create a topic: You can create topics either by manually typing the external message details or automatically collecting the available topics from the external system.
2. Create a topic subscription: You subscribe to the available topics for incoming notifications from the external system, based on the customer preference. Additionally, you generate the callback URL and register the subscription.
3. Activate the endpoint of the Telecommunications Alarm Management Open API connection: To receive responses from the external system, activate the subscribed endpoints of the Telecommunications Alarm Management Open API connection in the Workflow Studio.

4. Provide the callback URL to the external system for receiving notifications. Customer can also reuse the callback URL. When requests from TMF 688 hit the Callback URL, it initiates the *Default Alarm Event Notification Trigger* flow to create an event.

   To learn more about the functions to handle Event Notification Management Open API requests that are triggered by external trigger definitions to create, update, and delete events, see Event Notification Management Open API and TMFTopicEventAPIUtilOOB - Scoped.

This workflow creates an event in the Event Management application. To learn more about using Event Management, see Event Management.

Create a topic

Create a topic and publish the incoming notifications from the external system to the topic. By creating the topics, subscribers can select the topics to which they want to subscribe.

Before you begin

Make sure that the Telecommunications Alarm Management Open API (sn_ind_tmf642) application is installed with the ServiceNow AI Platform.

Role required: admin, sn_api_notif_ mgmt.topic_creator

About this task

You can create topics either by manually typing the external message details or automatically collecting the available topics from the external system. When you create a topic, it creates a record in the Topic [sn_api_notif_mgmt_topic] table.

Procedure

1. **All** > **Telecom API Notification** > **Topics**.
2. Select **New**.
   If you've integrated with an external system, you can select **Get Topics** to get the available topics automatically. This action triggers the *Event Alarm Notification API* subflow. To learn more about the functions that enable you to query and manipulate records in the topic, see TopicUtilOOB - Scoped.
3. On the form, fill in the fields.

Topic form

Field	Description
Topic id	Unique topic id.
Topic name	Name of the topic.
Type	Type of topic. Select one from the following: ○ Ingress: Option for inbound notification. ○ Egress: Option for outbound notification.
Header query	Encoded header query parameters. To learn more about the query parameters that follow the TMF 688 standards, see the TM Forum.
Content query	Encoded content query parameters.

Field	Description
	To learn more about the query parameters that follow the TMF 688 standards, see TM Forum.
Description	A brief description about the topic.

4. Select **Submit**.

Result
A topic is created.

What to do next
You can create the topic subscription according to the customer requirement.

Create a topic subscription

Subscribe to the topic in the ServiceNow AI Platform that you want respond to the incoming notification from the external system. By subscribing to the topic, the subscriber receives the notifications based on the topics that you subscribe to.

Before you begin

- Make sure that the Telecommunications Alarm Management Open API (sn_ind_tmf642) application is installed with the ServiceNow AI Platform.
- Create topics for the incoming notifications.

Role required: admin, sn_api_notif_mgmt.subscription_creator

About this task
You subscribe to the available topics for the incoming notifications from the external system, based on the customer preference. You generate the callback URL to share with the customers. When a request from an external system hits the callback URL, it initiates the creation of an event in the Event Management application.

Additionally, you register the topic subscription to start receiving the incoming notifications. When you create a topic subscription, it creates a record in the Topic Subscription [sn_api_notif_mgmt_subscription] table. To learn more about the methods to query and manipulate records in the Topic Subscription, see TopicSubscriptionUtilOOB - Scoped.

Procedure

1. **All** > **Telecom API Notification** > **Subscription**.
2. Select **New**.
3. On the form, fill in the fields.

Topics Subscription form

Field	Description
Topic	Topic that you want to subscribe.
CallbackURL	The callback URL that you're sharing with the external system to capture the incoming notification. The URL is generated

Field	Description
	automatically when you select **Generate CallbackURL**.
Filter query	Encoded content query parameters from the topic. You can also modify the filter query. To learn more about the query parameters that follow the TMF 688 standards, see TM Forum.
Registration status	Status of the Topic registration with the external system. By default, it's **Unregistered**. If the process is successful, the field value changes to **Registered**. Otherwise it's **Error**.
Registration message	Registration status message from the external system.
Subscription id	Unique subscription id from the external system.

4. Get the callback URL by selecting **Generate CallbackURL**.
5. Register the subscription by selecting **Register**.

Result
A trigger definition is created for the callback URL and the topic is registered to the external system.

What to do next
In the Workflow Studio, you activate the endpoints of the Telecommunications Alarm Management Open API connection.

Activate the endpoint of the Telecommunications Alarm Management Open API connection

Activate the endpoint of the Telecommunications Alarm Management Open API connection. By activating the endpoint, you receive the incoming notifications from the external system for the topic that you registered.

Before you begin
- Create the topic and subscribe to it to receive the incoming notifications.
- Generate a callback URL and register the topic subscription.

Role required: admin

About this task
You activate the subscribed endpoints of the Telecommunications Alarm Management Open API connection in the Workflow Studio to receive responses from the external system.

Procedure
1. Navigate to **All** > **Process Automation** > **Flow Designer**.
2. On the **Connections** tab, select **Telecommunications Alarm Management Open API**.

3. Open the endpoint record that you want to activate.
4. Select **Activate**.

Telecommunications API notification user roles

Administrators can assign user roles to grant access to the API notification database tables. The following standard roles for the Topic [sn_api_notif_mgmt_topic] and Topic Subscription [sn_api_notif_mgmt_subscription] tables are included in the ServiceNow system.

Telecommunications API notification roles

Role	Description
sn_api_notif_ mgmt.topic_subscription_viewer	Role that enables with read access to the Topic and Topic Subscription tables.
sn_api_notif_ mgmt.topic_creator	Role that enables with create, read, and edit access to the Topic table.
sn_api_notif_mgmt.subscription_creator	Role that enables with create and read access to the Topic Subscription table.
sn_api_notif_mgmt.subscription_admin	Role that enables with the following permissions: • Create and read access to the Topic and Topic Subscription tables. • Change the status of registration to deregister a topic subscription.

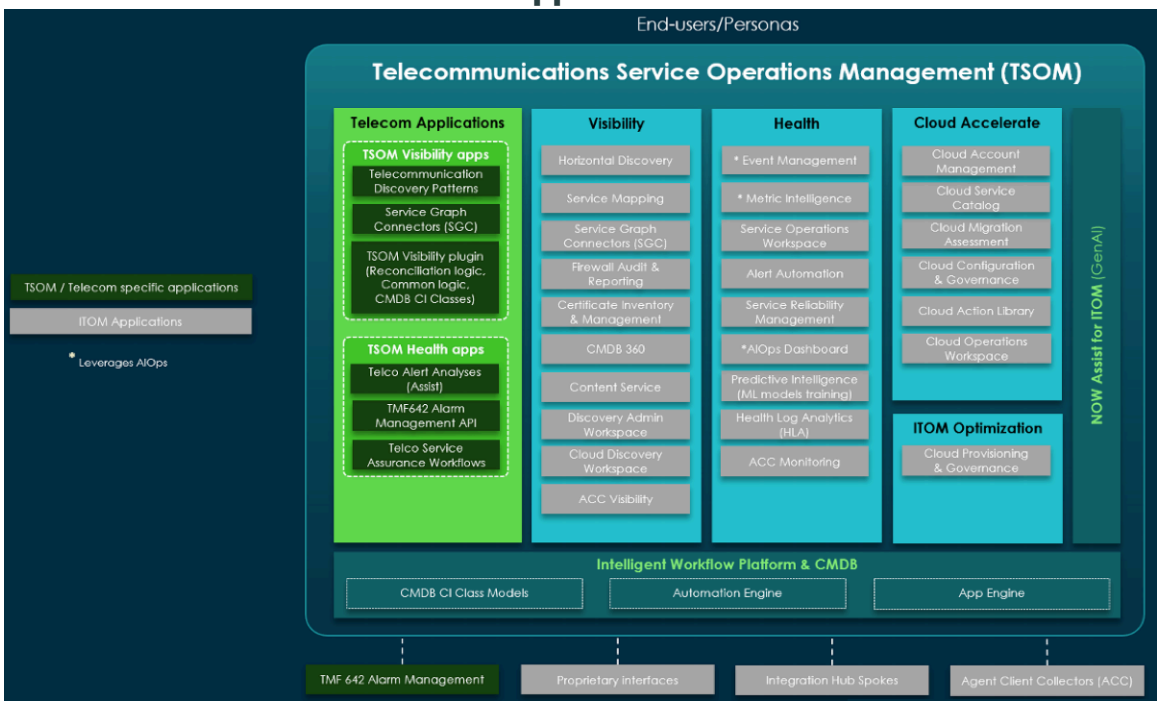
TSOM Visibility

The TSOM Visibility solution is built on the foundational principles of ITOM Visibility, leveraging its proven frameworks for discovering and mapping IT resources.

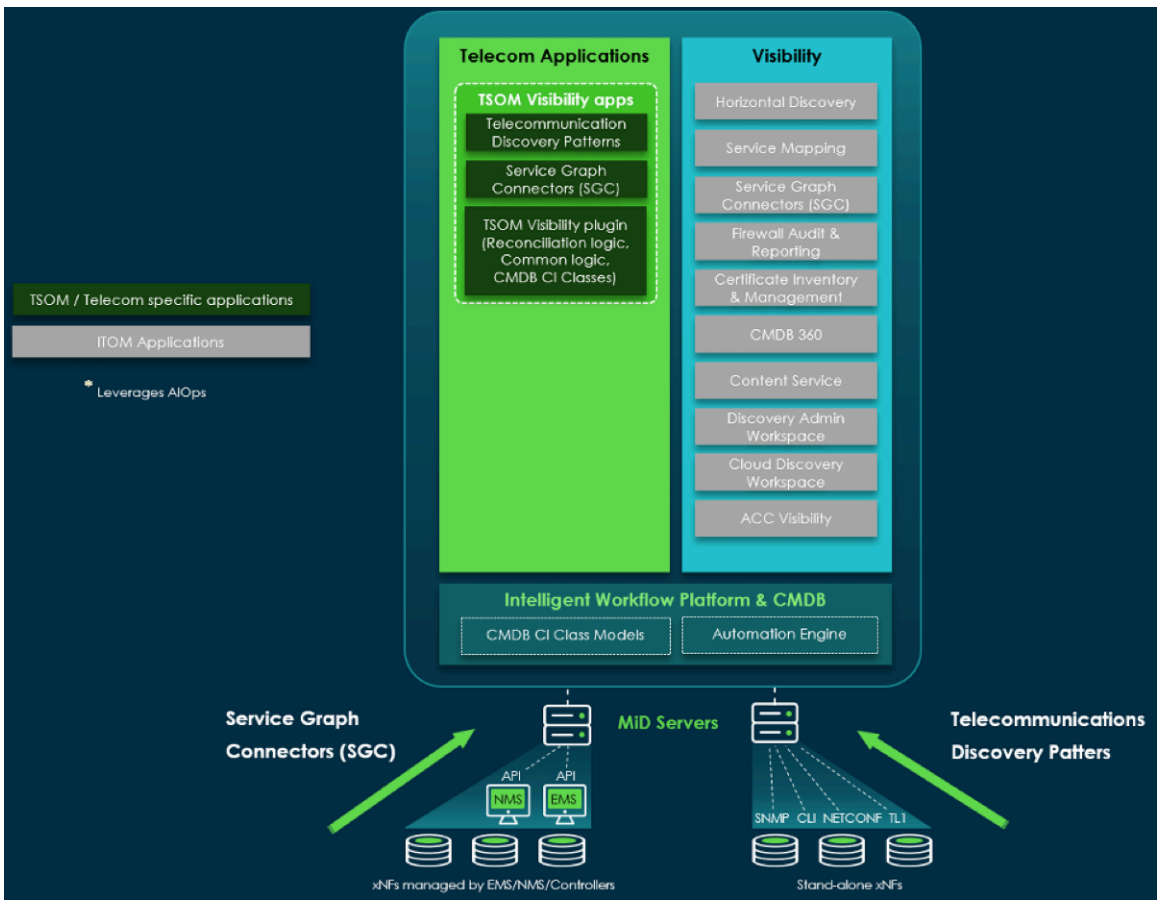
In addition to the core ITOM Visibility functionalities, TSOM Visibility introduces a unique telecom-specific layer with tailored applications. These include Telecom Discovery, which leverages the Horizontal Discovery and Telecommunications Discovery Patterns application, Service Graph Connectors (SGC), and Telecom Discrepancy Identification & Reconciliation, which incorporates telecom-specific logic introduced by the TSOM Visibility plugin. All components are designed to support the unique hierarchy and resource structures of the telecom data model. This combination enhances the discovery, reconciliation, and management of telecom infrastructure and services across complex multi-vendor, multi-technology networks, IT, and cloud environments.

Note: Additional TSOM applications and APIs related to TSOM Health and AIOps fall outside the scope of the TSOM Visibility solution and will be covered separately.

TSOM: Architecture and Telecom Applications



TSOM Visibility: Telecom Discovery and Discrepancy Identification and Reconciliation



Horizontal Discovery and Telecommunications Discovery Patterns

Horizontal Discovery is an automated process that continuously scans and identifies all the components within the network infrastructure. It plays a crucial role in maintaining an accurate and up-to-date CMDB and TNI with the information in the network.

In Telecom Discovery, we use Horizontal Discovery and Telecommunication (TSOM) Discovery Patterns (sn_tsom_patterns) to discover standalone xNFs using SNMP and CLI protocols.

For more information on the TSOM Patterns, see Telecom Discovery using Telecommunications Discovery Patterns.

For more information on the Horizontal Discovery application, see Horizontal discovery process flow with patterns.

Service Graph Connectors (SGC)

The Service Graph Connectors (SGC) are predefined integrations that ingest data into the Configuration Management Database (CMDB) from third-party sources. They can be used alongside any existing Service Graph Connectors, such as those for security, servers, software, monitoring, internet of Things (IoT), and cloud.

In Telecom Discovery, we use Service Graph Connectors to discover network resources via northbound APIs of EMS/NMS/Controllers, which manage various xNFs.

For more information on the Telecom SGSs, see Telecom Discovery using Service Graph Connectors (SGC).

TSOM Visibility Applications

- Service Graph Connectors (TSOM SGCs) –These new plugins enable the discovery of data from existing network management systems (for example, EMS/NMS/Controllers).
- ○ Plugin name: sn_sgc_altiplano_connector
  ○ Plugin description: Service Graph Connector for Nokia Altiplano

  See Telecom Discovery using Service Graph Connectors (SGC).

- Telecommunications Discovery Patterns (TSOM Patterns) – These new plugins include patterns for the discovery of standard SNMP-based standalone xNFs (for example, telecom routers/switches without a management system or those intended to be discovered directly by bypassing their management systems). They also include custom patterns for verifying specific Cisco and Juniper devices by leveraging their SNMP and CLI accessibility.
- ○ Plugin name: sn_tsom_patterns
  ○ Plugin description: Telecommunication Discovery Patterns

  See Telecom Discovery using Telecommunications Discovery Patterns.

- TSOM Visibility (plugin) – This plugin is an enabler of the TSOM Visibility applications. It contains logic that is common across our Telecom Discovery and Telecom Discrepancy Identification & Reconciliation solution. It includes telecom-specific discrepancy identification and remediation logic, along with other common logic that we are building or will build for use across the application.

- Plugin name: sn_tsom_core
- Plugin description: TSOM Visibility (plugin)

See Telecom Discrepancy Identification and Reconciliation.

CMDB 360

CMDB 360 retains a complete history of discovery sources and proposed values involved in updates to CI attributes. Use CMDB 360 data to track how the CMDB is populated by various discovery sources at the CI attribute level. You can also revert CI updates from a specific discovery source or recompute attribute values using updated reconciliation rules.

The CMDB 360 view provides aggregations and analyses of CMDB 360 data, which can be used to track activities and identify potential issues with discovery sources. Additionally, you can create custom queries, schedules, and reports to explore CMDB data.

For more information on CMDB 360, see CMDB 360/Multisource CMDB.

Discovery Admin Workspace

The Discovery Admin Workspace serves as a central location for monitoring, tracking, and completing discovery-related tasks. Experience a streamlined discovery process and greater efficiency with the integration of schedules, diagnostics, tuning, and more within this single workspace.

For more information on the Discovery Admin Workspace, see Discovery Admin Workspace.

Who uses TSOM Visibility

TSOM Visibility enables telecom operators and communication service providers (CSP) and providing platform as a service to discover their network resources.

The ServiceNow® Configuration Management Database (CMDB) and Telecom Network Inventory (TNI) aren't operational tools. They're strategic necessities in today's IT and telecom landscape. Maintaining an accurate and complete CMDB/TNI provides the foundation for delivering critical services and drives multiple outcomes important to telecom operations, such as order/service fulfillment, inventory/asset management, and assurance. Having an up-to-date CMDB/Network Inventory is crucial for enabling high levels of Autonomous Network Operations (ANO) and essential for real-time decision-making, automation, and closed-loop operations.

TSOM Visibility Installation Disclaimer

In order to support the TSOM Visibility solution, we have modified the CMDB CI Class Models store application, introducing updates to the IRE Identification Rules for the following telecom CIs:

- Interface Cards
- Slots
- Subslots
- Network Interfaces

TSOM Visibility requires CMDB CI Class Models Version 1.69.0 (sn_cmdb_ci_class).

If you install any of the TSOM Visibility applications (sn_sgc_altiplano_connector, sn_tsom_patterns, or the sn_tsom_core plugin), the CMDB CI Class Models store application is automatically updated (or installed) to Version 1.69.0.

Note: An administrator can still upgrade the CMDB CI Class Models store application to Version 1.69.0 at their discretion, regardless of whether their Yokohama instance has TSOM Visibility or even if their instance is on a pre-Yokohama release (for example, Washington DC or Xanadu).

IMPORTANT! If an administrator deploys CMDB CI Class Models Version 1.69.0—whether or not TSOM Visibility is installed—any customized IRE identification rules applied to one or more of the above-mentioned telecom CIs may be affected. These rules will require careful validation to ensure proper functionality.

TSOM Visibility Licensing

The ServiceNow AI Platform uses a licensing model in which your organization is billed for the use of TSOM Visibility applications. Telecom Discovery, Telecom Discrepancy Identification & Reconciliation and TSOM Visibility (plugin) are components of TSOM Visibility. To use TSOM Visibility, your organization must subscribe to TSOM.

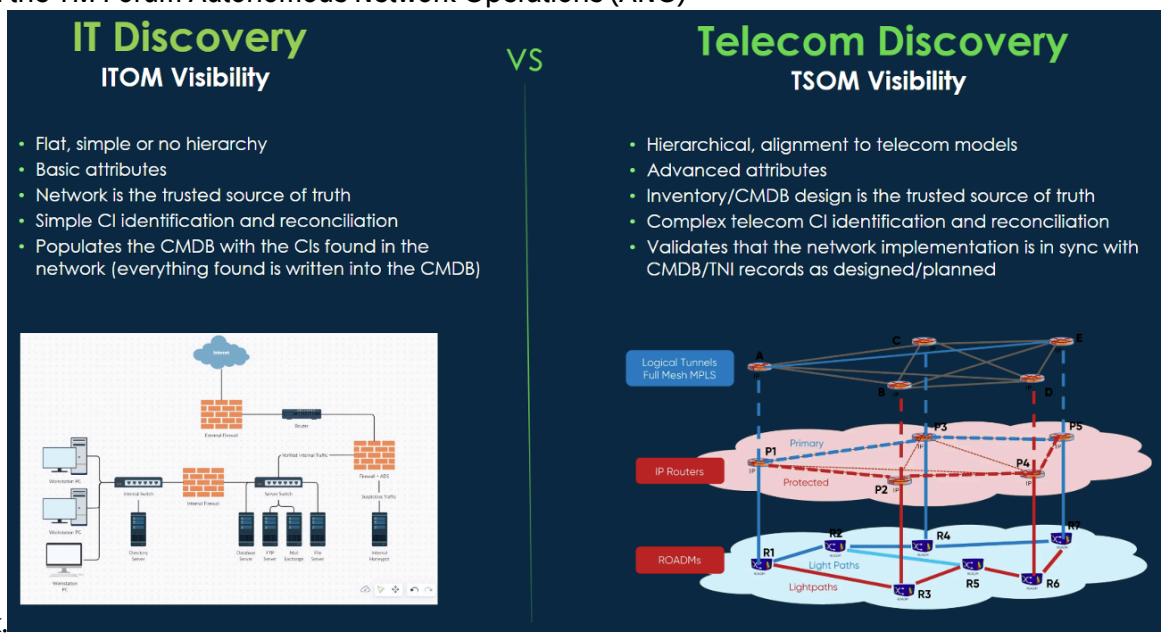
Note: ServiceNow's product documentation doesn't include details on pricing, packaging, or other specifics, as these are determined by your organization's customer contract.

Telecom Discovery

ServiceNow® Telecom Discovery (also known as TSOM Discovery) is a specialized solution within the broader ServiceNow ecosystem, designed to cater to the unique needs of telecommunications service providers (CSPs) in discovering and managing their telecom network assets.

This solution provides visibility into complex telecom infrastructures, enabling CSPs to automatically identify, map, and manage various network devices, services, and configurations across their multivendor environments, and update records in the CMDB/TNI with a real-time snapshot of network resources.

This capability is one of the key enablers for automation and aligns with the TM Forum Autonomous Network Operations (ANO) framework.



TSOM and ITOM Discovery can work together to provide holistic visibility and management across Telecom and IT environments.

For additional information on Telecom Discovery, refer to the following links:

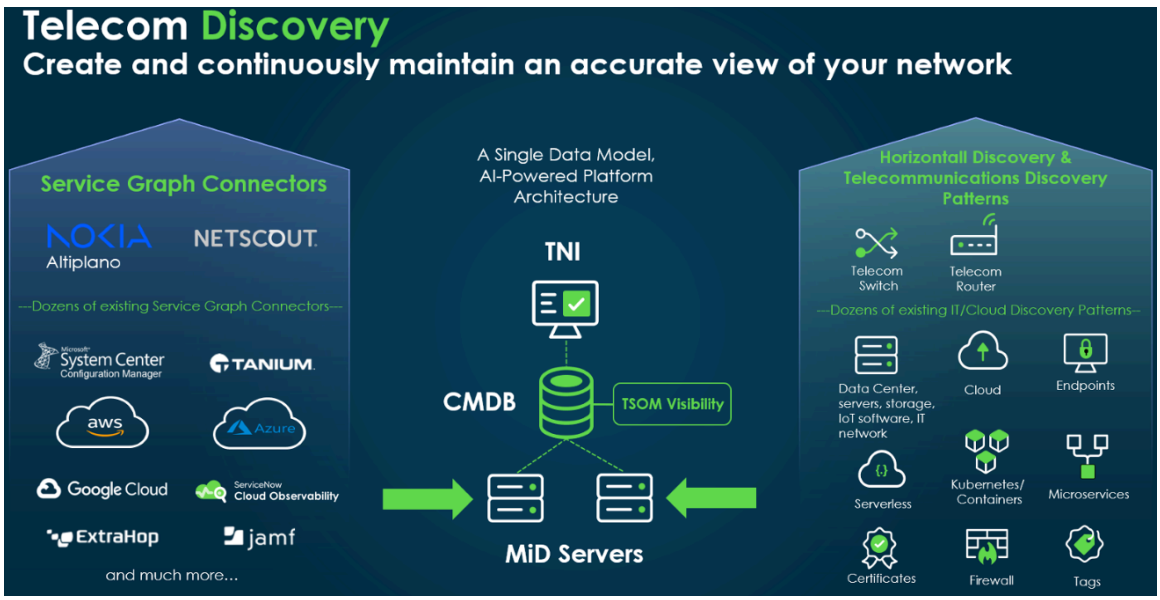
- Exploring Telecom Discovery
- Telecom Discovery using Service Graph Connectors (SGC)
- Telecom Discovery using Telecommunications Discovery Patterns

Telecom Discovery Licensing

The ServiceNow AI Platform uses a licensing model in which your organization is billed for the use of TSOM Visibility applications. Telecom Discovery, Telecom Discrepancy Identification & Reconciliation and TSOM Visibility (plugin) are components of TSOM Visibility. To use TSOM Visibility, your organization must subscribe to TSOM. Please note that ServiceNow® product documentation doesn't include details on pricing, packaging, or other specifics, as these are determined by your organization's customer contract.

Exploring Telecom Discovery

ServiceNow® Telecom Discovery solution builds on proven technologies within the ServiceNow® platform, leveraging powerful capabilities such as ITOM Visibility and Horizontal Discovery, and the Service Graph Connectors. Each of these components plays a key role in supporting telecom-specific use cases while also maintaining flexibility for general IT/Cloud discovery.



The TSOM Visibility Plugins can be seamlessly used alongside the ITOM Visibility to enhance infrastructure discovery and visibility. By integrating these plugins, organizations can leverage TSOMs advanced Telecommunications Discovery Patterns and Service Graph Connectors (SGC) with ITOM's capabilities, enabling comprehensive identification and mapping of network components, services, and configurations. This combined approach confirms an accurate and enriched data population within the CMDB, providing a unified and detailed view of IT and telecom network resources.

TSOM Visibility Plugins:

Service Graph Connectors (SGC)

This new plugin enables the discovery of data from existing network management systems (for example, EMS/NMS/Controllers), with a focus in this release on integrating Nokia Altiplano Access SDN Control via REST API. It's a customer-visible plugin and must be installed individually.

Plugin name: sn_sgc_altiplano_connector.

Plugin description: Service Graph Connector for Nokia Altiplano.

For more information, see Telecom Discovery using Service Graph Connectors (SGC).

Telecommunications Discovery Patterns (TSOM Patterns)

This new plugin includes patterns for the discovery of standard SNMP-based, standalone xNFs (for example, telecom routers and switches without a management system or those we want to discover directly by bypassing their management systems), as well as custom patterns for verifying specific Cisco and Juniper devices by leveraging their SNMP and CLI accessibility.

Plugin name: sn_tsom_patterns.

Plugin description: Telecommunication Discovery Patterns.

For more information, see Telecom Discovery using Telecommunications Discovery Patterns.

TSOM Visibility plugin

This plugin contains logic that is common across our TSOM Visibility application. It includes telecom-specific discrepancy and remediation logic, along with other common logic that we are building or will build for use across the application.

Plugin name: sn_tsom_core.

Plugin description: Telecom Service Operations Core.

For more information, see Telecom Discrepancy Identification and Reconciliation.

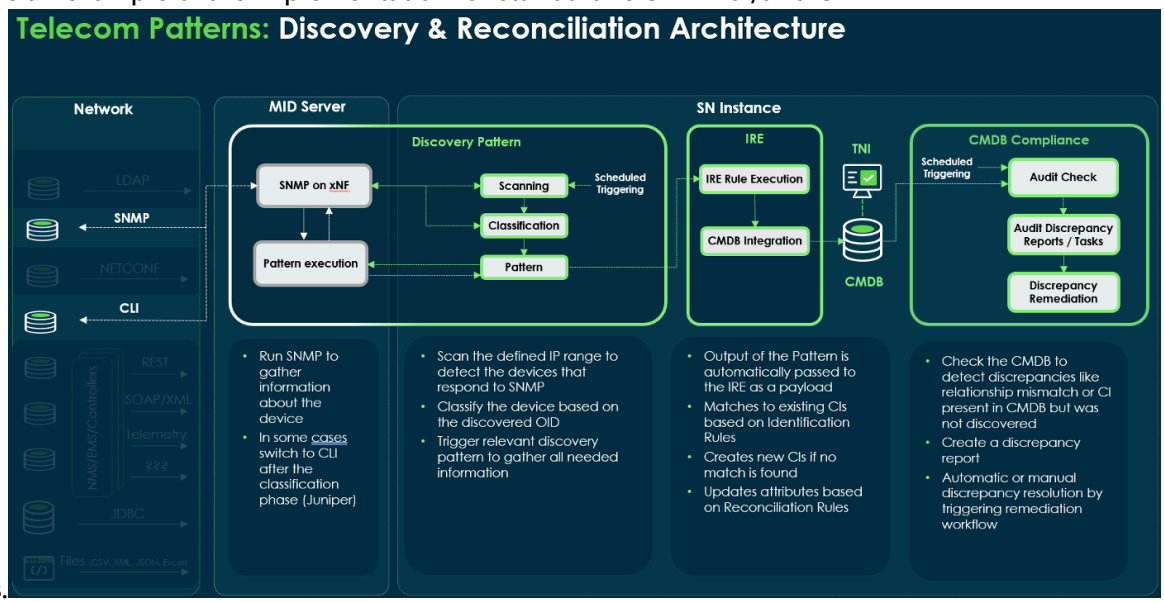
ServiceNow provides customers and partners with the capability to create and modify Service Graph Connectors and Telecommunications Discovery Patterns. To simplify this process, ServiceNow has developed a set of **no-code/low-code UI-based design tools**. These tools enable users to customize and extend Service Graph Connectors and Patterns without needing deep coding expertise, confirming flexibility, and faster deployment of tailored solutions for specific network environments.

Telecom Discovery using Telecommunications Discovery Patterns

The new Telecommunication Discovery Patterns (also known as TSOM Patterns) include patterns for discovering standalone xNFs, enabling the discovery of standalone network elements without a management system, using SNMP, CLI, and NETCONF (roadmap) protocols.

Architecture using Horizontal Discovery and Telecommunications Discovery Patterns

This is an example of the implementation for standalone SNMP or/and CLI xNFs.



Horizontal Discovery Application

The Horizontal Discovery application in ServiceNow is a versatile and highly scalable discovery engine designed to operate effectively across network, IT, and cloud environments, collecting data across multiple layers to provide a holistic view of the infrastructure.

For more information, see Horizontal discovery process flow with patterns.

Telecommunications Discovery Patterns (TSOM Patterns)

A pattern is a sequence of commands designed to detect attributes of a configuration item (CI) and its outbound connections. Telecom Discovery provides a set of preconfigured Patterns that cover a wide range of network elements. The TSOM Discovery Patterns fall under the infrastructure category, which are used by Horizontal Discovery to generate lists and resource structure of xNFs.

This plugin introduces Patterns for discovering standalone xNFs, such as telecom routers and switches (with support for other device types planned in the future) that don't rely on a management system or require direct discovery by bypassing their management systems. Additionally, custom patterns are included for verifying various network devices. This is a customer-visible plugin.

MID Server

MID Server is a Java application that runs as a Windows service or UNIX daemon on a server within your local network. The ServiceNow® MID Server facilitates communication and data transfer between a ServiceNow instance and external applications, data sources, and services.

For more information, see MID Server.

Identification & Reconciliation Engine (IRE)

IRE offers a centralized framework for identifying and reconciling data from multiple sources. It verifies the integrity of the CMDB and some non-CMDB tables when various data sources are used to create or update CI records.

For more information, see Telecom Discrepancy Identification and Reconciliation.

CMDB Compliance Certification Audits for Telecom Discrepancy Identification & Reconciliation

CMDB Compliance is a toolset that enables administrators to certify CMDB data for accuracy and resolve any discrepancies found. In Telecom Discrepancy Identification & Reconciliation, we use the Certification Audits feature to discover and analyze discrepancies in the CMDB, generate Certification Follow-on Tasks, and enable remediation workflows.

For more information on how it's used for Discrepancy Identification & Reconciliation, see Telecom Discrepancy Identification and Reconciliation.

ITOM vs TSOM Discovery Behavior for Stand-alone SNMP or/and CLI xNFs

ITOM Discovery	TSOM Discovery (Telecom)
Based on Horizontal Discovery Patterns.	Based on Horizontal Discovery Patterns (for Telecommunications).
Discover basic equipment flat information which is primarily used for the IT purposes.	Discover more attributes and a deeper CI hierarchy based on equipment type.
Only standard MIBs are supported.	Both standard and vendor proprietary MIBs are supported.
No discrepancy detection and remediation available OOB.	Discrepancy detection and automatic or manual remediation available OOB.

- Telecom Discovery is built on the ITOM Discovery application, leveraging the Nebula Discovery Language (NDL).
- The solution includes a set of telecom-specific patterns and system properties.
- Users can choose whether to run TSOM or ITOM patterns.
- Customers can extend TSOM patterns or create their own if necessary.
- There's no dependency between TSOM Visibility and Telecom Network Inventory (TNI).

Logic added to Telecom Discovery

Users can define whether they want to use Telecommunications Discovery Patterns (also known as TSOM patterns) with ITOM patterns for execution. By default, all patterns created in TSOM/Telecom Discovery execute the original ITOM pattern (for example, the 'Telco Router' pattern also executes the 'Router' pattern). However, customers have the option to choose whether they want to execute only the TSOM pattern and exclude the ITOM pattern.

This parameter affects only the patterns developed as part of the TSOM Patterns application for Telco customers. If the property is set to true (default), the TSOM pattern will also execute specific ITOM shared libraries from the TSOM pattern.

This logic is controlled via the system property: sn_tsom_patterns.itom_pattern_enabled.

Follows the TNI model – Telecommunications Discovery Patterns always follow the TNI data model. This means that if Telecom Discovery identifies a card-on-card scenario, it doesn't insert it as a card-on-card. Instead, it modifies the structure by synthesizing a subslot on the parent card and inserting the child card into that subslot.

TNI entity creation logic – Whenever the system identifies that the customer has TNI installed, it will automatically create a TNI entity record for all network data discovered. If TNI is installed, a payload like the one below will be added to the IRE payload for each item. As a result, the discovered CI is in both the cmdb_ci and tni_entity tables.

Telecom Router Pattern

The ServiceNow® Telecom Discovery application uses the Telecom Router discovery pattern to find SNMP-based routers in the network. Discovering some of these resources requires updating the Telecommunications Discovery Patterns (TSOM Patterns) from the ServiceNow Store.

Telecom Discovery uses the Telecommunications Discovery Patterns to run Horizontal Discovery. This Telecommunications Discovery Pattern uses a set of SNMP requests to find, classify, and discover network elements.

Telecom Router pattern is part of the Telecommunications Discovery Patterns application (sn_tsom_patterns), which is part of TSOM Visibility.

Request apps on the Store

Visit the ServiceNow Store website 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.

Prerequisites

- Subscription to TSOM.
- Confirm that your network router devices have SNMP access.
- On the ServiceNow instance, configure SNMP credentials. For more information, see SNMP support for Discovery.
- For setting up TSOM Patterns, see Configure Telecommunications Discovery (TSOM) Patterns.

Impacted CMDB CIs and CI Relationships (Physical Layer)

CIs	CI Relationships
IP Router CI	IP Router Device is represented by the IP Router CI. Table name: **cmdb_ci_ip_router** IP Router CI contains Slots or Network Interfaces.
Slot CI	Slots are represented by the Slot CI. Table name: **cmdb_ci_container_slot** Slot is contained by the IP Router.

CIs	CI Relationships
	Slot CI contains the Interface Card.
Subslot CI	Subslots are represented by the Subslot CI. Table name: **cmdb_ci_container_subslot** Subslot is contained by Interface Card CI. Subslot CI contains the Interface Card CI.
Interface Card CI	Different types of cards are represented by the Interface Card CI. Fan and Power Supply Units are also represented by the Card CI. Table name: **cmdb_ci_interface_card** Interface Cards are contained by Slots or Subslots. Interface Cards can contain Network Interface or Subslots.
Network Interface CI	Any type of Network Interface is represented by the Network Interface CI. Table name: **cmdb_ci_ni_interface** Network Interface is contained by Interface Card, IP Router.

Classifying xNFs for this Pattern

To access a full list of OIDs that will be classified.

Before you begin

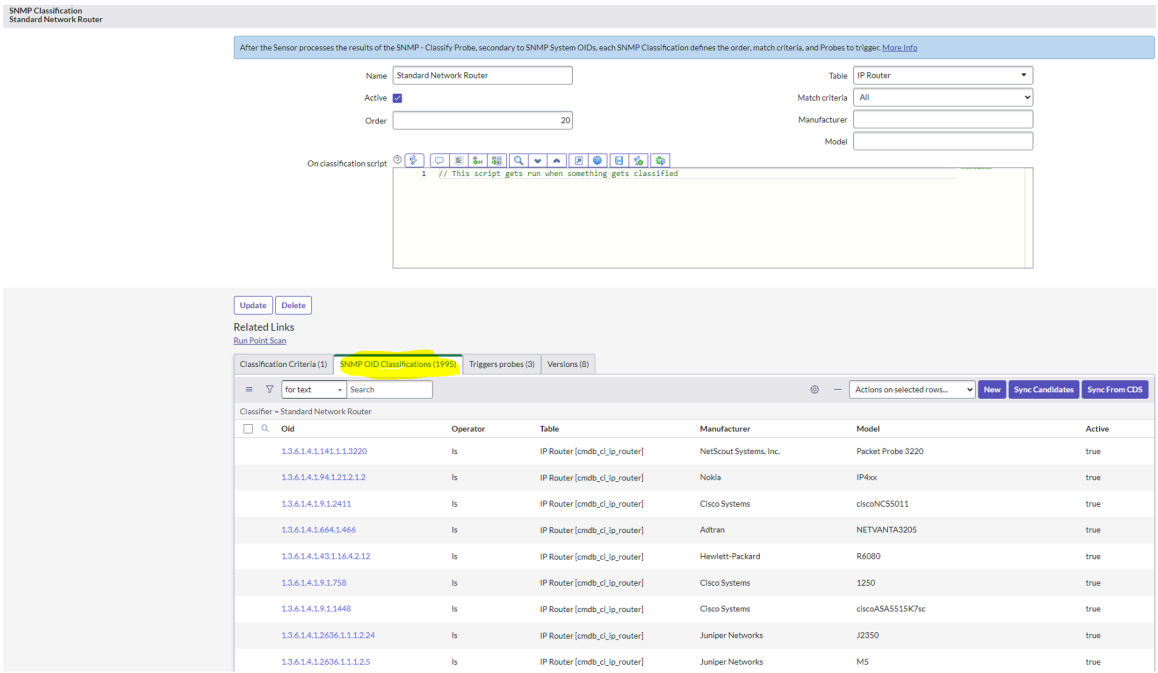
Role required: admin

Classifier name: **Standard Network Router**.

Procedure

1. Navigate to **All** > **Discovery Definition** > **CI Classification** > **SNMP**.
2. From the list, select **Standard Network Router**.

3. Open the tab **SNMP OID Classifications** and see the list of OIDs.



Note:

For more information on how to add additional OIDs to the classifier, see Telecom Discovery using Telecommunications Discovery Patterns.

MiB Tables Used on an xNF:

- SystemMIB
- EntityPhysicalMIB
- IfMIB
- IfXMIB
- IpMIB

Telecom Cisco 7613 Router Pattern

The ServiceNow Telecom Discovery application uses the Telecom Cisco 7613 Router discovery pattern to find SNMP-based Cisco 7613 in the network. Discovering some of these resources requires updating the Telecommunications Discovery Patterns (TSOM Patterns) from the ServiceNow Store.

Telecom Discovery uses the Telecommunications Discovery Patterns to run Horizontal Discovery. This Telecommunications Discovery Pattern uses a set of SNMP requests to find, classify, and discover network elements.

Telecom Cisco 7613 Router Pattern is part of the Telecommunications Discovery Patterns application (sn_tsom_patterns), which is part of TSOM Visibility.

Request apps on the Store

Visit the ServiceNow Store website 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.

Prerequisites

- Subscription to TSOM.
- Verify that your network router devices have SNMP access.
- On the ServiceNow instance, configure SNMP credentials. For more information, see SNMP support for Discovery.
- For setting up TSOM Patterns, see Configure Telecommunications Discovery (TSOM) Patterns.

Impacted CMDB CIs and CI Relationships (Physical Layer)

CIs	CI Relationships
IP Router CI	IP Router Device is represented by the IP Router CI. Table name: **cmdb_ci_ip_router** IP Router CI contains Slots or Network Interfaces.
Slot CI	Slots are represented by the Slot CI. Table name: **cmdb_ci_container_slot** Slot is contained by the IP Router. Slot CI contains the Interface Card.
Subslot CI	Subslots are represented by the Subslot CI. Table name: **cmdb_ci_container_subslot** Subslot is contained by Interface Card CI. Subslot CI contains the Interface Card CI.
Interface Card CI	Different types of cards are represented by the Interface Card CI. Fan and Power Supply Units are also represented by the Card CI. Table name: **cmdb_ci_interface_card** Interface Cards are contained by Slots or Subslots. Interface Cards can contain Network Interface or Subslots.
Network Interface CI	Any type of Network Interface is represented by the Network Interface CI.

CIs	CI Relationships
	Table name: **cmdb_ci_ni_interface** Network Interface is contained by Interface Card, IP Router.

Classifying xNFs for this Pattern

To access a full list of OIDs that will be classified.

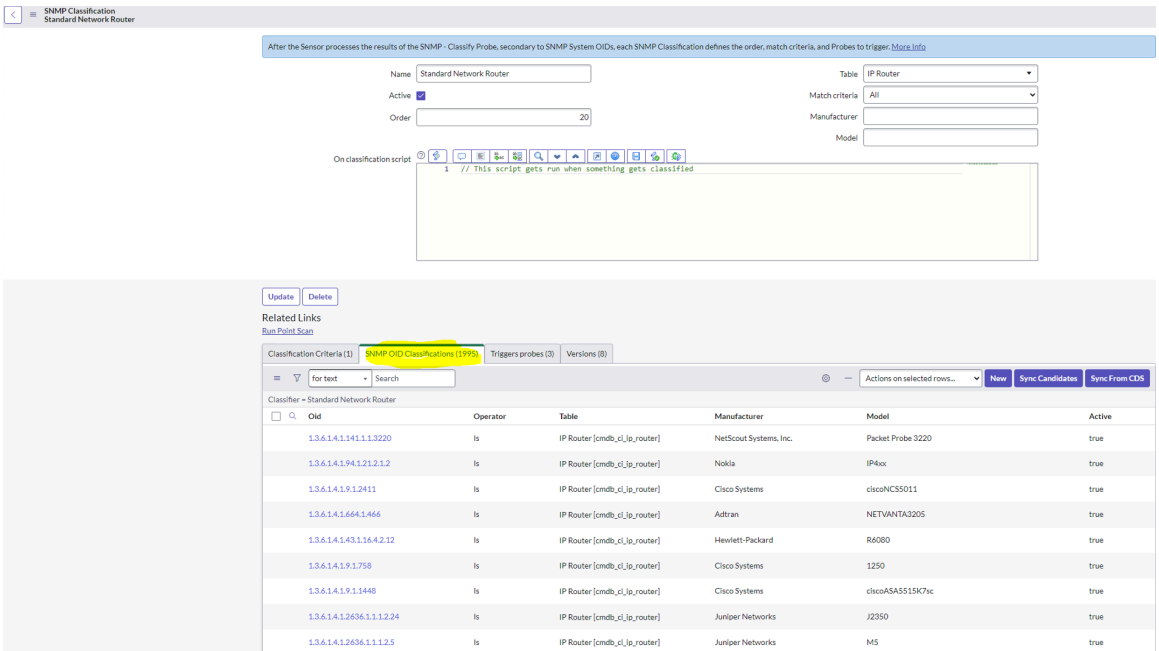
Before you begin

Role required: admin

Classifier name: **Standard Network Router**.

Procedure

1. Navigate to **All** > **Discovery Definition** > **CI Classification** > **SNMP**.
2. From the list, select **Standard Network Router**.
3. Open the tab **SNMP OID Classifications** and see the list of OIDs.



Note:

For more information on how to add additional OIDs to the classifier, see Telecom Discovery using Telecommunications Discovery Patterns.

The List of Specific OIDs to call this Pattern:

Vendor	Model	OID	Pattern
Cisco	7613	1.3.6.1.4.1.9.1.528	Telecom Cisco 7613 Router

MiB Tables Used on an xNF:

- SystemMIB
- EntityPhysicalMIB

- IfMIB
- IfXMIB
- IpMIB

Telecom Juniper MX SSH Router Pattern

The ServiceNow® Telecom Discovery application uses the Telecom Juniper MX SSHRouter discovery pattern to find SNMP and CLI -based Juniper MX Series routers in the network. Discovering some of these resources requires updating the Telecommunications Discovery Patterns (TSOM Patterns) from the ServiceNow® store.

Telecom Discovery uses the Telecommunications Discovery Patterns to run Horizontal Discovery. This Telecommunications Discovery Pattern uses a set of SNMP requests to find and classify CLI over SSH to discover network elements.

Telecom Juniper MX SSH Router pattern is part of the Telecommunications Discovery Patterns application (sn_tsom_patterns), which is part of TSOM Visibility.

Request apps on the Store

Visit the ServiceNow Store website 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.

Prerequisites

- Subscription to TSOM.
- Verify that your network router devices have SNMP access.
- On the ServiceNow instance, configure SNMP credentials. For more information, see SNMP support for Discovery.
- For setting up TSOM Patterns, see Configure Telecommunications Discovery (TSOM) Patterns.

Impacted CMDB CIs and CI Relationships (Physical Layer)

CIs	CI Relationships
IP Router CI	IP Router Device is represented by the IP Router CI. Table name: **cmdb_ci_ip_router** IP Router CI contains Slots or Network Interfaces.
Slot CI	Slots are represented by the Slot CI. Table name: **cmdb_ci_container_slot** Slot is contained by the IP Router. Slot CI contains the Interface Card.

CIs	CI Relationships
Subslot CI	Subslots are represented by the Subslot CI. Table name: **cmdb_ci_container_subslot** Subslot is contained by Interface Card CI. Subslot CI contains the Interface Card CI.
Interface Card CI	Different types of cards are represented by the Interface Card CI. Fan and Power Supply Units are also represented by the Card CI. Table name: **cmdb_ci_interface_card** Interface Cards are contained by Slots or Subslots. Interface Cards can contain Network Interface or Subslots.
Network Interface CI	Any type of Network Interface is represented by the Network Interface CI. Table name: **cmdb_ci_ni_interface** Network Interface is contained by Interface Card, IP Router.

Classifying xNFs for this Pattern

To access a full list of OIDs that will be classified.

Before you begin

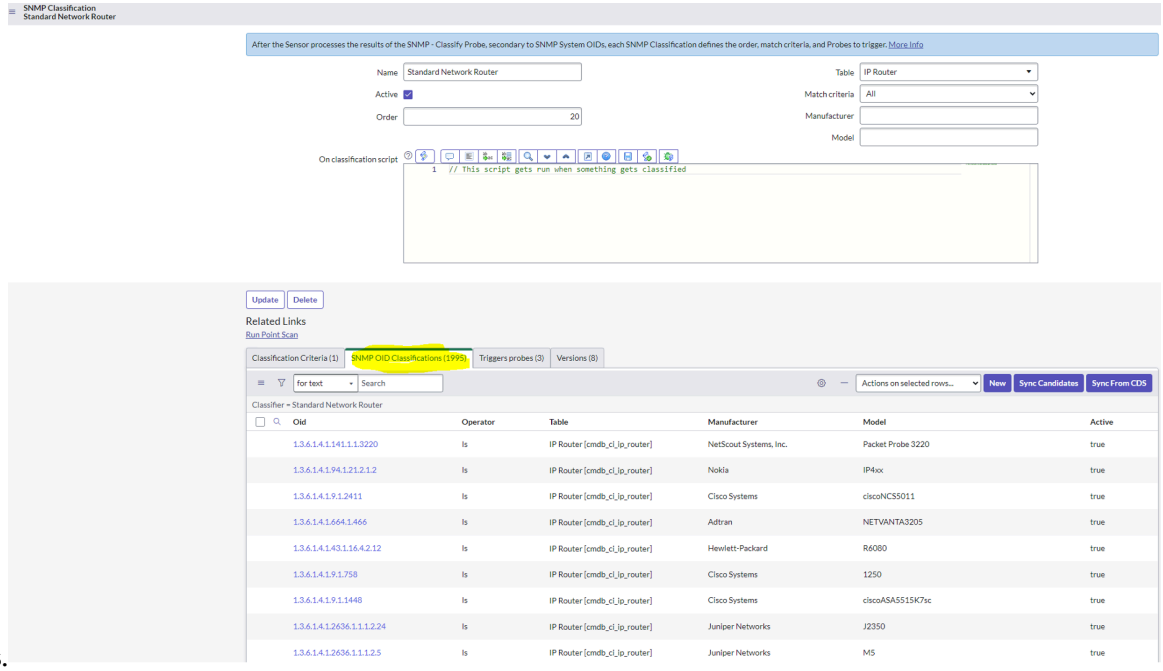
Role required: admin

Classifier name: **Standard Network Router**.

Procedure

1. Navigate to **All** > **Discovery Definition** > **CI Classification** > **SNMP**.
2. From the list, select **Standard Network Router**.

3. Open the tab **SNMP OID Classifications** and see the list of OIDs.



Note:

For more information on how to add additional OIDs to the classifier, see Telecom Discovery using Telecommunications Discovery Patterns.

The List of Specific OIDs to call this Pattern:

Vendor	Model	OID	Pattern
Juniper	MX80	1.3.6.1.4.1.2636.1.1.1.2.57	Telecom Juniper MX SSH Router
Juniper	MX104	1.3.6.1.4.1.2636.1.1.1.2.97	Telecom Juniper MX SSH Router
Juniper	MX240	1.3.6.1.4.1.2636.1.1.1.2.29	Telecom Juniper MX SSH Router
Juniper	MX480	1.3.6.1.4.1.2636.1.1.1.2.25	Telecom Juniper MX SSH Router

MiB Tables Used on an xNF: SystemMIB.

CLI Commands Used.

- show chassis hardware | no-more | display xml
- show interface media | no-more | display xml

Telecom Cisco Switch Pattern

The ServiceNow® Telecom Discovery application uses the Telecom Cisco Switch discovery pattern to find SNMP-based Cisco switches in the network. Discovering some of these resources requires updating the Telecommunications Discovery Patterns (TSOM Patterns) from the ServiceNow® Store.

Telecom Discovery uses the Telecommunications Discovery Patterns to run Horizontal Discovery. This Telecommunications Discovery Pattern uses a set of SNMP requests to find, classify, and discover network elements.

Telecom Cisco Switch pattern is part of the Telecommunications Discovery Patterns application (sn_tsom_patterns), which is part of TSOM Visibility.

Request apps on the Store

Visit the ServiceNow Store website 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.

Prerequisites

- Subscription to TSOM.
- Confirm that your network router devices have SNMP access.
- On the ServiceNow® instance, configure SNMP credentials. For more information, see SNMP support for Discovery.
- For setting up TSOM Patterns, see Configure Telecommunications Discovery (TSOM) Patterns.

Impacted CMDB CIs and CI Relationships (Physical Layer)

CIs	CI Relationships
IP Switch CI	IP Switch Device is represented by the IP Switch CI. Table name: **cmdb_ci_ip_switch** IP Switch CI contain Slots or Network Interfaces.
Slot CI	Slots are represented by the Slot CI. Table name: **cmdb_ci_container_slot** Slot is contained by the IP Switch. Slot CI contains the Interface Card.
Subslot CI	Subslots are represented by the Subslot CI. Table name: **cmdb_ci_container_subslot** Subslot is contained by Interface Card CI. Subslot CI contains the Interface Card CI.
Interface Card CI	Different types of cards are represented by the Interface Card CI. Fan and Power Supply Units are also represented by the Card CI. Table name: **cmdb_ci_interface_card**

CIs	CI Relationships
	Interface Cards are contained by Slots or Subslots. Interface Cards can contain Network Interface or Subslots.
Network Interface CI	Any type of Network Interface is represented by the Network Interface CI. Table name: **cmdb_ci_ni_interface** Network Interface is contained by Interface Card, IP Switch.

Classifying xNFs for this Pattern

To access a full list of OIDs that will be classified.

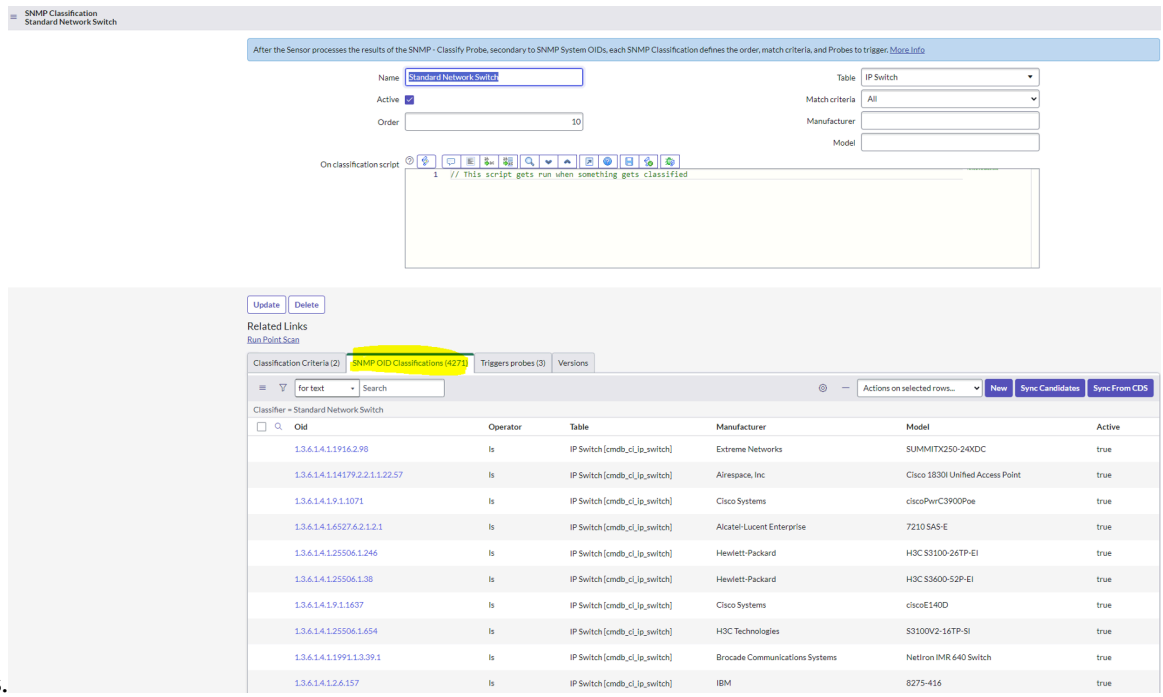
Before you begin

Role required: admin

Classifier name: **Standard Network Switch**.

Procedure

1. Navigate to **All** > **Discovery Definition** > **CI Classification** > **SNMP**.
2. From the list, select **Standard Network Switch**.
3. Open the tab **SNMP OID Classifications** and see the list of OIDs.



Note:

For more information on how to add additional OIDs to the classifier, see Telecom Discovery using Telecommunications Discovery Patterns.

The List of Specific OIDs to call this Pattern:

Vendor	Model	OID	Pattern
Cisco	Nexus 9000	1.3.6.1.4.1.9.12.3.1.3.1954	Telecom Cisco Switch
Cisco	Nexus 3548	1.3.6.1.4.1.9.12.3.1.3.1666	Telecom Cisco Switch

MiB Tables Used on an xNF:

- SystemMIB
- EntityPhysicalMIB
- IfMIB
- IfXMIB
- IpMIB

Telecom Switch Pattern

The ServiceNow® Telecom Discovery application uses the Telecom Switch discovery pattern to find SNMP-based Telecom switches in the network. Discovering some of these resources requires updating the Telecommunications Discovery Patterns (TSOM Patterns) from the ServiceNow® Store.

Telecom Discovery uses the Telecommunications Discovery Patterns to run Horizontal Discovery. This Telecommunications Discovery Pattern uses a set of SNMP requests to find, classify, and discover network elements.

Telecom Switch pattern is part of the Telecommunications Discovery Patterns application (sn_tsom_patterns), which is part of TSOM Visibility.

Request apps on the Store

Visit the ServiceNow Store website 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.

Prerequisites

- Subscription to TSOM.
- Confirm that your network router devices have SNMP access.
- On the ServiceNow® instance, configure SNMP credentials. For more information, see SNMP support for Discovery.
- For setting up TSOM Patterns, see Configure Telecommunications Discovery (TSOM) Patterns.

Impacted CMDB CIs and CI Relationships (Physical Layer)

CIs	CI Relationships
IP Switch CI	IP Switch Device is represented by the IP Switch CI. Table name: **cmdb_ci_ip_switch**

CIs	CI Relationships
	IP Switch CI contain Slots or Network Interfaces.
Slot CI	Slots are represented by the Slot CI. Table name: **cmdb_ci_container_slot** Slot is contained by the IP Switch. Slot CI contains the Interface Card.
Subslot CI	Subslots are represented by the Subslot CI. Table name: **cmdb_ci_container_subslot** Subslot is contained by Interface Card CI. Subslot CI contains the Interface Card CI.
Interface Card CI	Different types of cards are represented by the Interface Card CI. Fan and Power Supply Units are also represented by the Card CI. Table name: **cmdb_ci_interface_card** Interface Cards are contained by Slots or Subslots. Interface Cards can contain Network Interface or Subslots.
Network Interface CI	Any type of Network Interface is represented by the Network Interface CI. Table name: **cmdb_ci_ni_interface** Network Interface is contained by Interface Card, IP Switch.

Classifying xNFs for this Pattern

To access a full list of OIDs that will be classified.

Before you begin

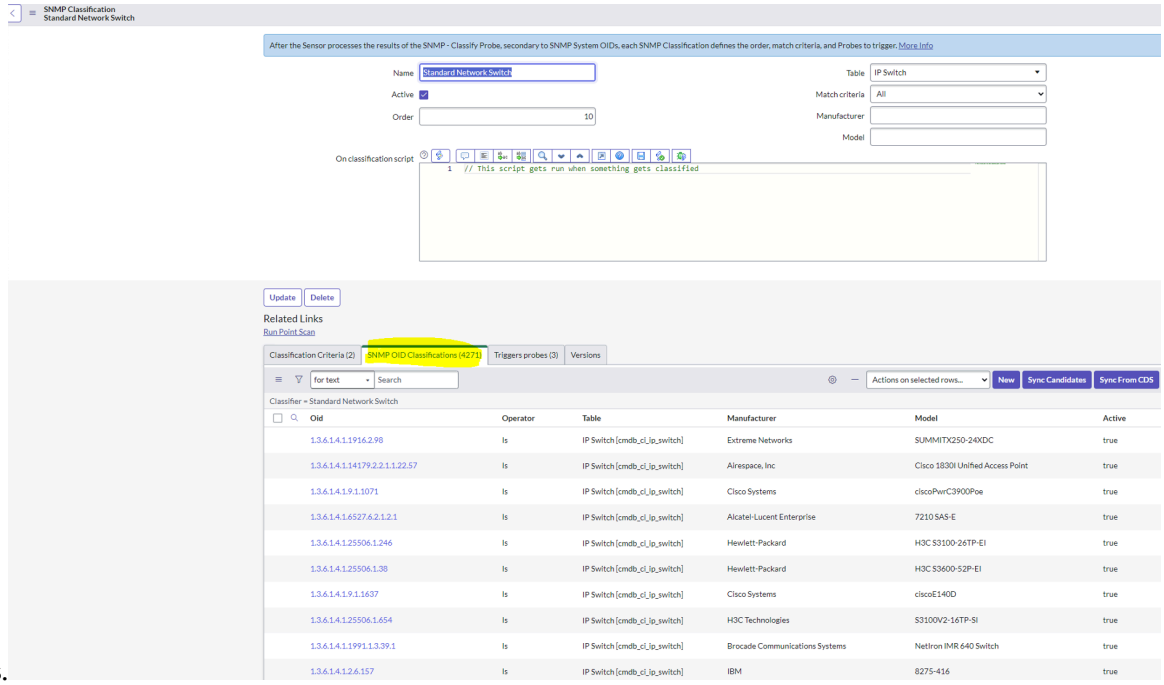
Role required: admin

Classifier name: **Standard Network Switch**.

Procedure

1. Navigate to **All** > **Discovery Definition** > **CI Classification** > **SNMP**.
2. From the list, select **Standard Network Switch**.

3. Open the tab **SNMP OID Classifications** and see the list of OIDs.



Note:

For more information on how to add additional OIDs to the classifier, see Telecom Discovery using Telecommunications Discovery Patterns.

MiB Tables Used on an xNF:

- SystemMIB
- EntityPhysicalMIB
- IfMIB
- IfXMIB
- IpMIB

Configure Telecommunications Discovery (TSOM) Patterns

This document outlines the dependencies, requirements, and installation steps necessary for setting up Telecommunications Discovery Patterns (also known as TSOM Patterns) in TSOM Visibility in ServiceNow®.

Before you begin

To use Telecommunications Discovery Patterns, you need a subscription to TSOM.

Role required: admin

Request apps on the Store

Visit the ServiceNow Store website 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.

Dependencies and Requirements

- Telecom Service Operation Core (sn_tsom_core)
- Discovery Core plugin (com.snc.discovery.core), which is automatically installed by Discovery.
- ITOM Discovery License plugin (com.snc.itom.discovery.license). You must activate this plugin.
- ITOM Licensing plugin (com.snc.itom.license). For more information, see Request Discovery.

TSOM Visibility Installation Disclaimer

See TSOM Visibility Installation Disclaimer for important information and requirements related to the installation process.

Install Horizontal Discovery and Telecommunication Discovery Patterns

The process of obtaining and installing the Telecommunications Discovery Patterns in TSOM.

Before you begin
Role required: admin

Procedure

1. Install the Horizontal Discovery application.
   See Discovery setup, as it is foundational for running Telecommunications Discovery Patterns.
2. Obtain and install Telecommunications Discovery Patterns:
   a. Install the Telecommunications Discovery Patterns (sn_tsom_patterns) from the ServiceNow® Store.
3. Set up a MID Server and synchronization Patterns:
   a. Synchronization the installed patterns with the appropriate MID Servers to confirm they're ready for use:
      i. Navigate to **Discovery** > **MID Servers**.
      ii. Select **Pattern Sync to Mid**.

      Note: This action synchronizes both TSOM and ITOM patterns.

      For more information on how to configure a MID Server, see Configuring MID Server.
4. Configure TSOM System Properties:
   a. Set the system property sn_tsom_patterns.itom_pattern_enabled to define the logic for whether to use only the TSOM Pattern or a combination of ITOM and TSOM patterns.
      i. Navigate to**All** > **System Properties** > **All Properties**.
      ii. Select **sn_tsom_patterns.itom_pattern_enabled**.
      iii. Check that the Value is set to **true** (default).

      If you want TSOM to run only TSOM patterns and exclude ITOM patterns, set the Value to **false**.

      Note: The default setting is configured to use both TSOM and ITOM patterns.

5. Enable the replacement of various ITOM patterns with TSOM patters on a specific MID Server:

   For example: The Telecom Router pattern replaces the Network Router pattern for a specific MID Server when **mid.telecom.discovery.patterns.enabled** is set to true for that MID Server.

   a. Go to the **Filter Navigator** and type **ecc_agent_config.list**.

   b. Select **mid.telecom.discovery.patterns.enabled** (each MID Server has this parameter).

   c. Check that the Value is set to **true**.

   Repeat this configuration for each MID Server that you want to use for running TSOM patterns.

TSOM Visibility Installation Disclaimer

In order to support the TSOM Visibility solution, we have modified the CMBD CI Class Models store application, introducing updates to the IRE Identification Rules for the following telecom CIs.

For more information about CMBD CI Class Models store application, see CMDB CI Class Models.

- **Interface Cards**
- **Slots**
- **Subslots**
- **Network Interfaces**

TSOM Visibility requires CMDB CI Class Models Version 1.69.0 (**sn_cmdb_ci_class**).

If you install any of the TSOM Visibility applications (**sn_sgc_altiplano_connector**, **sn_tsom_patterns**, or the **sn_tsom_core** plugin), the CMDB CI Class Models store application is automatically updated (or installed) to Version 1.69.0.

Note: An administrator can still upgrade the CMDB CI Class Models store application to Version 1.69.0 at their discretion, regardless of whether their Yokohama instance has TSOM Visibility or even if their instance is on a pre-Yokohama release (for example, Washington DC or Xanadu).

IMPORTANT! If an administrator deploys CMDB CI Class Models Version 1.69.0—whether or not TSOM Visibility is installed—any customized IRE identification rules applied to one or more of the above-mentioned telecom CIs may be affected. These rules will require careful validation to ensure proper functionality.

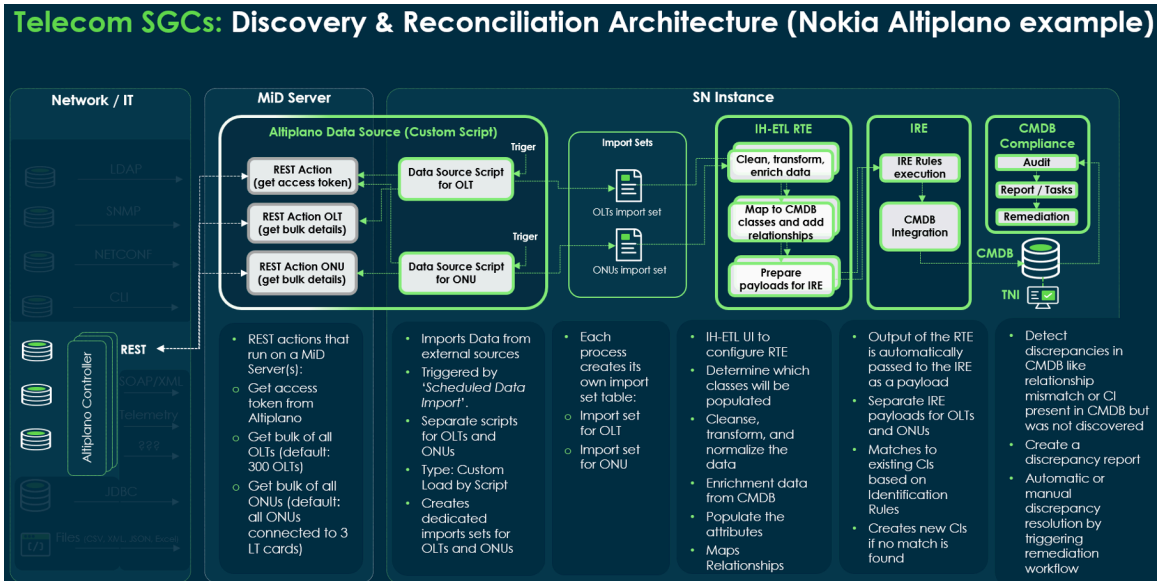
Telecom Discovery using Service Graph Connectors (SGC)

Service Graph Connectors are predefined integrations that ingest data into the Configuration Management Database (CMDB) from third-party sources (for example, northbound APIs of EMS/NMS/Controllers, which manage various xNFs) across different network domains, while enabling a structured, telecom-model-aligned view of network resources and services.

They can be used alongside any existing Service Graph Connectors, such as those for security, servers, software, monitoring, internet of Things (IoT), and cloud.

Architecture using Telecom Service Graph Connectors

This is an example of the implementation for the Nokia Altiplano Service Graph Connector. The architecture of other connectors may vary.



IntegrationHub ETL (3.2)

This store app to create and manage ETL transform maps, which integrate third-party data into the CMDB or into non-CMDB tables without compromising the integrity of data. IntegrationHub ETL provides a simplified user interface that guides you through the integration process end-to-end, including a test integration run of sample data.

For more information, see IntegrationHub ETL.

Robust Transform Engine (RTE)

This plugin is used to transform raw source data that is stored in staging tables, into the data that is mapped and integrated into the CMDB. RTE uses ETL transform maps that were created for integration during data transformation.

For more information, see Create a robust import set transformer.

MID Server

MID Server is a Java application that runs as a Windows service or UNIX daemon on a server within your local network. The ServiceNow® MID Server facilitates communication and data transfer between a ServiceNow instance and external applications, data sources, and services.

For more information, see MID Server.

Identification & Reconciliation Engine (IRE)

IRE offers a centralized framework for identifying and reconciling data from multiple sources. It confirms the integrity of the CMDB and some non-CMDB tables when various data sources are used to create or update CI records.

For more information, see the CMDB Identification and Reconciliation (IRE).

CMDB Compliance Certification Audits for Telecom Discrepancy Identification & Reconciliation

CMDB Compliance is a toolset that enables administrators to certify CMDB data for accuracy and resolve any discrepancies found. In Telecom Discrepancy Identification & Reconciliation,

we use the Certification audits feature to discover and analyze discrepancies in the CMDB, generate Certification follow-on tasks, and enable remediation workflows.

For more information on how it's used for Discrepancy Identification & Reconciliation, see Telecom Discrepancy Identification and Reconciliation.

TNI Entity Creation Logic

Whenever the system identifies that the customer has TNI installed, it will automatically create a TNI entity record for all network data discovered.

If TNI is installed, a payload like the following one will be added to the IRE payload for each item (with inventory_category populated based on the className):

```
related = [{
            "className": "tni_entity",
            "values": {
                "inventory_category": ""
            }
        }];
```

As a result, the discovered CI is in both the cmdb_ci and tni_entity tables.

Telecom Service Graph Connectors supported

Service Graph Connector for Nokia Altiplano.

See other available Service Graph Connectors.

Supported versions

Supported ServiceNow® versions: Yokohama and beyond.

Service Graph Connector for Nokia Altiplano

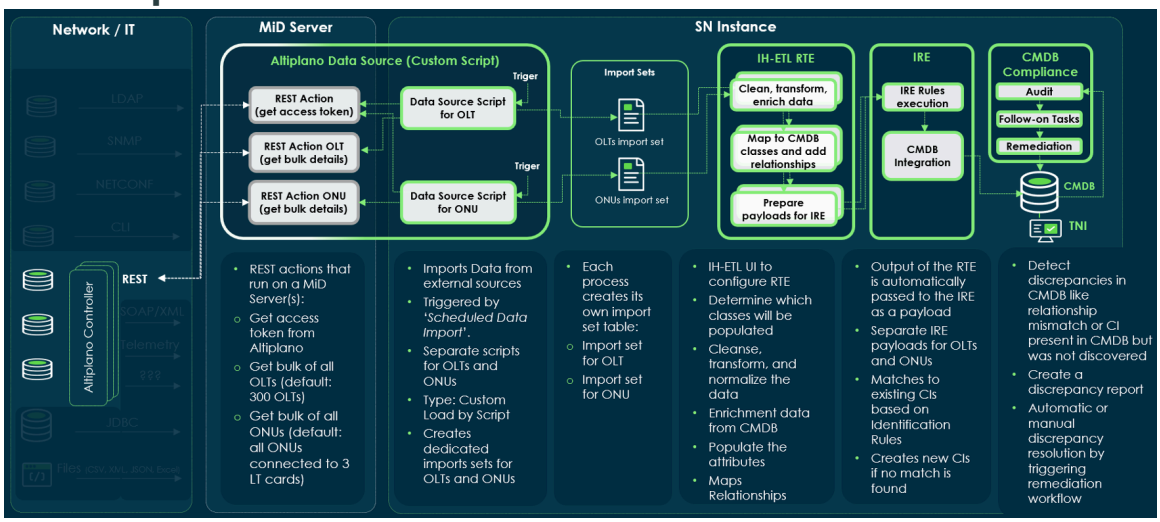
Use the Service Graph Connector for Nokia Altiplano Access Network SDN Controller to pull in data from the Nokia Altiplano software into your ServiceNow instance using REST APIs.

The Service Graphs Connector for Nokia Altiplano pulls in asset inventory data (physical network resources) from the Nokia Altiplano database.

Request apps on the Store

Visit the ServiceNow Store website 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.

Nokia Altiplano SGC Architecture



Data Source defined as the following:

- Custom load by script
- Data in single column

For a general overview of Service Graph Connector technology, see Getting started with Service Graph Connectors.

Supported versions

Supported Nokia Altiplano Controller minimum versions: 24.6.

Use cases

The following are examples on how you can use the Nokia Altiplano Service Graph Connector:

- **The Nokia Altiplano Service Graph Connector** can be used to pull data from the Nokia Altiplano Access Controller via REST API (through a MID Server), confirming that the CMDB is populated with accurate, up-to-date information about physical network resources such as OLTs and ONTs, among others. This integration provides a telecom-model-aligned view of network resources and their relationships.
- **Future Capabilities**: In upcoming releases, the Nokia Altiplano Service Graph Connector expands its capabilities to discover not only physical resources but also logical resources and services/connections, enabling a more comprehensive view of both the physical and logical aspects of the network. It will also support event-driven discovery, where the Altiplano Controller notifies the ServiceNow instance (via the MID Server) about a change and sequentially triggers a discovery task.
- Ability to configure and save synchronization schedules.

Guided setup

The guided setup for the Service Graphs Connector for Nokia Altiplano provides an organized sequence of tasks to configure the integration on your instance. To access the guided setup, see Configure Service Graph Connector for Nokia Altiplano.

CMDB Integrations Dashboard

The Integration Commons for CMDB store app provides a dashboard with a central view of the status, processing results, and processing errors of all installed Service Graph Connectors.

You can see metrics for all integration runs. You can filter the view to a specific integration, a specific time duration, or a specific integration run. For more details about monitoring SolarWinds integrations in the CMDB Integrations Dashboard, see Integration Commons for CMDB.

Import Sets

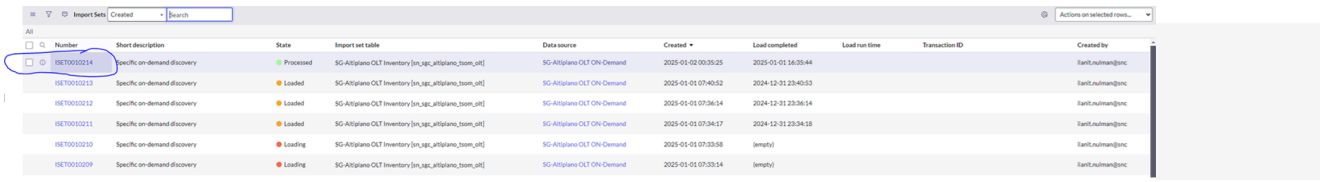
The Import Sets are the input for the IntegrationHub ETL, where the transformation maps create and model relationships. When data is loaded into the Import Set, the transformation process is triggered.

Before you begin

Role required: admin

Procedure

Navigate to **All** > **System Import sets** > **Administration** > **Import Sets**.
The Data Source script automatically creates staging tables.



Data Mapping

Data from data sources in the Nokia Altiplano software is mapped and transformed into ServiceNow CMDB tables using the Robust Transform Engine (RTE). Data is inserted into ServiceNow CMDB using the Identification and Reconciliation Engine (IRE).

When you complete the guided setup, you can configure the integration to periodically pull data from the SolarWinds software.

The data is loaded into staging tables, then inserted into the following CMDB target tables with the following relationships:

Mapping CMDB CIs and CI Relationships (Physical Layer)

CIs	CI Relationships
OLT CI	OLT Device is represented by the OLT CI. Table name: **cmdb_ci_optical_line_terminal** OLT CI contains Slot CIs.
ONT CI	ONT Device is represented by the ONT CI. Table name: **cmdb_ci_optical_network_terminal** ONT CI contains Network Interface CIs.

CIs	CI Relationships
	ONT Network Interface CIs is contained by the ONT CI.
Slot CI	Chassis slots are represented by the Slot CI. Table name: **cmdb_ci_container_slot** Slot CI is contained by OLT CI. Slot CI contains the Interface Card CI (LT/NT cards, Fan/PSU units).
Subslot CI	LT/NT card cages are represented by the Subslot CI. Table name: **cmdb_ci_container_subslot** Subslot CI is contained by OLT LT card CI. Subslot CI is contained by OLT NT card CI. Subslot CI contains the Interface Card CI (transceiver cards).
Interface Card CI	LT and NT cards are represented by the Interface Card CI. Transceiver cards are represented by the interface Card CI. Fan and Power Supply Units are represented by the Interface Card CI. (Special cards, Fan, and Power Supply Units CI might be changed in the next releases). Table name: **cmdb_ci_interface_card** LT Card contains the Subslots CIs. NT Card contains the Subslots CIs. NT cards CIs contains Network Interface CIs LT/NT transceiver cards CIs contains Network Interface CIs. LT/NT Card CI is contained by the LT/NT Slots CI. LT transceiver Card CI is contained by the subslot CI (LT Card). NT transceiver Card CI is contained by the Subslot CI (NT card).

CIs	CI Relationships
Network Interface CI	LT card PON access ports as well as NT card network ports are represented by the Network Interface CI. Table name: **cmdb_ci_ni_interface** Network Interface CIs contained by LT transceiver/NT transceiver/NT cards CIs. ONT Network Interface CI is contained by ONT CIs.

You can use the **IntegrationHub ETL** application to view and manage data maps.

For more information, see IntegrationHub ETL.

Supported xNFs

- Lightspan MF-2 (OLT)
- Lightspan ISAM FX-4 (OLT)
- Lightspan XS-010X-Q (ONT)
- Lightspan XS-010X-R (ONT)

Special System properties

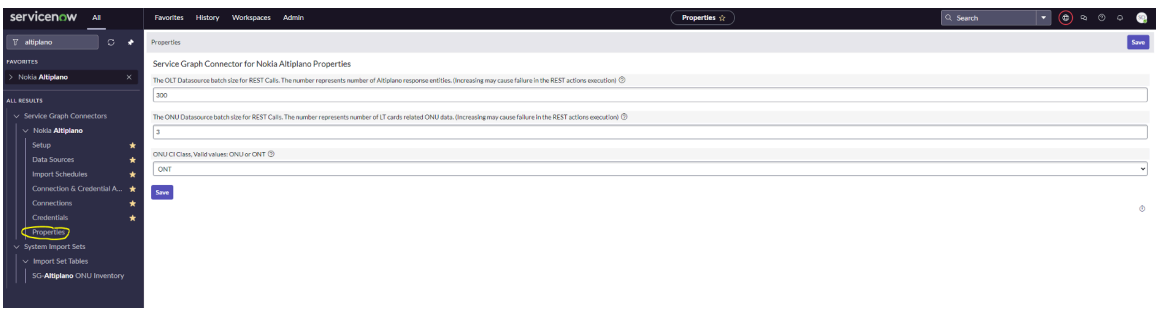
Service Graph Connector for Nokia Altiplano installs special system properties that control various behaviors of the application.

Before you begin

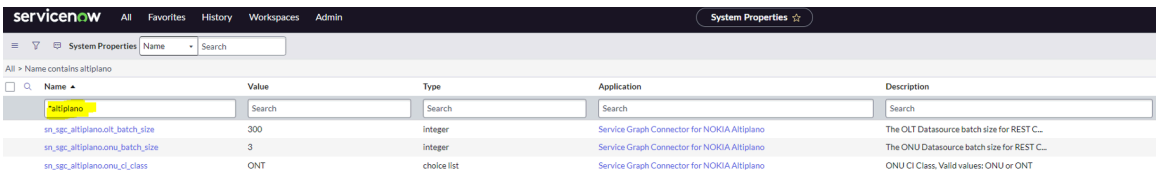
Role required: admin

Procedure

Navigate to **All** > **Service Graph Connectors** > **Nokia Altiplano** > **Properties**.



You can also access these properties by entering **sys_properties.list** and filtering by the name '*altiplano'.



Nokia Altiplano SGC System Properties

Property Name	Recommended / Default Value	Description
sn_sgc_altiplano.olt_batch_size	300	The OLT Datasource batch size for REST Calls. The number represents number of Altiplano response entities. (Increasing may cause failure in the REST actions execution)
sn_sgc_altiplano.onu_batch_size	8	The ONU Datasource batch size for REST Calls. The number represents number of LT cards related ONU data. (Increasing may cause failure in the REST actions execution)
sn_sgc_altiplano.onu_ci_class	ONT	ONU CI Class, Valid values: ONU or ONT

Examples of Retrieving Data from Nokia Altiplano via REST API

Examples of Retrieving Data from Nokia Altiplano via REST API.

URL format

Versioned URL: POST: `altiplano-indexsearch/latestcompleted-inv/_search`

Example: For OLT

```
{
    "_source": [
        "deviceAVmetadata",
        "inventorymetadata",
        "inventorydata.ietf-hardware:hardware",
        "inventorydata.ietf-hardware:hardware-state",
        "inventorydata.nokia-state:state"
    ],
    "sort": [{"_id": {"order": "asc"}}],
    "from": 0,
    "size":300
}
```

Example: For ONU

```
{
      "query": {
            "bool": {
                "should": [
                    {
                        "exists": {
                            "field":
 "inventorydata.ietf-interfaces:interfaces-state.interface.bbf-x
ponvani:v-ani.onu-present-on-this-olt.detected-serial-number"
```

```
                        }
                    }
                ]
            }
    },
    "_source": [

 "inventorydata.ietf-interfaces:interfaces-state.interface.bbf-x
ponvani:v-ani.onu-present-on-this-olt.detected-serial-number",

 "inventorydata.bbf-fiber-onu-emulated-mount:onus.onu.root.ie
tf-hardware-mounted:hardware-state",

 "inventorydata.bbf-fiber-onu-emulated-mount:onus.onu.name"
    ],
    "sort": [{"_id": {"order": "asc"}}],
    "from": 0,
    "size": 3
}
```

Configure Service Graph Connector for Nokia Altiplano

This document explains how to configure the Service Graph Connector for Nokia Altiplano using Guided setup to integrate network resource data from the Nokia Altiplano Access Controller (REST API) into the ServiceNow CMDB. It includes steps for setup, authentication, and scheduling to confirm accurate integration of network data.

To use Service Graph Connector for Nokia Altiplano, you need a subscription to TSOM.

Request apps on the Store

Visit the ServiceNow Store website 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.

Dependencies and requirements

- TSOM Visibility plugin (**sn_tsom_core**)
- Discovery Core plugin (**com.snc.discovery.core**), which is automatically installed by Discovery.
- ITOM Discovery License plugin (**com.snc.itom.discovery.license**). You must activate this plugin.
- ITOM Licensing plugin (**com.snc.itom.license**).

  For more information, see Request Discovery.
- For development environment only - IntegrationHub ETL (**sn_int_studio**).
- Nokia Altiplano Platform (access to its REST northbound API).

Roles required: admin

TSOM Visibility Installation Disclaimer

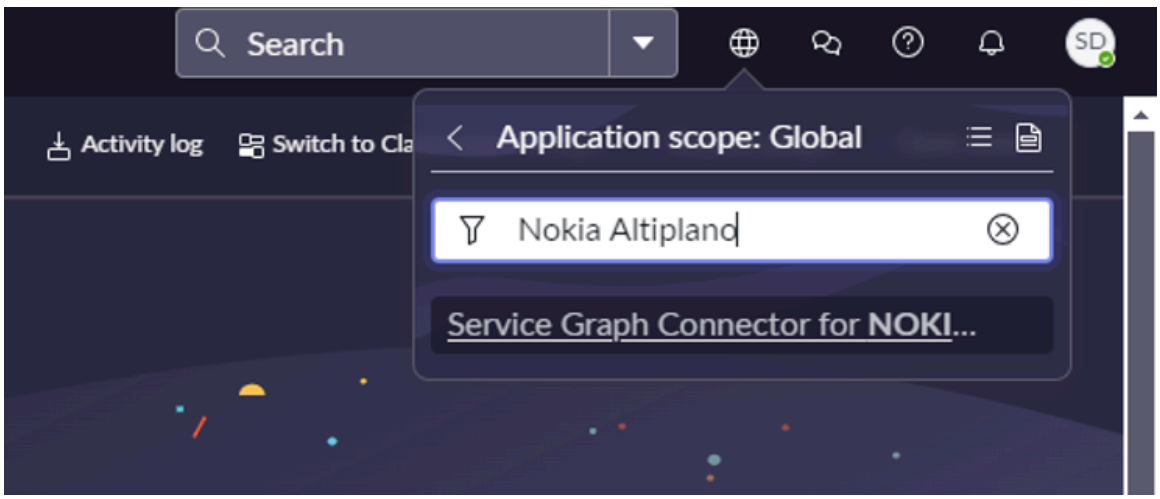
See TSOM Visibility Installation Disclaimer, for important information and requirements related to the installation process.

Initial Setup procedure

This procedure is applicable for initial setup of the Nokia Altiplano.

Before you begin
Role required: admin

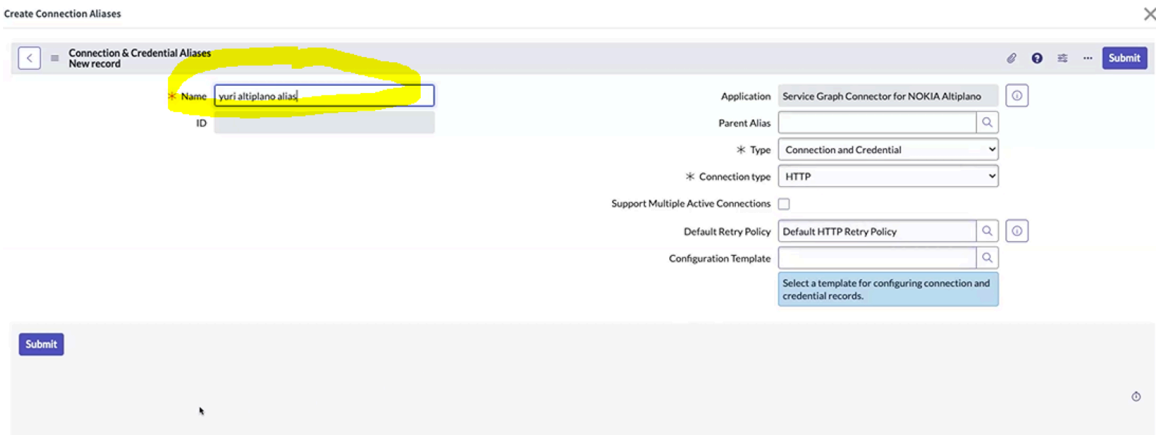
Change the application scope to 'Service Graph Connector for Nokia Altiplano' by selecting the , searching for Nokia Altiplano, and selecting it.



Procedure

1. Navigate to **All** > **Service Graph Connectors** > **Nokia Altiplano** > **Setup**.
2. On the Getting started page, select **Get Started**.
3. Configure MID Server:
   a. Select **Configure**.
   b. If a MID Server has been configured, set all to **Mark as Complete**.
      For more information on how to install and configure MID Server, see Configuring MID Server .
4. Configure Connectivity:
   a. Select **Get Started**.
   b. Configure aliases for the connections and credentials:
      i. Select **Configure**.
      ii. In the Name filed, specify the allies name.

iii. Leave the rest of the fields as default, select **Submit** and then **Mark as Complete**.



This enables using the connection by name rather than directly, enabling the collector to extract all active aliases from the CMDB and start performing data collection on the HTTP connection bound to it.

c. Create the credentials to access the Nokia Altiplano Controller:

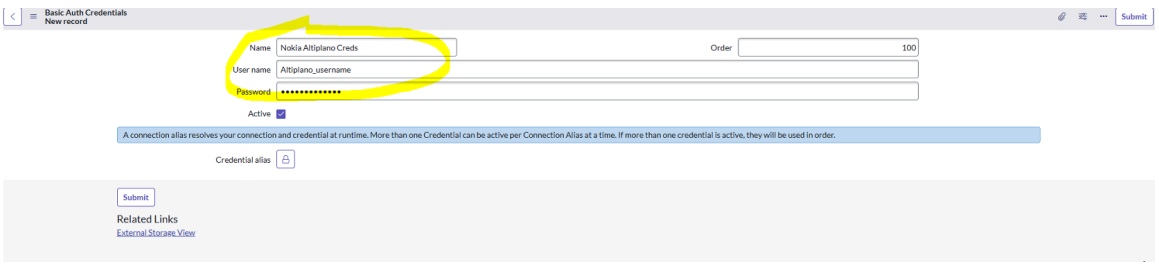
i. Select **Configure**.

ii. In the **Name** field, specify your Nokia Altiplano instance user name.

iii. In the **Password** field, specify your Nokia Altiplano instance password.

> **Note:**
>
> Other authentication fields may be required depending on the authentication methods used in your Nokia Altiplano instance. By default, we use Basic authentication credentials (as part of the Guided setup).

iv. Leave the rest of the fields as default, select **Submit** and then **Mark as Complete**.



d. Create HTTP Connection:

i. Select **Configure**.

ii. In the **Name** filed, specify the HTTP connection name.

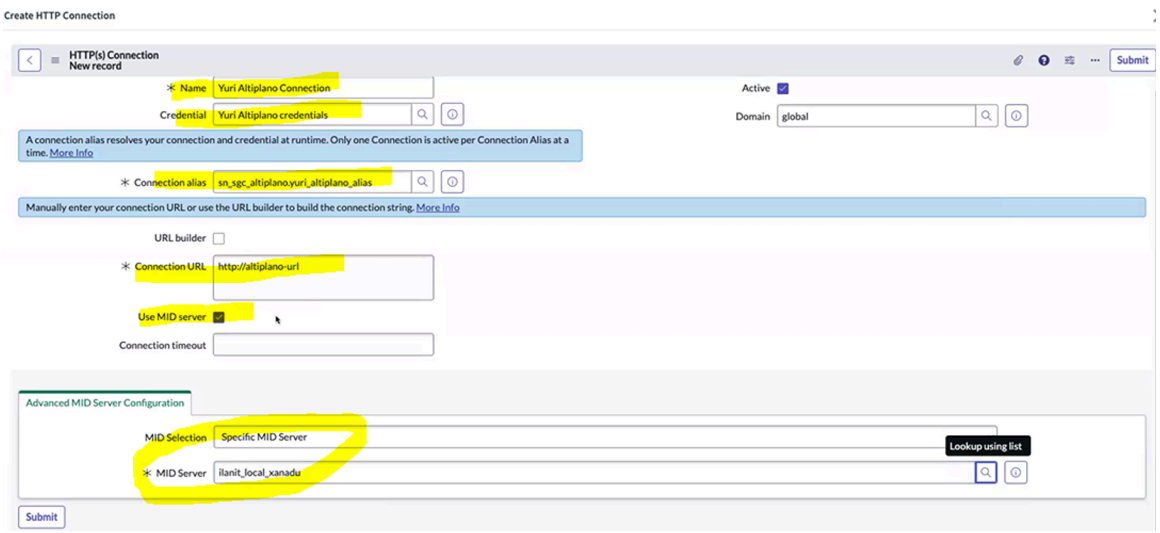
iii. In the **Credentials** field, select the magnifying glass icon and select the credential defined in section 5.b

iv. In the **Connection alias** field, select the icon and select the Connection allies defined in section 5.a

v. In the Connection URL field, specify the Nokia Altiplano URL.

vi. Check the **Use MID Server** check box and select the specific MID Server or MID Server Cluster to run discovery from.

vii. Leave the rest of the fields as default, select **Submit** and then **Mark as Complete**.



5. Configure Data Collection Schedule:

a. Select **Get Started**.

i. Select **Configure**.

ii. In the **Name** filed, specify the scheduler name.

iii. In the **Data source** field, select the icon and select a data source for OLT discovery:

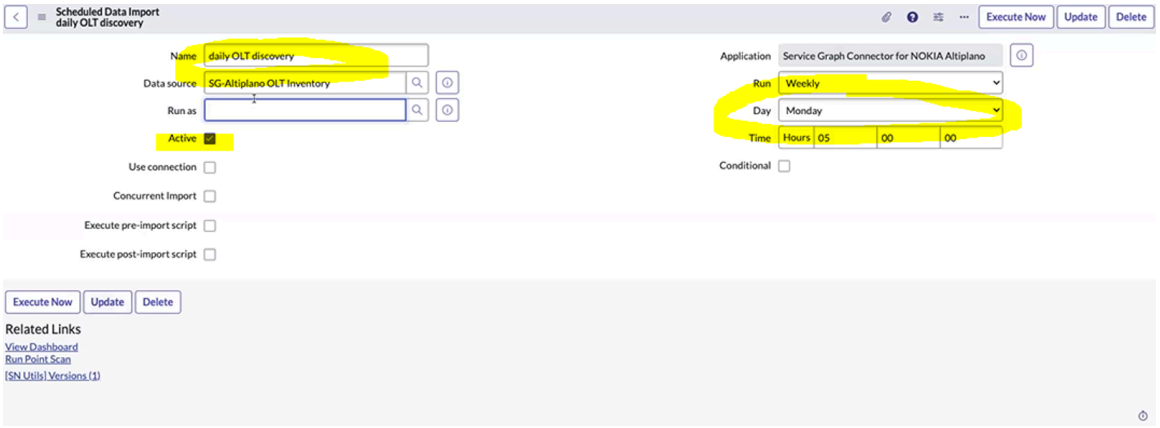
- **SG Altiplano OLT for All**- Select this option for all aliases (instances) of Nokia Altiplano.
- **SG Altiplano OLT [ALIAS_NAME]**- Select this option for the specific alias of an Altiplano instance.

iv. Check the **Active** check box to activate.

Specify when do you want this schedule to run (daily, weekly, monthly. periodically, after parent run, once).

v. You can run it now by selecting the **Execute Now**.

vi. Leave the rest of the fields as default, select **Update**, and then **Mark as Complete**.



b. Schedule Data Collection for ONU:

i. Select **Configure**.

ii. In the **Name** field, specify the scheduler name.

iii. In the **Data source** field, select the 🔍 and select a data source for ONU discovery:

- **SG Altiplano ONU for All**- Select this option for all aliases (instances) of Nokia Altiplano.
- **SG Altiplano ONU [ALIAS_NAME]**- Select this option for the specific alias of an Altiplano instance.

iv. Check the **Active** check box to activate.

> **Note:** Specify when do you want this schedule to run (daily, weekly, monthly. periodically, after parent run, once).

v. You can run it now by selecting the **Execute Now**.

vi. Leave the rest of the fields as default, press **Update** then **Mark as Complete**.

6. Test the connection by using the **Test Load 20 Records** related link.

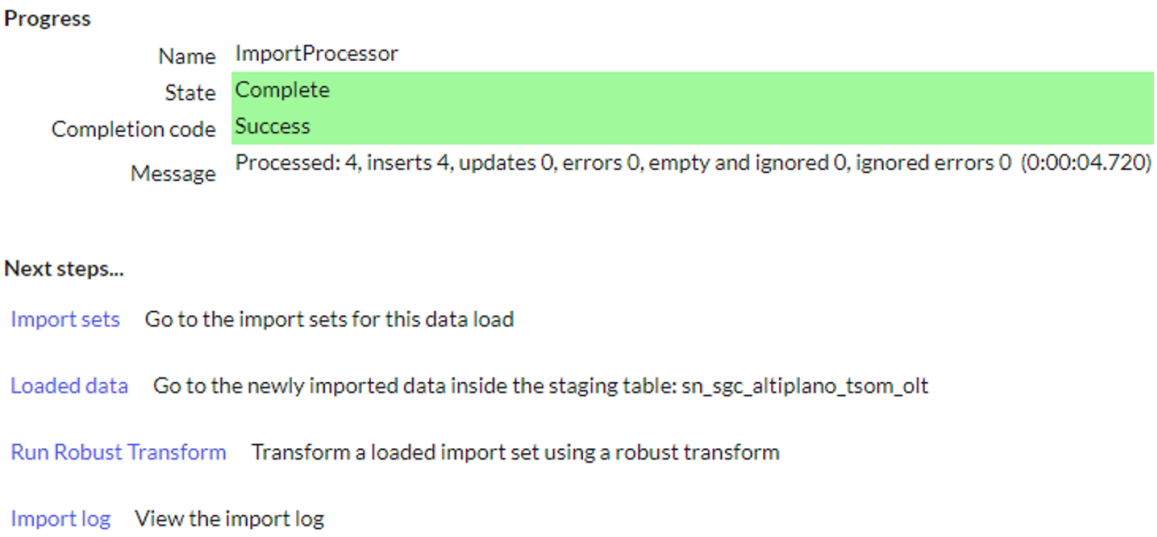
This step tests the selected data source and confirms that the data is loaded into the staging table. A successful connection indicates that the selected Nokia Altiplano data source has connected successfully. Perform this action for both **SG-Altiplano OLT for all** and **SG-Altiplano ONU for all** data sources, as well as for any other data sources you have created.

a. Navigate to **Service Graph Connectors** > **Nokia Altiplano** > **Data Sources**.

7. Select the desired Data Source.

8. On the Data Source form, under Related Links, select Test Load 20 Records.

9. Wait for the test result State to be Complete with a Completion code of Success.



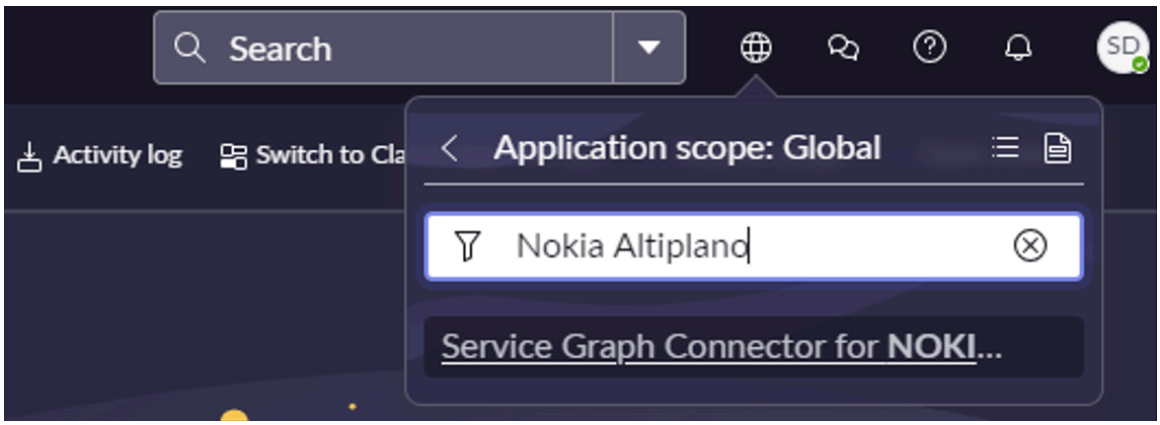
Multi-Instance setup

Add Second instance of a Nokia Altiplano and subsequent instances.

Before you begin

Role required: admin

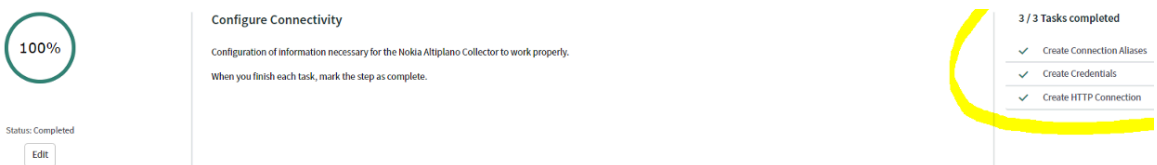
Change the application scope to Service Graph Connector for Nokia Altiplano by selecting the icon, searching for Nokia Altiplano, and selecting it.



Procedure

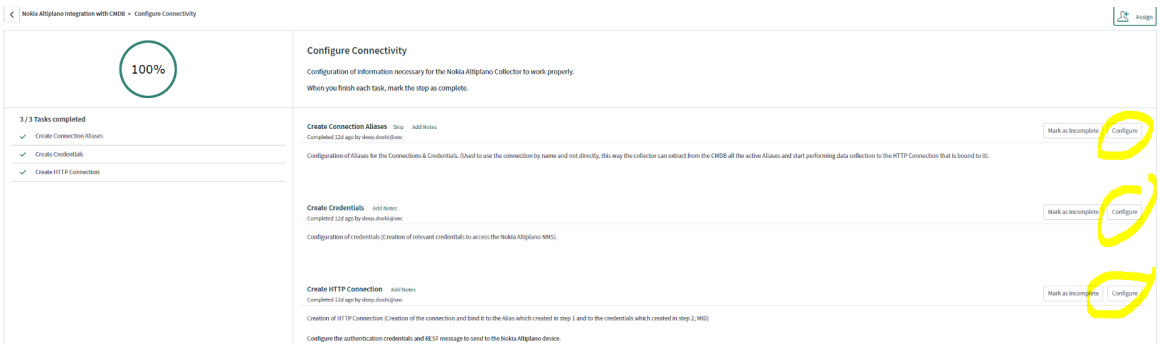
1. Navigate to **All** > **Service Graph Connectors** > **Nokia Altiplano** > **Setup**.

2. On the Getting started page, select **Get Started**.
Repeat all the steps under the Configured Connectivity section. It creates configuration entries for the new instance of Nokia Altiplano.



3. Select **Create Connection Alias**.

4. Configure Create Connection Alias, Create Credentials, and Create HTTP Connection.

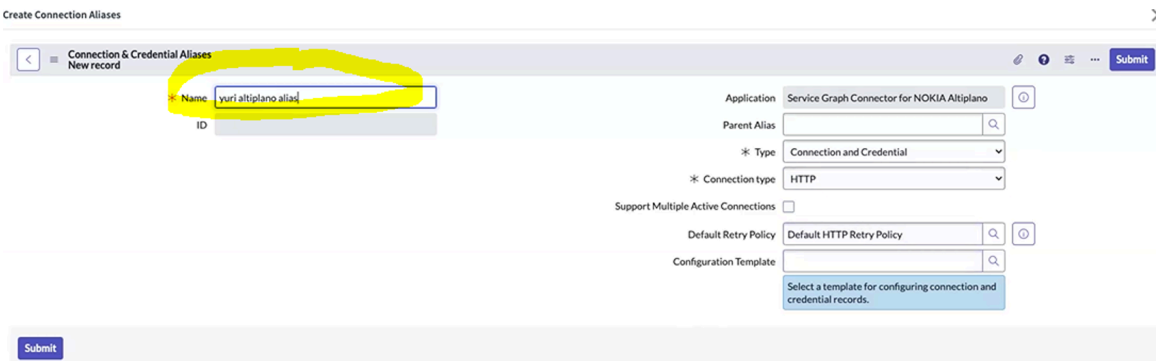


a. Configure aliases for the connections and credentials:

i. Select **Configure**.

ii. In the **Name** filed, specify the allies name.

iii. Leave the rest of the fields as default, select **Submit**, and then **Mark as Complete**.



This enables using the connection by name rather than directly, enabling the collector to extract all active aliases from the CMDB and start performing data collection on the HTTP connection bound to it.

b. Create the relevant credentials to access the Nokia Altiplano Controller:

i. Select **Configure**.

ii. In the **Name** field, specify your Nokia Altiplano instance user name.

iii. In the **Password** field, specify your Nokia Altiplano instance password.

> **Note:** Other authentication fields may be required depending on the authentication methods used in your Nokia Altiplano instance.

iv. Leave the rest of the fields as default, select **Submit** and then **Mark as Complete**.

c. Create HTTP Connection:

i. Select **Configure**.

ii. In the **Name** filed, specify the HTTP connection name.

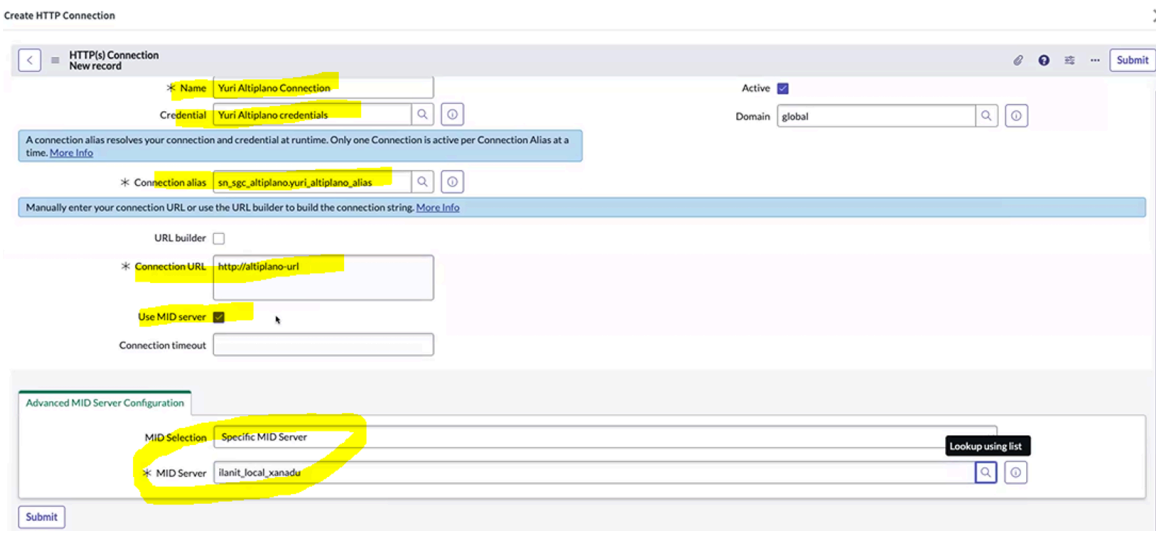
iii. In the **Credentials** field, select the icon and select the credential defined in section 5.b

iv. In the **Connection alias** field, select the icon and select the Connection allies.

v. In the **Connection URL** field, specify the Nokia Altiplano URL.

vi. Check the **Use MID Server** check box and select the specific MID Server or MID Server Cluster to run discovery from.

vii. Leave the rest of the fields as default, select **Submit** and then **Mark as Complete**.



5. Test the connection by using the **Test Load 20 Records** related link.

Telecom Discrepancy Identification and Reconciliation

Telecom Discrepancy Identification & Reconciliation confirms that ServiceNow's CMDB/TNI accurately reflects the updated state of the network and remains synchronized with planned or designed inventory resources.

It continuously compares network data with CMDB/TNI CI records through audits, identifying and resolving discrepancies to maintain alignment between the two. This process is essential for confirming data integrity, helping to prevent inconsistencies, and keeping the CMDB/TNI in synchronization with the actual network deployment.

Maintaining reliable and accurate data in your CMDB/TNI is a key enabler for seamless order and service fulfillment, assurance, security, and asset management. This capability is one of the key enablers for automation and aligns with the TM Forum Autonomous Network Operations (ANO) framework.

Telecom Discrepancy Identification & Reconciliation Licensing

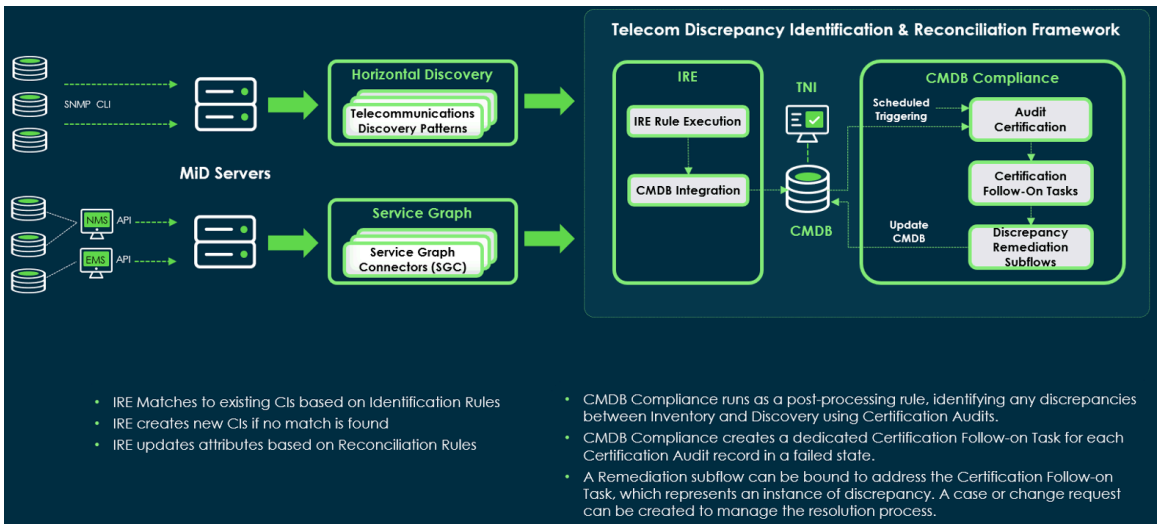
The ServiceNow AI Platform uses a licensing model in which your organization is billed for the use of TSOM Visibility applications. Telecom Discovery, Telecom Discrepancy Identification & Reconciliation and TSOM Visibility (plugin) are components of TSOM Visibility. To use TSOM Visibility, your organization must subscribe to TSOM. Please note that ServiceNow's product documentation doesn't include details on pricing, packaging, or other specifics, as these are determined by your organization's customer contract.

Exploring Telecom Discrepancy Identification and Reconciliation

Telecom Discrepancy Identification & Reconciliation solution is designed to confirm the accuracy and consistency of network resource data between network systems and inventory management databases, such as CMDB/TNI.

Telecom Discrepancy Identification & Reconciliation relies on Telecom Discovery and platform capabilities to perform its functions.

Telecom Discrepancy Identification & Reconciliation Overview



TSOM Visibility plugin

The Telecom Discrepancy Identification & Reconciliation logic is a component of the TSOM Visibility plugin (sn_tsom_core). This plugin encompasses shared logic essential for both Telecom Discovery and Telecom Discrepancy Identification & Reconciliation processes. It includes telecom-specific discrepancy detection and remediation capabilities, along with other foundational logic designed to support current and future telecom application functionalities.

Identification & Reconciliation Engine (IRE)

IRE offers a centralized framework for identifying and reconciling data from multiple sources. It confirms the integrity of the CMDB and some non-CMDB tables when various data sources are used to create or update CI records.

- IRE matches existing CIs based on Identification Rules.
- IRE creates CIs if no match is found.
- IRE updates are attributed based on the Reconciliation Rules.

For more information, see CMDB Identification and Reconciliation (IRE).

CMDB Compliance and Telecom Discrepancy Identification & Reconciliation

CMDB Compliance is a toolset that enables administrators to certify CMDB data for accuracy and identify discrepancies detected during compliance audits. It can also automatically generate and assign Follow-on Tasks for failed audit records, which serve as tasks to trigger an appropriate remediation subflow to correct discrepancies. CMDB Compliance Audits form the foundation of our Telecom Discrepancy Identification & Reconciliation.

- CMDB Compliance runs audits as a post-processing rule, identifying anomalies (discrepancies) in the CMDB.
- CMDB Compliance creates a Follow-On Task for each Audit Record in a failed state (the failed state is the result of an audit finding an anomaly or discrepancy in the CMDB). A remediation flow can be designed and triggered for each Follow-On Task to address and resolve the discrepancy.

The logic for Telecom Discrepancy Identification & Reconciliation, as well as the example remediation subflows, are included in the Yokohama release and will be installed automatically with the TSOM Visibility plugin.

For more information on the general CMDB Compliance toolset, see CMDB Compliance.

Discrepancy Identification Scenarios (using Certification Audits)

There are two key discrepancy categories that can be detected between Inventory (CMDB) and Discovery that are described below:

- Entities that exist in the Inventory but don't exist in the Network.
- Entities that exist both in the network and in the inventory but differ in their hierarchy.

Discrepancy identification in TSOM Visibility relies on using CMDB Compliance (Certification Audits) and has extended it by adding specific logic that uses model relationships and information to identify mismatches.

Note: The current release focuses on the identification and reconciliation of discrepancies at the physical resource layer. In the upcoming release, this functionality is extended to include support for the logical resource layer and attributes as well.

For more information on the general Certifications Audits feature, see Certification audits.

Follow-On Task types created for failed Audit Result Records

The following discrepancy types (Audit Results) can be found for Parent CI and child CIs for each relationship record in the CI Relationship table (cmdb_rel_ci) that matches the conditions, and the following Follow-On Tasks can be created for each of the failed Audit Results:

1. The most recent discovery date not set- generated in case the Most recent discovery date field in CI is missing.
2. The most recent discovery date not within configured threshold- generated in case the difference in the Most recent discovery date field value between a Parent CI and child CI is more than 2.5 days.

   By default, it is set to 2.5 days in the sn_tsom_core.discovered_date.diff.threshold.in.days system property and can be changed.

3. CI model not found–(the 'Model ID' field isn't set or data is invalid). Generated in case a corresponding CI model isn't found. If a CI model isn't found, the next validations (4-6) are

irrelevant because they rely on CI models. In case a CI model is found, the audit will continue to the next validations (4-6).

4. Slots occupied discrepancy-Generated in case a Card occupies an incorrect number of Slots.
5. Model relationships not defined-relevant only if TNI is installed. Generated if the audit is unable to find a relationship between Parent and child CI models in the Network Model Relationships table.
6. Incorrect number of relationships - relevant only if TNI is installed. Generated if the audit finds that the number of discovered child CI records exceeds the maximum number of its corresponding Parent CI record in the model relationship **Count** field in the Network Model Relationship table.

For more information on the general Follow-on Tasks feature, see Building Subflows.

Discrepancy Remediation Subflows

Once an audit identifies a discrepancy, it's logged as a Follow-on Task. The system enables users to define a subflow for specific discrepancy scenarios, enabling them to distinguish between various types of discrepancies and create custom flows to remediate them.

For more information on how to build a subflow, see Building subflows.

Usage Example

The following is an example of a specific scenario on how you can use Telecom Discrepancy Identification & Reconciliation:

Assume that a piece of equipment was initially discovered with a card (Card40) in its slot (Slot40). Over time, an issue was identified with Card40, and it was replaced by Card41. The inventory (CMDB), however, still contains a Card40 CI, while on the network, it has been replaced by Card41. When the next discovery job is executed, the Card41 CI will be discovered and added to the CMDB in the same slot (Slot40). As a result, we have two CIs (the old one—Card40—and the newly discovered one—Card41) placed in the same Slot40.

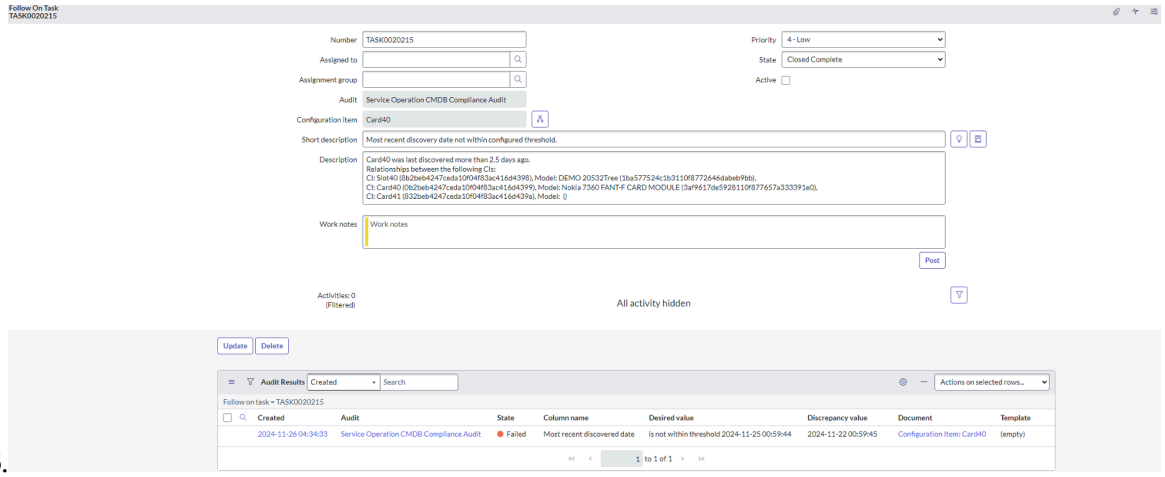
Audit identifies this discrepancy, create a Follow-on task, and enable a user to Remediate. (resolve this discrepancy and decommission Card40).

When the Service Operation CMDB Compliance Audit runs, it identifies this discrepancy and create an Audit Record in 'failed' state (in our example AUDR0001283).

1. Navigate **All** > **Compliance** > **Audits** > **.**
2. Select **Service Operation CMDB Compliance Audit**.
3. Select the **Run Audits** to run the audit.

   A Follow-on Task is automatically created for each failed audit record (in our example, TASK0020215).

4. Select **TASK0020215**.



The Follow-On Task contains a detailed description of the discrepancy. As you can see in the description, the Card40 CI is in discrepancy.

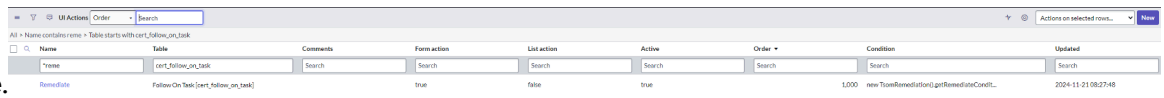
Note: This is an example of the TASK0020215 description created for the "Incorrect number of relationships" scenario. Other scenarios and environments might have different descriptions.

Card40 was last discovered more than 2.5 days ago.

Relationships between the following CIs:

CI	Model
Slot40 (8b2beb4247ceda10f04f83ac416d4398)	DEMO 20532Tree (1ba577524c1b3110f8772646dabeb9bb)
Card40 (0b2beb4247ceda10f04f83ac416d4399)	Nokia 7360 FANT-F CARD MODULE (3af9617de5928110f877657a333391e0)
Card41 (832beb4247ceda10f04f83ac416d439a)	

5. Select the **Remediate** button to remediate.
6. Remark: Remediate is a UI action which can be accessed in the following way:
7. **All** > **System Definition** > **UI Actions**.
8. Open the **Remediate** UI action to observe.

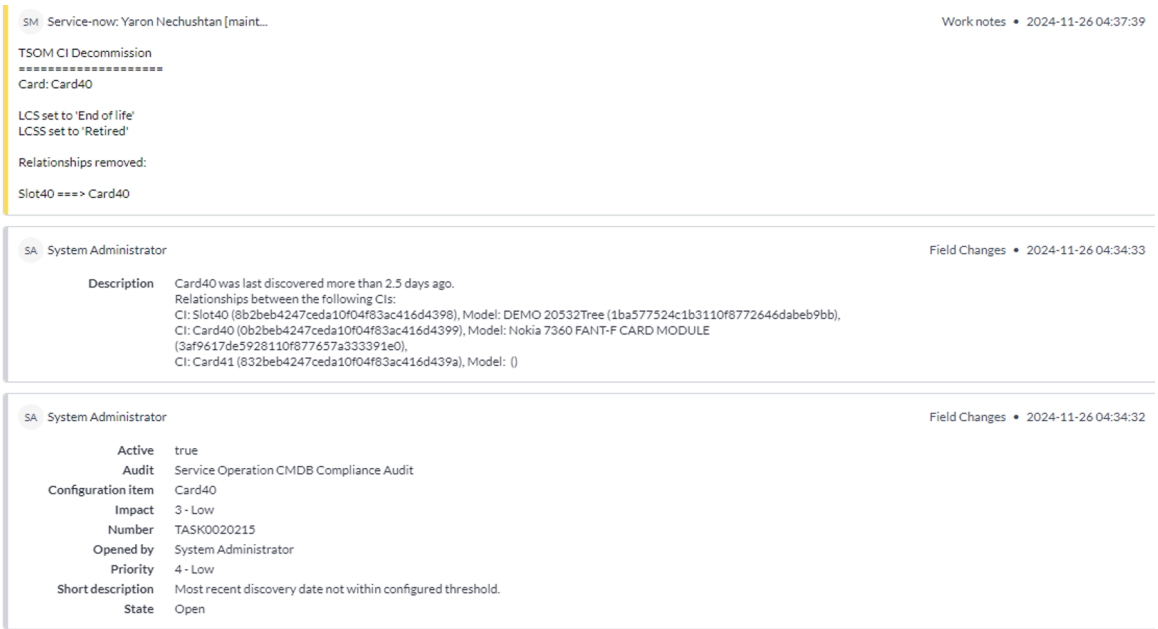


For more information on UI actions, see Defining UI actions.

For this example, the Remediate UI action (triggered by the **Remediate**) calls the Execute TSOM CI Decommission subflow to address and resolve the discrepancy specified in the Follow-On Task TASK0020215. Additionally, we must decommission an old Card40, which will be executed automatically by calling the subflow 'TSOM Decommission Card'.

Once the remediation is successfully completed, work notes are generated with the remediation results in the Follow-On Task window (TASK0020215).

As you can see in the work notes, we successfully retired Card40 and removed the relationship of Slot40 → Slot40. The discrepancy has been successfully resolved, and the CMDB CI records are now synchronized with the network state.



This example subflow is shipped with the solution. Users can define custom remediation subflows using Flow Designer.

System Properties Affecting Telecom Discrepancy Identification & Reconciliation

These system properties are part of the TSOM Visibility plugin (sn_tsom_core) and control the Telecom Discrepancy Identification & Reconciliation log (TSOM CMDB Audit). The TSOM Visibility plugin serves as an enabler for the TSOM Visibility applications, containing logic that is shared across the Telecom Discovery and Telecom Discrepancy Identification & Reconciliation solution.

TSOM Visibility System Properties (Impacts CMDB Audit)

Property Name	Recommended / Default Val
sn_tsom_core.audit.interface_card_tables	cmdb_ci_interface_card
sn_tsom_core.audit.discovery_sources	SG-Altiplano, ServiceNow

TSOM Visibility System Properties (Impacts CMDB Audit) (continued)

Property Name	Recommended / Default Valu
sn_tsom_core.audit.relationship_types	Contains:Contained by
sn_tsom_core.audit.slot_tables	cmdb_ci_container_slot
sn_tsom_core.audit.log.level	info
sn_tsom_core.audit.subslot_tables	cmdb_ci_container_subslo

TSOM Visibility System Properties (Impacts CMDB Audit) (continued)

Property Name	Recommended / Default Valu
sn_tsom_core.audit.interface_tables	cmdb_ci_ni_interface
sn_tsom_core.audit.equipment_tables	• cmdb_ci_ni_telco_equip t • cmdb_ci_ip_switch • cmdb_ci_ip_router
sn_tsom_core.audit.discovered_date.diff.threshold.in.days	2.5
sn_tsom_core.audit.max_number_of_records_to_process	100000

TSOM Visibility System Properties (Impacts CMDB Audit) (continued)

Property Name	Recommended / Default Val

Configure Reconciliation

See Configure Telecom Discrepancy Identification and Reconciliation.

Configure Telecom Discrepancy Identification and Reconciliation

This guide outlines the steps to configure Telecom Discrepancy Identification & Reconciliation (part of TSOM Visibility) to confirm accurate discovery and resolution of discrepancies in telecom network resources within your ServiceNow CMDB/TNI instance.

Before you begin

To use Telecom Discrepancy Identification & Reconciliation, you need a subscription to TSOM.

Role required: admin

Request apps on the Store

Visit the ServiceNow Store website 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.

Dependencies and Requirements

TSOM Visibility plugin dependencies:

- Telecom Service Operation Core (sn_tsom_core) CMDB CI Class Models(App ID: sn_cmdb_ci_class, Type: Store)
- Expanded Model and Asset Classes Application (App ID: sn_ent, Type: Store)
- Visibility Content (App ID: sn_pattern_design, Type: Store)
- Integration Commons for CMDB (App ID: sn_cmdb_int_util, Type: Store)
- ServiceNow IntegrationHub Starter Pack Installer (Plugin ID: com.glide.hub.integration, Type: Family)

Discovery Core plugin (com.snc.discovery.core), which is automatically installed by Discovery.

ITOM Discovery License plugin (**com.snc.itom.discovery.license**). You must activate this plugin.

ITOM Licensing plugin (**com.snc.itom.license**).

For more information, see Request Discovery.

Installation

The TSOM Visibility plugin (sn_tsom_core) is automatically installed with Telecommunications Discovery Patterns (sn_tsom_patterns) or with the Nokia Altiplano Service Graph Connector

(sn_sgc_altiplano_connector). All logic and system properties are installed on your ServiceNow instance.

To install TSOM Visibility plugin, see Configure Telecommunications Discovery (TSOM) Patterns or Configure Service Graph Connector for Nokia Altiplano.

TSOM Visibility Installation Disclaimer

See TSOM Visibility Installation Disclaimer for important information and requirements related to the installation process.

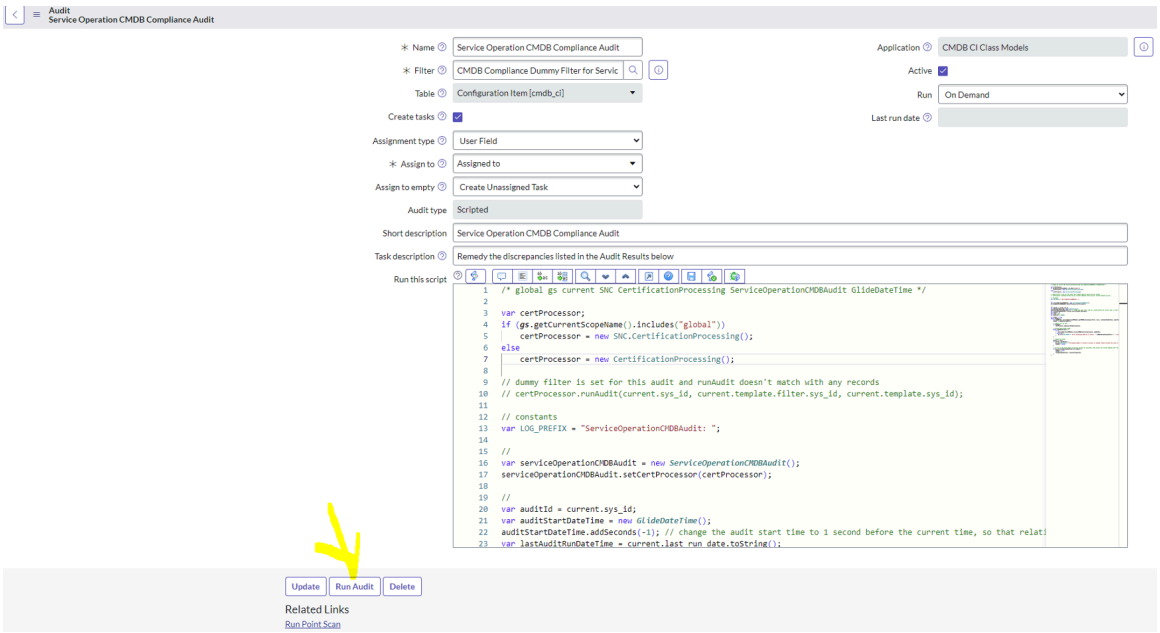
Execution

The Telecom Discrepancy Identification & Reconciliation solution relies on CMDB Health/ Compliance, which runs Certification Audits on selected tables and records in the CMDB. It operates independently of Discovery and can be run on-demand or scheduled.

For more information about CMDB Compliance, see CMDB Compliance and Certification audits.

Procedure

1. Navigate to **All** > **Compliance** > **Audits** > **Service Operation CMDB Compliance Audit ("cert_audit" table)**.
2. Select **Run Audit**.



It executes various scripts and operations.

Certification Audit Logics

Audit Results are created for each audit executed on records that matched the selection (see matching conditions in Initial Certification Audit Run).

The result's state can be certified or failed. A Follow-On Task is created for each 'failed' Audit Result record.

Initial Certification Audit Run

Validating specific CMDB tables for anomalies.

The Service Operation CMDB Compliance Audit starts to run on the CI Relationship table (cmdb_rel_ci), but only on specific records with matching conditions as follows:

- Parent AND child CI classes are supported classes, including extended tables as follows:

  slot (cmdb_ci_container_slot), subslot (cmdb_ci_container_subslot), card (cmdb_ci_interface_card), interface (cmdb_ci_ni_interface), telco equipment (cmdb_ci_ni_telco_equipment), IP switch (cmdb_ci_ip_switch), and IP router (cmdb_ci_ip_router).

  Note: These properties can be configured via sn_tsom_core.audit.* system properties.

- Parent OR child is created or updated by Discovery (discovery_source = SG-Altiplano, ServiceNow).

  Note: This property can be configured in the sn_tsom_core.audit.discovery_sources system property.

- Parent AND child life-cycle Stage is Operational.
- The CI Relationship Type is Contains::Contained By.

  Note: This property can be configured in the sn_tsom_core.audit.relationship_types system property.

Subsequent Certification Audit Runs

Follows the same logic as the Initial Certification Audit Run, but with the following additional matching selection criteria:

The timestamp in the Updated field in the CI Relationship table, or the timestamp in the Updated field of a Parent CI, or the timestamp in the Updated field of child CIs is later than the timestamp in the ‘Last run date’ field in the Service Operation CMDB Compliance Audit (this means that there was a change since the last audit).