



チューリッヒ 通信事業サービスオペレーション管理

自動翻訳

最終更新日: 2025年11月10日

これらの資料は、翻訳ソフトウェアを使用してお客様の便宜のために翻訳されています。正確な翻訳をご提供できるよう相当な努力を払っておりますが、いかなる自動翻訳も人間の翻訳者に代わることはなく、そのようなことは意図されておりません。翻訳は「現状のまま」提供されています。他言語への翻訳の的確性、信頼性または正確性については、明示または黙示を問わず、いかなる保証も行われません。翻訳ソフトには限界があるため、一部のコンテンツが正確に翻訳されていない場合があります。これらの資料の公用言語は英語です。翻訳の際に生じる相違または不一致は、コンプライアンスまたは履行の目的に関しては拘束力を有さず、法的効力はないものとします。

ここで示したいくつかの例と図は、説明の目的でのみ提供されています。ServiceNow 製品またはサービスへの実際に関連付けやつながりは意図されておらず、推測されるものではありません。

ServiceNow, ServiceNow のロゴ、Now、その他の ServiceNow マークは米国および/またはその他の国における ServiceNow, Inc. の商標または登録商標です。その他の会社名および製品名は、それぞれの所有者の商標です。

下記の ServiceNow ウェブサイト利用規約をお読みください。 www.servicenow.com/terms-of-use.html

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

目次

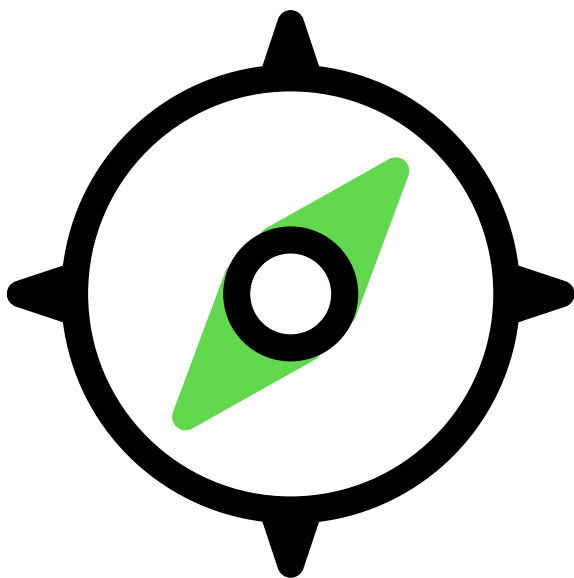
通信事業サービスオペレーション管理.....	4
通信事業サービスオペレーション管理 の探索.....	6
Telecom API イベント通知.....	9
電気通信のヴィジビリティ.....	10
通信事業サービスオペレーション管理の構成.....	40
電気通信 API 通知の構成.....	41
Telecom Visibility を設定.....	44
通信事業サービスオペレーション管理 の使用.....	80
電気通信ディスカバリーパターンの使用.....	80
Nokia Altiplano インポートを実行.....	85
TSOM スキーマバリデータを使用した JSON ペイロードの検証.....	87
電気通信不一致監査の実行.....	89
属性値不一致のレポートを生成.....	92
通信事業サービスオペレーション管理の参照.....	93
通信事業 API 通知とともにインストールされるシステムコンポーネント.....	93
Nokia Altiplano とともにインストールされるシステムコンポーネント.....	94
Telecom Discrepancy Identification & Reconciliation コンポーネント.....	95

通信事業サービスオペレーション管理

通信事業サービスオペレーション管理 (TSOM) を使用して、通信サービスプロバイダー (CSP) がネットワークとサービスの問題をプロアクティブに監視、分析、解決し、顧客に影響を及ぼす前に解決する方法について説明します。ServiceNow AI Platform上に構築された TSOM は、分散型のマルチドメイン通信環境全体で統一された運用ビューを提供し、チームによるサービスの可用性、運用効率、顧客満足度の向上を支援します。

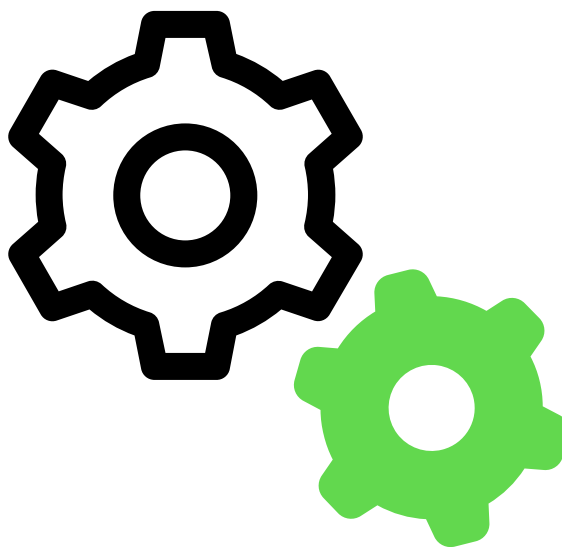
開始するには

探索



通信サービスプロバイダーが通信事業サービスオペレーション管理を使用する方法について説明します。

構成



通信事業サービスオペレーション管理を計画および構成します。

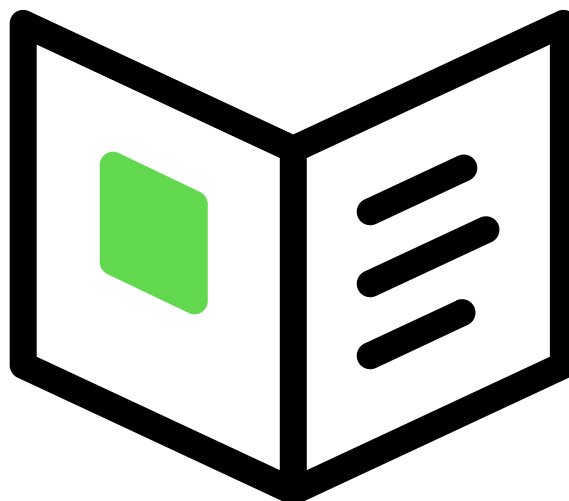
自動翻訳

使用方法



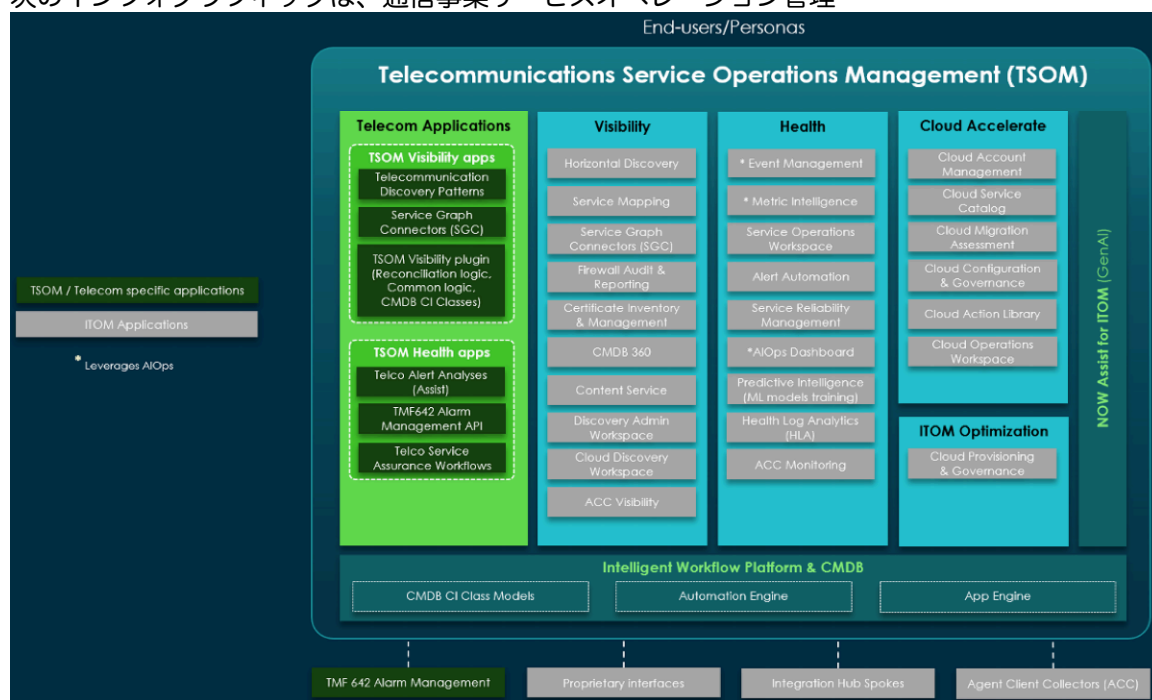
通信事業サービスオペレーション管理を使用して、包括的な通信サービスオペレーションを追跡します。

参照



通信事業サービスオペレーション管理参照情報を取得します。

次のインフォグラフィックは、通信事業サービスオペレーション管理



キテクチャを理解するのに役立ちます。

その他のリソース

役立つ情報を提供できる ServiceNow リソースは次のとおりです。

リリースノート

このリリースの新機能と変更内容の詳細については、以下を参照してください。 [通信事業サービスオペレーション管理 \(TSOM\) リリースノート](#)

ServiceNow コミュニティ

で他の 通信事業サービスオペレーション管理 ユーザーとつながる [ServiceNow コミュニティ](#)

Customer Success Center

自分のロールに関連する有用なリソースを検索し、ベストプラクティスを調べる [Customer Success Center](#)

ServiceNow University

[ServiceNow University](#) で、リアルタイムコース、自分のペースで進められるトレーニング、キャリアリソースを確認することができます。

支える

カスタマーサービスサポートへのお問い合わせ <https://support.servicenow.com/now>

通信事業サービスオペレーション管理の探索

通信事業サービスオペレーション管理 (TSOM) を使用して、通信サービスプロバイダー (CSP) がネットワークとサービスの問題を顧客に影響が及ぶ前にプロアクティブに監視、分析、解決する方法について説明します。ServiceNow AI Platform上に構築された TSOM は、分散型のマルチドメイン通信環境全体で統一された運用ビューを提供し、チームによるサービスの可用性、運用効率、顧客満足度の向上を支援します。

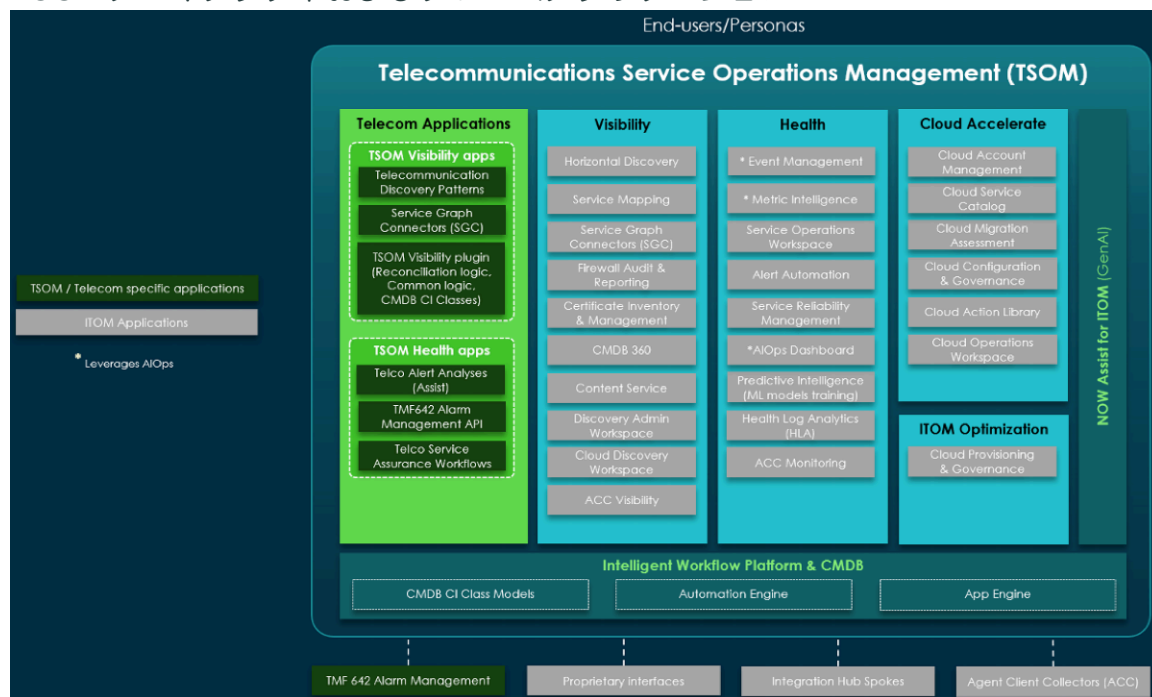
通信事業サービスオペレーション管理 は、ネットワークとサービスの健全性を完全に可視化する通信プロバイダー向けのソリューションです。TSOM は、標準 API と ServiceNow AI Platform の機能を使用して、アクセス、トランスポート、コアなどのネットワークドメイン全体からイベントを収集、関連付け、優先順位付けします。TSOM は、リアルタイムで実用的なインサイトを提供することで、フロントラインチームとバックオフィスチームの両方がサービスに影響を与える問題に対処し、一貫したパフォーマンスを維持できるようにします。

TSOM の仕組み

通信事業サービスオペレーション管理 は、既存の監視およびテレメトリプラットフォームに接続し、実用的なパターンを特定し、解決ワークフローを自動化することで、電気通信運用を簡素化します。以下を活用します。

- 標準化されたアラームの取り込みのための電気通信 API 通知を介した外部イベント管理。
- イベント管理 イベント相関、抑制、および優先順位付け。
- しきい値違反、パフォーマンスの低下、および例外動作をリアルタイムで検出するメトリックインテリジェンス。
- ユーザーが影響を受ける前にログデータを分析し、関連する問題をトリアージして根本原因を特定するヘルスログアナリティクス。
- 影響を受けるインフラストラクチャとビジネスサービスに基づいてサービス中断を評価および追跡するためのサービス影響分析。
- サービスグラフコネクタとディスカバリーにより、動的で電気通信対応 CMDB をビルドします。

TSOMアーキテクチャおよびテレコムアプリケーション



主な機能

TSOM の主な機能

機能	説明
リアルタイムのイベントモニタリング	電気通信 API 通知を介した外部イベント管理を使用して、マルチドメインネットワークモニタリングシステムからアラームとイベントを取り込みます。
イベント相関と分析	イベント管理とメトリックインテリジェンスを活用して、関連するイベントを関連付け、ノイズを低減し、異常を検出します。詳細については、「 イベント通知管理オープン API 」を参照してください。
電気通信の可視性	影響のトレーサビリティを含む、ネットワークとサービスの健全性をエンドツーエンドで可視化します。
サービスの影響分析	ネットワークまたはインフラストラクチャの問題がサービスにどのように影響するかを理解し、ビジネスインパクトに基づいて修復に優先順位を付けます。
メトリクスインテリジェンス	パフォーマンスの傾向を監視し、しきい値違反とメトリクスの例外を検出して、問題をプロアクティブに特定します。詳細については、「 メトリックインテリジェンス 」を参照してください。
ヘルスログアナリ	ServiceNow ヘルスログアナリティクスアプリケーションは、ユーザーが影響を受ける前に IT の問題を防止するために役立ちます。このアプリケーションによって、関連ログをトリージングして生データを分析できるようになり、その結果、問題の根本原因を特定でき

TSOM の主な機能 (続く)

機能	説明
ティクス	るようになります。詳細については、「ヘルスログアナリティクス」を参照してください。
自動修復	ガイド付きワークフローとプレイブックを使用して、迅速で一貫性のある監査可能な問題解決を推進します。
電気通信対応 CMDB	電気通信固有のモデルを使用してインフラストラクチャ、サービス、物理/論理構成アイテム (CI) をリンクし、正確な根本原因分析を行います。
アラート管理	相関、グループ化、および自動応答アクションにより、アラートを効率的に管理します。詳細については、「アラートの管理と監視」を参照してください。

主な特長

- MTTD および MTTR の削減: エスカレーションの前に、サービスに影響を与える問題を特定して対応します。
- 運用効率の向上: ガイド付きワークフローでノイズを低減し、運用を簡素化します。
- エンドツーエンドの可視化の獲得: インフラストラクチャの問題が顧客向けサービスにどのように影響するかを理解します。
- シームレスな統合: オープンスタンダードを備えたレガシーおよび最新の NMS/EMS ツールからデータを取り込みます。
- コンプライアンスの確保: 通信サービス保証に関する TM Forum 標準に準拠します。

主要なペルソナ

- ネットワークオペレーションセンター (NOC) エージェント: ネットワークイベントを監視し、ガイド付きの解決策を実行します。
- サービス保証マネージャー: サービスの傾向を分析し、解決メトリクスを追跡します。
- システムインテグレーター/アドミン: データの取り込み、相関ルール、およびワークフローを構成します。
- バックオフィスアナリスト: 根本原因を調査し、CMDB 内の CI 関係を追跡します。

との統合 ServiceNow AI Platform

TSOM は Now Platform に組み込まれており、次のようなコア機能とシームレスに統合されます。

- ディスカバリー
- サービスマッピング
- CMDB
- フローデザイナーとプレイブック

この統合により、電気通信ドメイン全体で一貫したワークフロー、正確なサービスモデル、および統合運用管理を確認できます。

関連トピック

[電気通信 API 通知を介した外部イベント管理](#)

[電気通信のヴィジビリティ](#)

電気通信 API 通知を介した外部イベント管理

Telecommunications API 通知を使用して、顧客のネットワークシステムで発生した外部イベントを受信し、ServiceNowで迅速に対応できるようにします。

電気通信 API 通知の概要

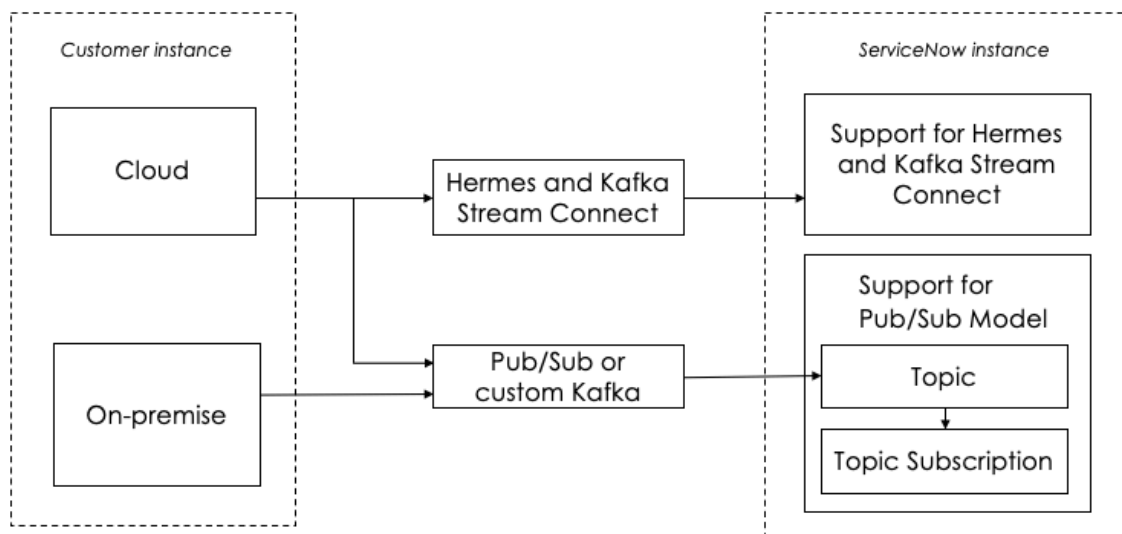
Telecommunications API 通知は、Telecommunications Alarm Management Open API アプリケーションで利用可能な機能です。Telecommunications API 通知を使用すると、ServiceNow 外部ネットワークシステムで発生した受信通知を受信し、適切なタイミングで応答できます。これにより、ポイントツーポイント接続の必要性がなくなるため、プラットフォーム機能を介して外部システムへのイベントのブロードキャストが可能になります。

電気通信 API 通知は、ネットワークに登録されている外部システムからの受信通知を受信します。外部システムから通知を受信したら、イベント管理 アプリケーションを使用して応答用のイベントを作成できます。収集された情報に基づいて、イベント管理 はすべてのサービス影響度イベントの統合ビューを表示するダッシュボードを提供します。

電気通信 API 通知データモデル

次の図は、電気通信 API 通知のデータモデルのコンポーネントを示しています。

API 通知データモデル



Telecommunications API 通知を使用すると、パブリッシャー/サブスクライバー (Pub/Sub) サブスクリプションモデル、Hermes、Kafka Stream Connect などのイベント駆動型アーキテクチャを介して ServiceNow 受信通知を受信できます。クラウドのお客様は両方のアーキテクチャから柔軟に選択できますが、オンプレミスのお客様は独自のKafkaまたはPub/Subサブスクリプションモデルの使用に制限されています。

- Apache Kafka Stream のストリームコネクタの詳細については、「[Apache Kafka のストリームコネクタの使用](#)」を参照してください。
- Hermes Messaging Service の詳細については、「[Hermes メッセージングサービス](#)」を参照してください。

Pub/Sub モデルでは、受信通知はトピックに分類されます。ServiceNow を使用して受信通知をこれらのトピックに公開すると、サブスクライバー（顧客）はサブスクライブするトピックを柔軟に選択できます。このプロセスにより、購読者は自分の関心に一致するメッセージのみを選択できます。たとえば、外部システムからの受信メッセージに 10 個のトピックがある場合、顧客は要件に基づいてそのうちの 2 つに登録することを選択できます。したがって、外部システムから通知を受信すると、顧客が登録している 2 つのトピック専用のイベントが生成されます。

関連トピック

[電気通信 API 通知の構成](#)

[通信事業 API 通知とともにインストールされるシステムコンポーネント](#)

電気通信のヴィジビリティ

Telecom Visibility は、ITOM ヴィジビリティの検出および調整機能を拡張して、通信サービスプロバイダーの特定のニーズを満たします。リアルタイムのネットワークデータを検出し、不一致を調整し、CMDB と TNI (通信ネットワークインベントリ) で一貫した通信モデルを維持することで、正確で通信認識型ネットワークインベントリを実現します。

電気通信ドメイン全体で物理および論理ネットワークコンポーネントを検出し、検出データとインベントリデータの不一致を調整し、高度な自動化と保証のユースケースをサポートする電気通信固有の CMDB 構造を維持します。

Telecom Visibility の概要

Telecom Visibility は、電気通信ネットワークリソースを検出および管理するための統合ソリューションです。水平ディスカバリー、電気通信ディスカバリーパターン、サービスグラフコネクタ (SGC) などの実績のあるディスカバリーテクノロジーを活用して、さまざまなシステム (CLI/SNMP ベースのスタンドアロンデバイスや API 駆動型の EMS/NMS/コントローラーなど) から CMDB にネットワークデータを取り込みます。

Telecom Visibility は、通信サービスプロバイダー (CSP) や通信事業者に次のようなメリットをもたらします。

- 通信固有のネットワークインベントリの検出と維持
- 計画/設計インベントリと実際のネットワーク状態の不一致を調整する
- 自律ネットワーク運用 (ANO)、サービスフルフィルメント、および保証ワークフローをサポート

主要機能

TSOM の主な機能

機能	説明
パターンを使用した電気通信ディスカバリー	事前定義された電気通信ディスカバリーパターンを介して、SNMP および CLI プロトコルを使用して、ルーターやスイッチなどのスタンドアロンネットワーク機能 (xNF) を検出します。
サービスグラフコネクタの統合	ノースバウンド API を使用して、外部 EMS/NMS/コントローラーから CMDB にインベントリおよび構成データを取り込みます。
不一致の識別と調整	インベントリされたデータと検出されたデータの間の不一致 (CI の欠落、モデルの不一致、古いエントリなど) を検出して解決します。
通信中心のデータモデリング	インターフェイスカード、スロット、サブスロット、VLAN、ポートなど、構造化された階層型の通信データを維持します。

TSOM の主な機能 (続く)

機能	説明
CMDB 統合	CMDB 360 とディスカバリーアドミンワークスペースを使用して、ディスカバリーソース、属性変更、および調整結果を監視します。

アプリケーションとプラグイン

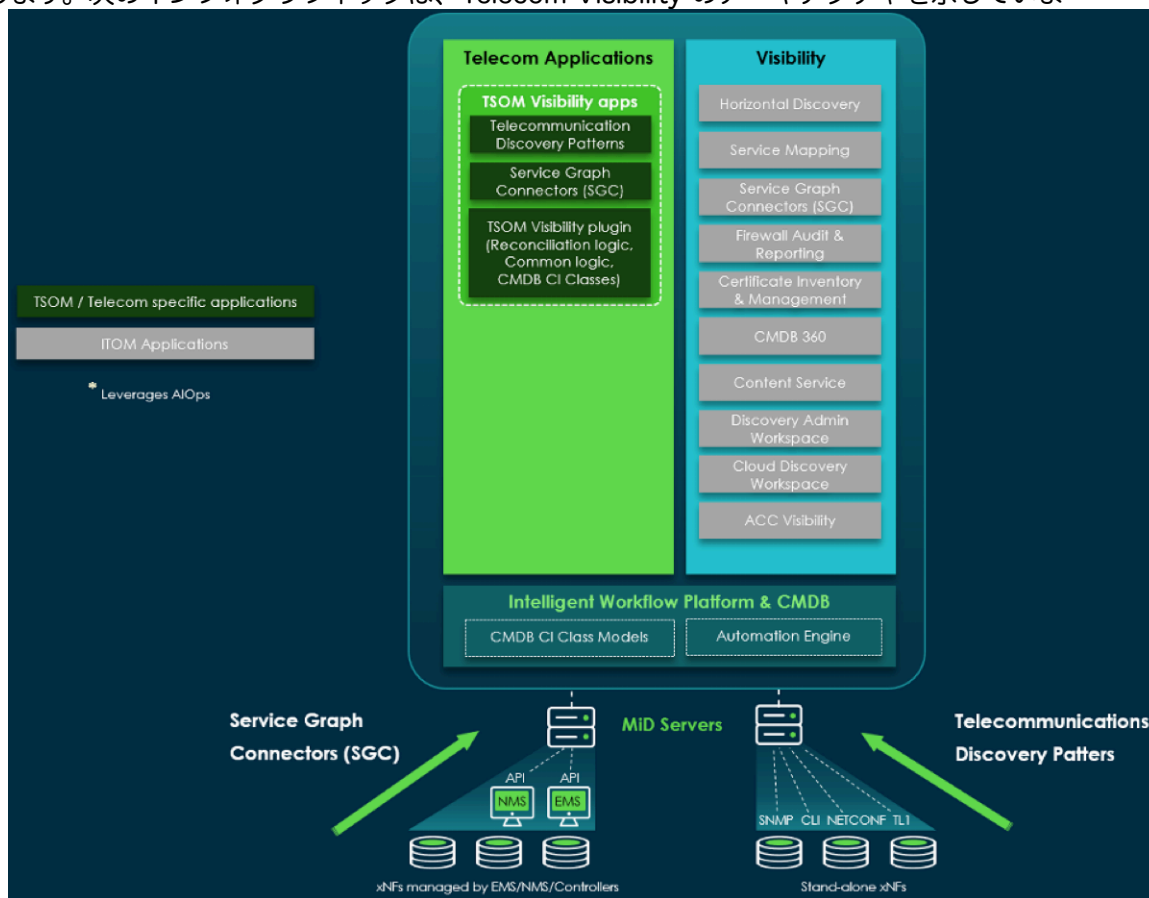
1. 通信の水平ディスカバリーパターン:SNMP/CLI を使用して、ルーターやスイッチなどの電気通信デバイス (xNF) を検出します。ライブネットワークインフラストラクチャをスキャンすることで、正確なインベントリの更新を可能にします。次のようなベンダー固有のパターンをサポートします。
 - Cisco(ASR1K、7613、Nexus 9000、Nexus 3548)
 - ジュニパー(MX80、MX104、MX240、MX460)
2. TSOM のサービスグラフコネクタ (SGC):API を介してインベントリデータをインポートするための、EMS/NMS/コントローラーとの事前構築された統合。楨：
 - スケジュール済みディスカバリーまたはオンデマンドディスカバリー
 - マルチインスタンス構成
 - Nokia Altiplano(GPONネットワーク)などのシステムとの統合
3. Telecom Visibility プラグイン:ディスカバリーと不一致調整の両方のための共有ロジックと基本コンポーネントを提供します。により：
 - TSOM ワークフロー全体で一貫した動作
 - 調整と修復のための再利用可能なコンポーネント
4. 電気通信の CMDB CI クラスモデル (必要なバージョン:1.69.0):CMDB を拡張して、次のような電気通信固有のクラスと関係を含めます。
 - インターフェイスカード
 - スロットとサブスロット
 - ネットワークインターフェイス

Telecom Visibility アーキテクチャ

Telecom Visibility は、直接ディスカバリー方法と間接ディスカバリー方法を組み合わせたものです。

- SNMP/CLI を使用したダイレクトディスカバリー (スタンドアロンルーター/スイッチなど)
- EMS/NMS/コントローラー (Nokia Altiplano など) からの API を使用した間接ディスカバリー

これらは CMDB と TNI にフィードされ、意思決定、自動化、およびネットワーク運用をサポートします。次のインフォグラフィックは、Telecom Visibility のアーキテクチャを示しています。



す。

ユースケース

Telecom Visibility のユースケース

ユースケース	結果
自律型ネットワーク運用	信頼性の高いネットワークインベントリは、AI/ML 主導の自動化をサポートします。
サービス遂行	正確なCMDB/TNIにより、注文の正確性と迅速なプロビジョニングが保証されます。
サービス保証	調整されたインベントリにより、トラブルシューティング時間が短縮され、アップタイムが増加します。
監査とコンプライアンス	ディスカバリーアドミンワークスペースと CMDB 360 は、監査の準備状況とトレーサビリティをサポートします。

主要なペルソナ

Telecom Visibility は以下によって使用されます。

- ネットワークおよびインフラストラクチャチーム:ベンダーやテクノロジー間でリアルタイムのネットワークリソースデータを検出して管理します。
- インベントリおよび履行チーム:正確で完全な CMDB/TNI を維持して、サービスフルフィルメント、注文管理、およびプロビジョニングをサポートします。
- プラットフォームアドミニストレーター:プラグインの設定、ディスカバリーソースの管理、調整ルールの検証、ディスカバリーパフォーマンスの監視を行います。

モニタリングツール

ディスカバリーアドミンワークスペース - セントラルコンソールで以下を行います。

- ディスカバリータスクの監視
- 調整プロセスの検証
- スケジュールの調整と診断の表示

CMDB 360 - 可視化:

- 属性レベルの更新履歴
- ソース属性
- 動作中の調整ルール

主な特長

Telecom Visibility の主なメリット

機能	利益
CLI/SNMP ディスカバリー	外部 API に依存せずにネットワークカバレッジを拡大します。
Nokia Altiplano 統合	複数のコントローラーインスタンスにわたる GPON ネットワークディスカバリーをサポートします。
不一致の調整	インベントリをクリーンで実用的なものにし、実際のネットワーク状態に即したものにします。

関連トピック

[Telecom Visibility を設定](#)

[ディスカバリーアドミンワークスペース](#)

電気通信の可視性とITOM ヴィジビリティ

通信ネットワークが進化し、よりハイブリッドで複雑になるにつれて、インフラストラクチャの可視性がこれまで以上に重要になっています。さまざまな環境固有のニーズを満たすために、ServiceNow AI Platform は ITOM ヴィジビリティと通信ビジビリティという 2 つの専用可視化ソリューションを提供しています。

これらは同じ基盤上に構築されていますが、それぞれが、従来の IT インフラストラクチャ向けの ITOM や通信ネットワーク向けの TSOM など、異なる世界に対応するように調整されています。

ITOM ヴィジビリティと電気通信ヴィジビリティはどちらも、ServiceNow ディスカバリーエンジン、識別および調整エンジン (IRE)、および CMDB の同じ基本機能に基づいて構築されています。どちらも次の機能を提供します。

- エージェントベースおよびエージェントレスディスカバリー。
- パターンベースの水平ディスカバリー。
- 検出されたデータの CMDB への調整。
- 依存関係マッピングと CI 関係の作成
- ディスカバリーアドミンワークスペースおよび CMDB 360 との統合。

この共有アーキテクチャにもかかわらず、提供するスコープ、ユースケース、およびデータモデルは異なります。

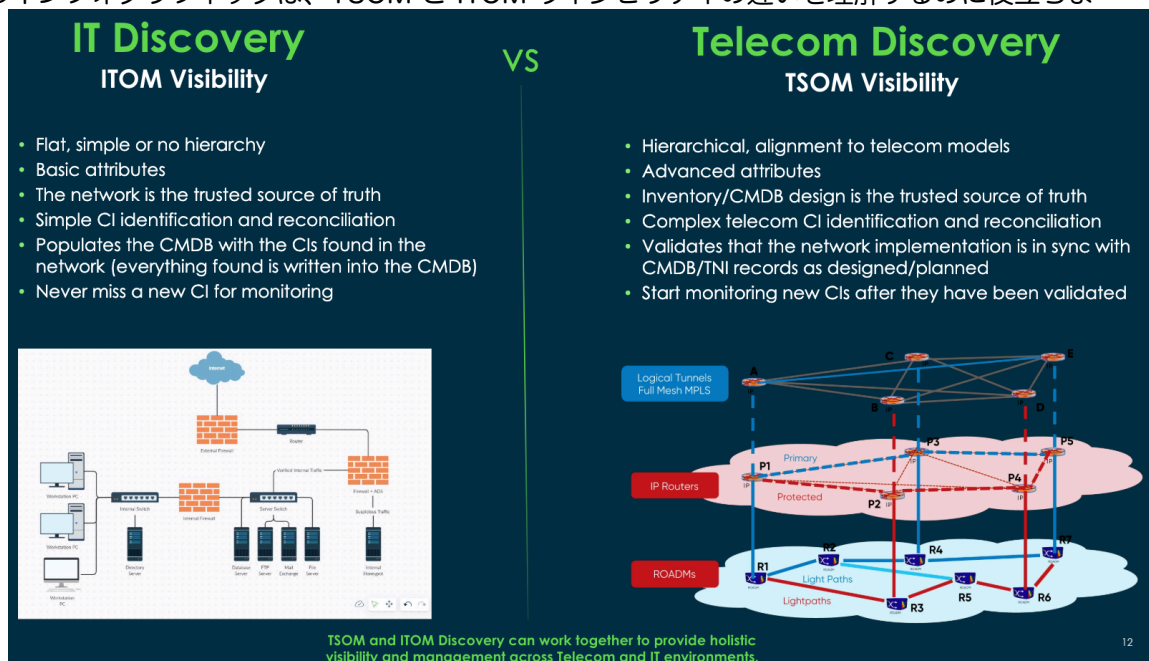
TSOM と ITOM ヴィジビリティの主な違い

電気通信の可視性とITOM ヴィジビリティ

機能/フォーカスエリア	ITOM ヴィジビリティ	電気通信のヴィジビリティ
ターゲット環境	従来の IT インフラストラクチャ (サーバー、アプリケーション、データベース、クラウドリソース)	電気通信インフラストラクチャ (xNF、ネットワーク要素、EMS/NMS コントローラー)
ディスカバリー方法	IT パターンによる水平ディスカバリー	電気通信ディスカバリーパターン (SNMP/CLI) と通信用サービスグラフコネクタを使用した水平ディスカバリー
CMDB モデル	ITOM CMDB クラス (Windows Server、アプリケーション、ネットワークアダプタなど)	テレコム対応 CMDB クラスおよびテレコムネットワークインベントリ (TNI) (インターフェイスカード、スロット、ラグ、サブスロット、VLAN など)
必要なプラグイン	com.snc.discovery および com.snc.itom.visibility	sn_tsom_core、sn_tsom_patterns、および通信固有の SGC プラグイン (sn_sgc_altiplano_connector など)
ユースケースの焦点	アプリケーション依存関係マッピング、サービスモデリング、クラウドインフラストラクチャディスカバリー	通信ネットワークインベントリの検出、調整、自律的なネットワーク運用
不一致の処理	一般的な IRE 調整ルール	電気通信固有の不一致の識別と調整 (階層の不一致、属性レベルの競合など)
ベンダーデータの取り込み	主に検出パターン経由	SGC (EMS/NMS/コントローラー) を使用したノースバウンド API 統合を重視
サポートされるネットワークタイプ	エンタープライズネットワーク、データセンター、クラウド	マルチベンダー、マルチドメイン通信ネットワーク (RAN、コア、トランスポート、アクセス)

インフォグラフィック

次のインフォグラフィックは、TSOM と ITOM ヴィジビリティの違いを理解するのに役立ちま



す。

主な特長

次の場合に ITOM ヴィジビリティを使用します。

- IT インフラストラクチャコンポーネント (Windows サーバー、クラウド VM、データベース、ロードバランサーなど) を検出しています。
- 主な目標には、サービスマッピング、オペレーショナルレジリエンス、またはクラウドの最適化が含まれます。
- ITSM、ITOM、または DevOps のユースケースに焦点を当てている。

次の場合に電気通信の可視性を使用します。

- 従来の IT システムで管理されていないデバイスを含む、通信ネットワークインフラストラクチャを検出しています。
- カード、ポート、サブスロット、LAG などの電気通信固有のネットワーク階層を扱っています。
- 信頼できるデータソースとして EMS/NMS/コントローラーに依存しています。
- 電気通信インベントリモデルに合わせて調整された不一致の検出と調整が必要です。
- TM Forum の標準に準拠し、自律的なネットワーク運用をサポートしたり、クローズドループ保証を有効にしたりします。

例

ユースケース	ITOM ヴィジビリティ	電気通信のヴィジビリティ
AWS での仮想マシンのフリートの検出	はい	いいえ
API を使用して EMS からルーターとスイッチのデータを取り込む	いいえ	はい

アプリケーション間の依存関係を特定する	はい	いいえ
テレコムカード階層の不一致を検出して調整	いいえ	はい

ITOM ヴィジビリティとテレコムヴィジビリティはどちらも正確な CMDB の入力と維持に役立ちますが、さまざまなドメイン向けに最適化されています。ITOM ヴィジビリティはエンタープライズ IT 環境を対象としており、テレコムヴィジビリティはテレコムインフラストラクチャの検出、不一致管理、およびインベントリ調整の特殊なニーズに合わせて調整されています。

適切な可視化ソリューションを選択するか、両方を併用することで、IT と通信の両方のランドスケープにわたって、信頼できるドメイン固有の運用の可視性を確実に維持できます。

電気通信ディスカバリー

ServiceNow AI Platform Telecom ディスカバリーは、ITOM ヴィジビリティの機能を拡張して電気通信固有のユースケースをサポートすることで、電気通信ネットワークインフラストラクチャを包括的に可視化するのに役立ちます。

通信サービスプロバイダー (CSP) 向けに構築されたこのソリューションは、標準化されたプロトコルとネットワーク管理システムとの統合を使用して、マルチベンダー環境全体でネットワーク要素のディスカバリーとマッピングを可能にします。

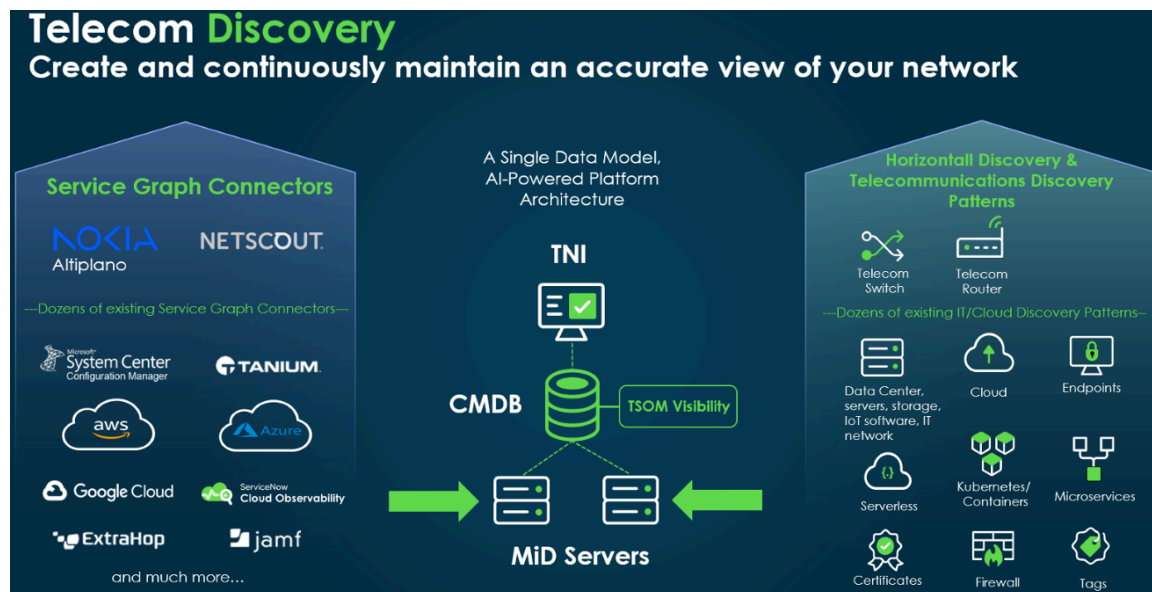
Telecom Discovery プラグインをサービスグラフコネクタおよびディスカバリーパターンの機能と組み合わせることで、構成管理データベース (CMDB) に電気通信リソースの正確なレコードを自動的に入力して維持し、IT インフラストラクチャとネットワークインフラストラクチャの両方の統一されたビューを提供できます。

- i** 注: Telecom Discovery は TSOM Visibility サブスクリプションの一部であり、TM Forum Autonomous Network Operations (ANO) フレームワークと連携しています。

Telecom ディスカバリーでは、次のことができます。

- ドメインやベンダー間で物理および論理の電気通信ネットワークリソースを検出します。
- 要素管理システム (EMS)、ネットワーク管理システム (NMS)、および SDN コントローラーと統合します。
- リアルタイムのネットワークデータに基づいて CMDB/TNI レコードを自動的に入力および更新します。
- SNMP およびコマンドラインインターフェイス (CLI) を使用してスタンドアロン xNF を検出します。
- サービスグラフコネクタと特殊なディスカバリーパターンを使用して CMDB データを拡張します。
- 検出されたネットワークデータとインベントリレコードの間の不一致を特定します。
- 一貫性のある正確なインフラストラクチャの可視化を通じて、自動化のユースケースをサポートします。

Telecom Discovery のアーキテクチャ



ITOM ヴィジビリティとの統合

Telecom ディスカバリーは、既存の ITOM ヴィジビリティ 機能を補完するように設計されています。以下のことを行うことができます。

- TSOM プラグインと一緒に水平ディスカバリーと ITOM 機能を活用します。
- IT リソースおよび電気通信リソースに対して一貫したディスカバリープラクティスを維持します。
- 同じ CMDB データモデルを使用して、クロスドメインサービスのヴィジビリティを管理します。

この統合により、IT ドメインとネットワークドメインの両方にわたる統合資産管理、問題解決の迅速化、運用の合理化が確認されます。

ローコード/ノーコードツールによるカスタマイズ

コードを記述せずにディスカバリーロジックを拡張できる直感的な設計ツールを提供します。以下のことを行うことができます。

- カスタムサービスグラフコネクタをビルドまたは変更します。
- 電気通信ディスカバリーパターンを拡張してベンダー固有の要件に合わせる。
- 新しいデバイスタイプとネットワークドメインのオンボーディングを加速します。

このアプローチにより、CSP は俊敏性を維持し、検出フットプリントを拡大する際に価値実現までの時間を短縮できます。

主要コンポーネント

- 電気通信ディスカバリーパターン (sn_tsom_patterns): スタンドアロンルーター、スイッチ、および xNF の SNMP ベースのディスカバリー用のパターンを提供します。Cisco および Juniper 固有のディスカバリーロジックが含まれています。
- Nokia Altiplano のサービスグラフコネクタ (sn_sgc_altiplano_connector): REST API を介して Nokia Altiplano アクセス SDN コントローラーからのデータ収集を有効にします。
- Telecom Core (sn_tsom_core): 不一致識別、修正ロジック、共有電気通信ディスカバリー機能などの基本的な機能を提供します。

Telecom Discovery Builder フレームワーク

Telecom Discovery Builder フレームワーク ETL (抽出、変換、ロード) は、電気通信サービスオペレーション管理 (TSOM) Core アプリケーションとともに提供される、再利用可能でスキーマに合わせたコンポーネントです。これは、複数のサービスグラフコネクタ (SGC) にまたがる電気通信インベントリデータを ServiceNow 構成管理データベース (CMDB) に取り込むための一貫した拡張可能な方法を提供します。

Telecom Discovery Builder フレームワーク ETL は、電気通信固有の構成アイテム (CI) データを処理するために設計されたベースラインデータ取り込みユーティリティとして機能します。これにより、コネクタ開発チームは、コネクタごとに複製およびカスタマイズできる標準化された変換ロジックを提供することで、ETL をゼロからビルドする必要がなくなります。

TSOM Core プラグインがアクティブ化されると、汎用 ETL が自動的にプロビジョニングされ、統合ハブ ETL Studio で利用できるようになり、すぐに再利用してコネクタ固有のニーズに適合させることができます。

Telecom Discovery Builder フレームワークを使用する理由とタイミング

新しい電気通信コネクタごとに ETL ロジックまたは変換マップをゼロからビルドすると、時間がかかり、エラーが発生しやすく、一貫性がなくなる可能性があります。Telecom Discovery Builder フレームワークは、実装全体で一貫性を確保する、標準化された再利用可能な基盤を提供することで、プロセスを簡素化します。

次の場合は、Telecom ディスカバリービルダーフレームワークを使用します。

- 電気通信環境に新しいサービスグラフコネクタを展開し、ETL を最初からビルドすることを避けたい。
- 複数のコネクタとプラットフォームインスタンス間でスキーマに合わせた整合性を維持します。
- TNI (テレコムネットワークインベントリ) に準拠したディスカバリーペイロードと CI 関係。詳細については、「[通信事業ネットワークインベントリ](#)」を参照してください。
- SGC (サービスグラフコネクタ) アプリケーションスコープ内で作業し、コアロジックを変更せずに ETL の動作をカスタマイズしたい。
- 事前定義されたマッピング、検証済みの JSON スキーマのサポート、および UI 駆動型の構成インターフェイスを利用できます。

主な機能

- TSOM コアでの自動プロビジョニング: 自動的にインストールされ、電気通信コネクタ全体で使用する準備が整います。
- 汎用スキーマベースの取り込み: 電気通信 CI の統一データスキーマをサポートします。
- 複製とカスタマイズのサポート: 統合ハブ ETL Studio を使用して、Telecom Discovery Builder フレームワーク ETL をアプリケーションスコープに複製します。

- TNI のサポート:TNI データモデルに合わせて調整され、ETL ロジックを拡張して TNI エンティティを生成およびリンクできます。
- 柔軟なフィールドマッピングインターフェイス:UI 主導のエクスペリエンスを通じて、インポートセット、データソース、ターゲット、および変換ロジックを構成します。

主な特長

Telecom Discovery Builder フレームワーク ETL を使用して、次のものを取得します。

- 電気通信構成アイテム (CI) を構造化して CMDB にロードする方法の一貫性。
- コネクタ間での複製とカスタマイズによる再利用性。
- TNI スキーマおよびディスカバリーモデルの要件への準拠。
- ベーススキーマを損なうことなくデータ変換を拡張および調整できる柔軟性。

関連トピック

- コネクタでの Telecom Discovery Builder フレームワーク ETL の構成
- 重複した ETL に対する TNI エンティティサポートの拡張

ディスカバリーパターンを使用した直接ディスカバリー

Telecommunications Discovery Patterns プラグイン (TSOM パターンとも呼ばれる) は、従来のネットワーク管理システムに依存することなく、ルーターやスイッチなどのスタンドアロンネットワーク要素を直接検出できるように ServiceNow AI Platform Telecom ディスカバリーを拡張します。これらのパターンにより、通信サービスプロバイダー(CSP)は、NETCONF をサポートする SNMP や CLI などのプロトコルを使用して、マルチベンダー xNF を識別してマッピングできます。

電気通信ディスカバリーパターンは、ネットワーク要素から直接電気通信ネットワークリソースを検出および管理するための強力なパターンベースのアプローチを提供します。これらのパターンは、従来の EMS/NMS システムで管理されていないスタンドアロン xNF(ルーターやスイッチなど)を検出する場合に特に役立ちます。

この機能により、電気通信インフラストラクチャ全体の可視性が向上し、TNI データモデルに従って、ベンダーニュートラルおよびベンダー固有のデバイスデータがキャプチャされ、CMDB および通信ネットワークインベントリ (TNI) に反映されます。

- 📌 注: 通信ディスカバリーパターンは TSOM ヴィジビリティサブスクリプションの一部であり、ServiceNow Store から顧客に見えるプラグインとして入手できます。

主な機能

- 直接ネットワーク要素ディスカバリー
 - SNMP、CLI、および NETCONF を使用して、物理ネットワーク要素と直接通信します。
 - EMS/NMS 仲介に依存せずに、インターフェイスカード、スロット、ポート、デバイスなどの物理ネットワークインベントリを検出します。
- 自動 CI マッピングと CMDB の統合
 - 検出されたデータは、通信事業者向けの構成アイテム (CI) クラスに自動的にマッピングされます。
 - 識別および調整エンジン (IRE) と直接統合して、CMDB およびテレコムネットワークインベントリ (TNI) に正確で重複のない CI レコードを確保します。
- CMDB の精度とコンプライアンス

- パターンは、CMDB コンプライアンス認定監査をトリガーして、不一致や古いレコードを検出します。
- 不一致の特定と修正をサポートし、データの整合性を維持するのに役立ちます。
- ローコードパターンのカスタマイズ:特定のネットワークアーキテクチャとビジネスニーズを満たすために、ローコードパターンデザイナーを使用してディスカバリーパターンを拡張またはカスタマイズします。

仕組み

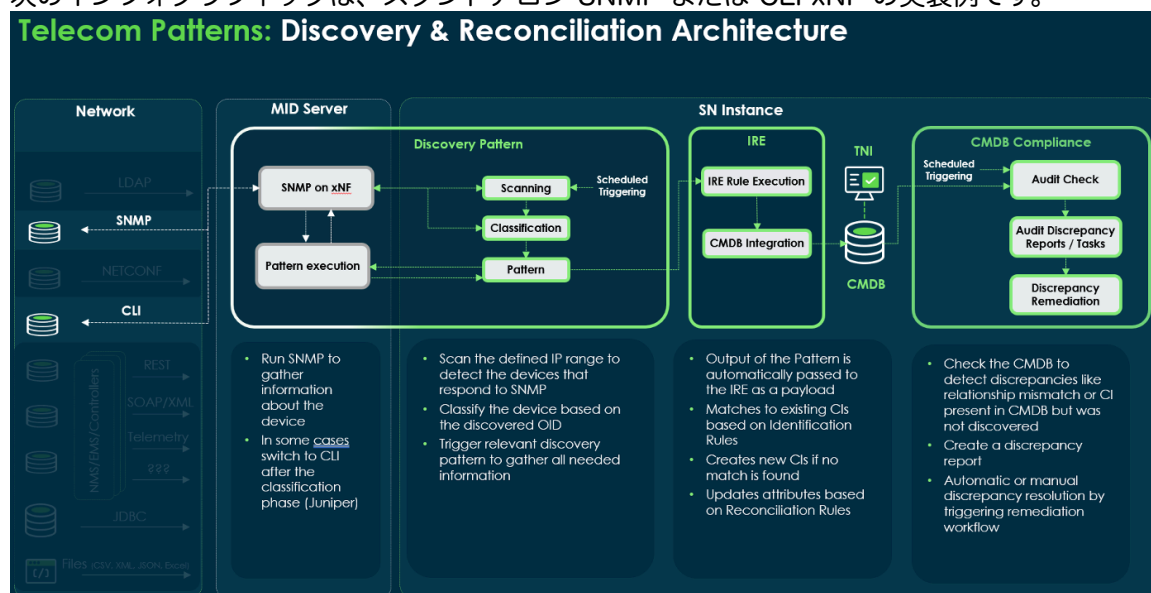
電気通信ディスカバリーパターンは、Nebula Discovery Language (NDL) を使用した水平ディスカバリーによって実行されます。パターンは、次のような一連のステップです。

1. ターゲット・デバイスへの接続を確立します。
2. SNMP、CLI、またはその両方を使用してコマンドを実行します。
3. CI 属性と関係性を抽出します。
4. 調整と CMDB および TNI への挿入のために結果を IRE に送信します。

ディスカバリーロジックは、通信モデリングのベストプラクティスに合わせて複雑な CI 関係 (カードオンカードシナリオなど) を再ビルドする TNI データモデルに従います。たとえば、親カード内で子カードが検出された場合、Telecom ディスカバリーはサブロットを合成して子を挿入し、無効なカードオンカード構成を回避します。

水平ディスカバリーと電気通信ディスカバリーパターンを使用したアーキテクチャ

次のインフォグラフィックは、スタンドアロン SNMP または CLI xNF の実装例です。



水平ディスカバリーアプリケーション

ServiceNow の水平ディスカバリーアプリケーションは、ネットワーク、IT、クラウド環境全体で効果的に動作するように設計された、汎用性が高く拡張性の高いディスカバリーエンジンであり、複数のレイヤーにわたるデータを収集してインフラストラクチャの全体像を提供します。

詳細については、「[パターンを使用した水平ディスカバリープロセスフロー](#)」を参照してください。

サポートされているディスカバリーパターン

パターンは、構成アイテム (CI) とその送信接続の属性を検出するために設計された一連のコマンドです。Telecom ディスカバリーには、幅広いネットワーク要素をカバーする一連の事前設定されたパターンが用意されています。ServiceNow には、次のようなすぐに利用可能な (OOB) TSOM パターンがいくつか用意されています。

- 電気通信ルーターパターン:汎用 SNMP ベースのルーターディスカバリー。
- Telecom Cisco 7613 ルータ パターン:SNMP を使用する Cisco 7613 ルータ用。
- Telecom Juniper MX SSH ルーターパターン:Juniper MX ルーターの SNMP + CLI ディスカバリー。
- 電気通信 Cisco スイッチパターン:SNMP ベースの Cisco スイッチディスカバリー。
- 電気通信スイッチパターン:SNMP ベースの汎用スイッチディスカバリー。

TNI エンティティの作成

インスタンスに Telecom Network Inventory (TNI) プラグインがインストールされている場合:

- 検出されたすべての CI は、自動的に TNI エンティティレコードになります。
- IRE ペイロードには、cmdb_ci テーブルと tni_entity テーブルの両方へのマッピングが含まれています。

これにより、受注処理、保証、およびネットワーク計画に不可欠な、運用システムとインベントリシステム間のシームレスな調整が確保されます。

MID サーバー

MID サーバー は、ローカルネットワーク内のサーバー上で Windows サービスまたは UNIX デーモンとして動作する Java アプリケーションです。この ServiceNow MID サーバー により、ServiceNow インスタンスと外部のアプリケーション、データソース、サービスとの間の通信やデータ転送が容易になります。

詳細については、「[MID サーバー](#)」を参照してください。

識別および調整エンジン (IRE)

IRE は、複数のソースからのデータを識別および調整するための一元化されたフレームワークを提供します。さまざまなデータソースを使用して CI レコードを作成または更新するときに、CMDB および一部の非 CMDB テーブルの整合性を検証します。

関連トピック

[水平ディスカバリーのインストールとディスカバリーパターンの設定](#)

電気通信ルーターパターン

ServiceNow Telecom ディスカバリーアプリケーションは、Telecom ルーターディスカバリーパターンを使用して、ネットワーク内の SNMP ベースのルーターを検索します。これらのリソースを検出するには、ServiceNow ストアから電気通信ディスカバリーパターン (TSOM パターン) を更新する必要があります。

電気通信ディスカバリーは、電気通信ディスカバリーパターンを使用して水平ディスカバリーを実行します。この電気通信ディスカバリーパターンでは、一連の SNMP 要求を使用して、ネットワーク要素を検索、分類、および検出します。

電気通信ルーターパターンは、TSOM ヴィジビリティの一部である電気通信ディスカバリーパターンアプリケーション (sn_tsom_patterns) の一部です。

ストアでアプリを要求する

ServiceNow Store Web サイトにアクセスして利用可能なすべてのアプリを表示し、ストアにリクエストを送信する方法について確認してください。リリースされたすべてのアプリのリリースノート情報については、「[ServiceNow Storeバージョン履歴のリリースノート](#)」を参照してください。

必須条件

- TSOM のサブスクリプション。
- ネットワークルーターデバイスに SNMP アクセス権があることを確認します。
- ServiceNow インスタンスで、SNMP 認証情報を構成します。詳細については、「[ディスカバリーにおける SNMP サポート](#)」を参照してください。
- TSOM パターンの設定については、[水平ディスカバリーのインストールとディスカバリーパターンの設定](#)を参照してください。

影響を受ける CMDB CI と CI 関係 (物理レイヤー)

CI	CI リレーションシップ
IP ルーター CI	<p>IP ルーターデバイスは、IP ルーター CI によって表されます。</p> <p>テーブル名: cmdb_ci_ip_router</p> <p>IP ルーター CI にはスロットまたはネットワークインターフェイスが含まれています。</p>
スロット CI	<p>スロットはスロット CI によって表されます。</p> <p>テーブル名: cmdb_ci_container_slot</p> <p>スロットは IP ルータに含まれています。</p> <p>スロット CI にはインターフェイスカードが含まれています。</p>
サブスロット CI	<p>サブスロットは、サブスロット CI によって表されます。</p> <p>テーブル名: cmdb_ci_container_subslot</p> <p>サブスロットはインターフェイスカード CI に含まれています。</p> <p>サブスロット CI には、インターフェイスカード CI が含まれています。</p>

CI	CI リレーションシップ
インターフェイスカード CI	<p>さまざまなタイプのカードがインターフェイスカード CI によって表されます。</p> <p>ファンと電源ユニットもカード CI で表されません。</p> <p>テーブル名: cmdb_ci_interface_card</p> <p>インターフェイスカードは、スロットまたはサブスロットに含まれています。</p> <p>インターフェイスカードには、ネットワークインターフェイスまたはサブスロットを含めることができます。</p>
ネットワークインターフェイス CI	<p>あらゆるタイプのネットワークインターフェイスがネットワークインターフェイス CI によって表されます。</p> <p>テーブル名: cmdb_ci_ni_interface</p> <p>ネットワークインターフェイスは、インターフェイスカード、IP ルーターに含まれています。</p>

Telecom Cisco 7613 ルータパターン

ServiceNow Telecom ディスカバリーアプリケーションは、Telecom Cisco 7613 ルーターディスカバリーパターンを使用して、ネットワーク内の SNMP ベースの Cisco 7613 を検索します。これらのリソースを検出するには、ServiceNow ストアから電気通信ディスカバリーパターン (TSOM パターン) を更新する必要があります。

電気通信ディスカバリーは、電気通信ディスカバリーパターンを使用して水平ディスカバリーを実行します。この電気通信ディスカバリーパターンでは、一連の SNMP 要求を使用して、ネットワーク要素を検索、分類、および検出します。

Telecom Cisco 7613 ルータパターンは、TSOM ヴィジビリティの一部である Telecommunications Discovery Patterns アプリケーション (sn_tsom_patterns) の一部です。

ストアでアプリを要求する

[ServiceNow Store](#) Web サイトにアクセスして利用可能なすべてのアプリを表示し、ストアにリクエストを送信する方法について確認してください。リリースされたすべてのアプリのリリースノート情報については、「[ServiceNow Storeバージョン履歴のリリースノート](#)」を参照してください。

必須条件

- TSOM のサブスクリプション。
- ネットワークルーターデバイスに SNMP アクセス権があることを確認します。

- ServiceNow インスタンスで、SNMP 認証情報を構成します。詳細については、「[ディスカバリーにおける SNMP サポート](#)」を参照してください。
- TSOM パターンの設定については、[水平ディスカバリーのインストールとディスカバリーパターンの設定](#)を参照してください。

影響を受ける **CMDB CI** と **CI** 関係 (物理レイヤー)

CI	CI リレーションシップ
IP ルーター CI	<p>IP ルーターデバイスは、IP ルーター CI によって表されます。</p> <p>テーブル名: cmdb_ci_ip_router</p> <p>IP ルーター CI にはスロットまたはネットワークインターフェイスが含まれています。</p>
スロット CI	<p>スロットはスロット CI によって表されます。</p> <p>テーブル名: cmdb_ci_container_slot</p> <p>スロットは IP ルータに含まれています。</p> <p>スロット CI にはインターフェイスカードが含まれています。</p>
サブスロット CI	<p>サブスロットは、サブスロット CI によって表されます。</p> <p>テーブル名: cmdb_ci_container_subslot</p> <p>サブスロットはインターフェイスカード CI に含まれています。</p> <p>サブスロット CI には、インターフェイスカード CI が含まれています。</p>
インターフェイスカード CI	<p>さまざまなタイプのカードがインターフェイスカード CI によって表されます。</p> <p>ファンと電源ユニットもカード CI で表されません。</p> <p>テーブル名: cmdb_ci_interface_card</p> <p>インターフェイスカードは、スロットまたはサブスロットに含まれています。</p> <p>インターフェイスカードには、ネットワークインターフェイスまたはサブスロットを含めることができます。</p>

CI	CI リレーションシップ
ネットワークインターフェイス CI	<p>あらゆるタイプのネットワークインターフェイスがネットワークインターフェイス CI によって表されます。</p> <p>テーブル名: cmdb_ci_ni_interface</p> <p>ネットワークインターフェイスは、インターフェイスカード、IP ルーターに含まれています。</p>

Telecom Juniper MX SSH ルーターパターン

ServiceNow Telecom ディスカバリーアプリケーションは、Telecom Juniper MX SSHRouter ディスカバリーパターンを使用して、ネットワーク内の SNMP および CLI ベースの Juniper MX シリーズルーターを検索します。これらのリソースを検出するには、ServiceNow ストアから電気通信ディスカバリーパターン (TSOM パターン) を更新する必要があります。

電気通信ディスカバリーは、電気通信ディスカバリーパターンを使用して水平ディスカバリーを実行します。この電気通信ディスカバリーパターンでは、一連の SNMP 要求を使用して SSH 経由で CLI を検索および分類し、ネットワーク要素を検出します。

Telecom Juniper MX SSH ルーターパターンは、TSOM ヴィジビリティの一部である Telecommunications Discovery Patterns アプリケーション (sn_tsom_patterns) の一部です。

ストアでアプリを要求する

[ServiceNow Store](#) Web サイトにアクセスして利用可能なすべてのアプリを表示し、ストアにリクエストを送信する方法について確認してください。リリースされたすべてのアプリのリリースノート情報については、「[ServiceNow Storeバージョン履歴のリリースノート](#)」を参照してください。

必須条件

- TSOM のサブスクリプション。
- ネットワークルーターデバイスに SNMP アクセス権があることを確認します。
- ServiceNow インスタンスで、SNMP 認証情報を構成します。詳細については、「[ディスカバリーにおける SNMP サポート](#)」を参照してください。
- TSOM パターンの設定については、[水平ディスカバリーのインストールとディスカバリーパターンの設定](#)を参照してください。

影響を受ける CMDB CI と CI 関係 (物理レイヤー)

CI	CI リレーションシップ
IP ルーター CI	<p>IP ルーターデバイスは、IP ルーター CI によって表されます。</p> <p>テーブル名: cmdb_ci_ip_router</p> <p>IP ルーター CI にはスロットまたはネットワークインターフェイスが含まれています。</p>

CI	CI リレーションシップ
<p>スロット CI</p>	<p>スロットはスロット CI によって表されます。</p> <p>テーブル名: cmdb_ci_container_slot</p> <p>スロットは IP ルータに含まれています。</p> <p>スロット CI にはインターフェイスカードが含まれています。</p>
<p>サブスロット CI</p>	<p>サブスロットは、サブスロット CI によって表されます。</p> <p>テーブル名: cmdb_ci_container_subslot</p> <p>サブスロットはインターフェイスカード CI に含まれています。</p> <p>サブスロット CI には、インターフェイスカード CI が含まれています。</p>
<p>インターフェイスカード CI</p>	<p>さまざまなタイプのカードがインターフェイスカード CI によって表されます。</p> <p>ファンと電源ユニットもカード CI で表されます。</p> <p>テーブル名: cmdb_ci_interface_card</p> <p>インターフェイスカードは、スロットまたはサブスロットに含まれています。</p> <p>インターフェイスカードには、ネットワークインターフェイスまたはサブスロットを含めることができます。</p>
<p>ネットワークインターフェイス CI</p>	<p>あらゆるタイプのネットワークインターフェイスがネットワークインターフェイス CI によって表されます。</p> <p>テーブル名: cmdb_ci_ni_interface</p> <p>ネットワークインターフェイスは、インターフェイスカード、IP ルーターに含まれています。</p>

電気通信 Cisco スイッチパターン

ServiceNow Telecom ディスカバリーアプリケーションは、Telecom Cisco スイッチディスカバリーパターンを使用して、ネットワーク内の SNMP ベースの Cisco スイッチを検索します。これらのリソースを検出するには、ServiceNow ストアから電気通信ディスカバリーパターン (TSOM パターン) を更新する必要があります。

電気通信ディスカバリーは、電気通信ディスカバリーパターンを使用して水平ディスカバリーを実行します。この電気通信ディスカバリーパターンでは、一連の SNMP 要求を使用して、ネットワーク要素を検索、分類、および検出します。

電気通信 Cisco スイッチパターンは、TSOM ヴィジビリティの一部である電気通信ディスカバリーパターンアプリケーション (sn_tsom_patterns) の一部です。

ストアでアプリを要求する

ServiceNow Store Web サイトにアクセスして利用可能なすべてのアプリを表示し、ストアにリクエストを送信する方法について確認してください。リリースされたすべてのアプリのリリースノート情報については、「[ServiceNow Storeバージョン履歴のリリースノート](#)」を参照してください。

必須条件

- TSOM のサブスクリプション。
- ネットワークルーターデバイスに SNMP アクセス権があることを確認します。
- ServiceNow インスタンスで、SNMP 認証情報を構成します。詳細については、「[ディスカバリーにおける SNMP サポート](#)」を参照してください。
- TSOM パターンの設定については、[水平ディスカバリーのインストールとディスカバリーパターンの設定](#)を参照してください。

影響を受ける CMDB CI と CI 関係 (物理レイヤー)

CI	CI リレーションシップ
IP スイッチ CI	<p>IP スイッチデバイスは、IP スイッチ CI によって表されます。</p> <p>テーブル名: cmdb_ci_ip_switch</p> <p>IP スイッチ CI には、スロットまたはネットワークインターフェイスが含まれています。</p>
スロット CI	<p>スロットはスロット CI によって表されます。</p> <p>テーブル名: cmdb_ci_container_slot</p> <p>スロットは IP スイッチに含まれます。</p> <p>スロット CI にはインターフェイスカードが含まれています。</p>
サブスロット CI	<p>サブスロットは、サブスロット CI によって表されます。</p> <p>テーブル名: cmdb_ci_container_subslot</p> <p>サブスロットはインターフェイスカード CI に含まれています。</p>

CI	CI リレーションシップ
	サブスロット CI には、インターフェイスカード CI が含まれています。
インターフェイスカード CI	<p>さまざまなタイプのカードがインターフェイスカード CI によって表されます。</p> <p>ファンと電源ユニットもカード CI で表されません。</p> <p>テーブル名: cmdb_ci_interface_card</p> <p>インターフェイスカードは、スロットまたはサブスロットに含まれています。</p> <p>インターフェイスカードには、ネットワークインターフェイスまたはサブスロットを含めることができます。</p>
ネットワークインターフェイス CI	<p>あらゆるタイプのネットワークインターフェイスがネットワークインターフェイス CI によって表されます。</p> <p>テーブル名: cmdb_ci_ni_interface</p> <p>ネットワークインターフェイスは、インターフェイスカード、IP スイッチに含まれています。</p>

電気通信スイッチパターン

ServiceNow Telecom ディスカバリーアプリケーションは、電気通信スイッチディスカバリーパターンを使用して、ネットワーク内の SNMP ベースの電気通信スイッチを検索します。これらのリソースを検出するには、ServiceNow ストアから電気通信ディスカバリーパターン (TSOM パターン) を更新する必要があります。

電気通信ディスカバリーは、電気通信ディスカバリーパターンを使用して水平ディスカバリーを実行します。この電気通信ディスカバリーパターンでは、一連の SNMP 要求を使用して、ネットワーク要素を検索、分類、および検出します。

電気通信スイッチパターンは、TSOM ヴィジビリティの一部である電気通信ディスカバリーパターンアプリケーション (sn_tsom_patterns) の一部です。

ストアでアプリを要求する

[ServiceNow Store](#) Web サイトにアクセスして利用可能なすべてのアプリを表示し、ストアにリクエストを送信する方法について確認してください。リリースされたすべてのアプリのリリースノート情報については、「[ServiceNow Storeバージョン履歴のリリースノート](#)」を参照してください。

必須条件

- TSOM のサブスクリプション。
- ネットワークルーターデバイスに SNMP アクセス権があることを確認します。

- ServiceNow[®] インスタンスで、SNMP 認証情報を構成します。詳細については、「[ディスカバリーにおける SNMP サポート](#)」を参照してください。
- TSOM パターンの設定については、[水平ディスカバリーのインストールとディスカバリーパターンの設定](#)を参照してください。

影響を受ける **CMDB CI** と **CI** 関係 (物理レイヤー)

CI	CI リレーションシップ
IP スイッチ CI	<p>IP スイッチデバイスは、IP スイッチ CI によって表されます。</p> <p>テーブル名: cmdb_ci_ip_switch</p> <p>IP スイッチ CI には、スロットまたはネットワークインターフェイスが含まれています。</p>
スロット CI	<p>スロットはスロット CI によって表されます。</p> <p>テーブル名: cmdb_ci_container_slot</p> <p>スロットは IP スイッチに含まれます。</p> <p>スロット CI にはインターフェイスカードが含まれています。</p>
サブスロット CI	<p>サブスロットは、サブスロット CI によって表されます。</p> <p>テーブル名: cmdb_ci_container_subslot</p> <p>サブスロットはインターフェイスカード CI に含まれています。</p> <p>サブスロット CI には、インターフェイスカード CI が含まれています。</p>
インターフェイスカード CI	<p>さまざまなタイプのカードがインターフェイスカード CI によって表されます。</p> <p>ファンと電源ユニットもカード CI で表されません。</p> <p>テーブル名: cmdb_ci_interface_card</p> <p>インターフェイスカードは、スロットまたはサブスロットに含まれています。</p> <p>インターフェイスカードには、ネットワークインターフェイスまたはサブスロットを含めることができます。</p>

CI	CI リレーションシップ
ネットワークインターフェイス CI	あらゆるタイプのネットワークインターフェイスがネットワークインターフェイス CI によって表されます。 テーブル名: cmdb_ci_ni_interface ネットワークインターフェイスは、インターフェイスカード、IP スイッチに含まれています。

サービスグラフコネクタを使用した間接ディスカバリー

サービスグラフコネクタ (SGC) を使用する ServiceNow Telecom ディスカバリーを使用すると、外部管理システム (EMS、NMS、SDN コントローラーなど) からのネットワークインフラストラクチャデータを構成管理データベース (CMDB) にシームレスに統合できます。

このアプローチは、通信サービスプロバイダー (CSP) がマルチベンダーの電気通信ネットワークのリソース、サービス、および構成の最新かつ正確なビューを維持するのに役立ちます。

事前定義されたコネクタと堅牢なデータ変換ツールを活用することで、ドメイン間でインベントリを統合し、CMDB と通信ネットワークインベントリ (TNI) に電気通信データモデルに沿ったリアルタイムのネットワークインサイトを確実に反映させることができます。

i 注: 電気通信サービスグラフコネクタは TSOM ヴィジビリティサブスクリプションの一部であり、標準のサービスグラフコネクタフレームワークの機能を拡張します。

サービスグラフコネクタの概要

電気通信サービスポータル SGC では、次のことができます。

- ノースバウンド REST API を使用して、EMS/NMS/コントローラーからデータを取り込みます。
- CMDB および TNI レコードに、拡張された通信関連データを自動的に入力して更新します。
- 識別および調整エンジン (IRE) を使用して、受信データを既存の CI レコードと調整します。
- TNI プラグインのインストール時に、TNI エンティティレコードを自動的に生成します。
- Telecom Discrepancy Identification & Reconciliation の一部としての不整合の検出と修復をサポートします。
- ローコードツールを使用して、統合をエンドツーエンドで構成、テスト、管理します。

アーキテクチャの概要

電気通信サービスグラフコネクタは、モジュール式でスケーラブルなアーキテクチャに依存しています。

サービスグラフコネクタの主要コンポーネント

コンポーネント	ロール
サービスグラフコネクタ	EMS/NMS システム (Nokia Altiplano や Nokia NSP など) からデータを抽出してステージングするための統合ロジックを定義します。

サービスグラフコネクタの主要コンポーネント (続く)

コンポーネント	ルール
MID サーバー	ServiceNow インスタンスと外部ネットワークシステム間の安全なブリッジとして機能します。 詳細については、「 統合ハブ ETL 」を参照してください。
統合ハブ ETL (3.2)	ETL 変換マップを作成、テスト、管理するためのガイド付き UI を提供します。 詳細については、「 MID サーバー 」を参照してください。
強力な変換エンジン (RTE)	定義された ETL ロジックを使用して、ステージングされたソースデータを CMDB 準拠のレコードに変換します。 詳細については、「 強力なインポートセットトランスフォーマーを作成する 」を参照してください。
識別および調整エンジン (IRE)	CI を識別して調整することで、データの一貫性を確保し、重複を防止します。 詳細については、「 CMDB 識別および調整 (IRE) 」を参照してください。
CMDB/TNI	構造化された正確な電気通信インフラストラクチャデータを保存して、可視化と下流プロセスを実現します。

サポートされているサービスグラフコネクタ

- Nokia Altiplano SGC (sn_sgc_altiplano_connector):REST API を介して Nokia Altiplano Access Network SDN コントローラーと統合します。

i 注: コネクタは、IT およびクラウドサービスグラフコネクタ (サーバー、監視ツール、IoT など) と共存できます。

主な特長

- 価値実現までの時間を短縮 – 最小限の構成で済む、事前定義されたサポート対象コネクタを使用します。
- マルチベンダーサポート – アクセス、コア、およびトランスポートネットワークにわたるさまざまな管理プラットフォームと統合します。
- モデルに合わせた可視化 – 電気通信固有の階層と関係が CMDB で正確にモデル化されるように確保します。

- 不一致検出準備完了 – CMDB コンプライアンスのために、ネットワークデータを電気通信不一致識別および調整に直接フィードします。
- スケーラブルな統合 – パフォーマンスと拡張性を重視して構築された ServiceNow の実績ある統合フレームワークを活用します。

テレコムネットワークインベントリ (TNI) データモデル

電気通信 SGC には、通信ネットワークインベントリ (TNI) データモデルとの互換性を確保するためのロジックが含まれています。

- TNI プラグインがインストールされると、検出された各ネットワーク要素に TNI エンティティレコードが自動的に含まれます。
- tni_entity は、システム生成ペイロードマッピング (inventory_category など) を使用して、対応する cmdb_ci レコードと一緒に作成されます。
- これにより、運用システムと計画システム全体で一貫性が確保されます。

TNI がインストールされている場合、次のようなペイロードが各アイテムの IRE ペイロードに追加されます (className に基づいて inventory_category が設定されます)。

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

その結果、検出された CI は cmdb_ci テーブルと tni_entity テーブルの両方にあります。

関連トピック

[Nokia Altiplano 経由の電気通信ディスカバリー](#)

Nokia Altiplano 経由の電気通信ディスカバリー

Nokia Altiplano のサービスグラフコネクタは、Nokia Altiplano アクセスネットワーク SDN コントローラーから ServiceNow CMDB にリアルタイムのネットワークインベントリを取り込む、電気通信対応の統合を提供します。

サービスプロバイダーや通信企業向けに設計されたこのコネクタは、物理ネットワークインフラストラクチャと論理ネットワークインフラストラクチャの完全な可視化、制御、同期を可能にします。この統合では、REST API と MID サーバーを使用して、通信モデルに合わせたネットワークのビューを提供し、より正確なサービスモデリング、インベントリ管理、および運用効率を実現します。

主な特長

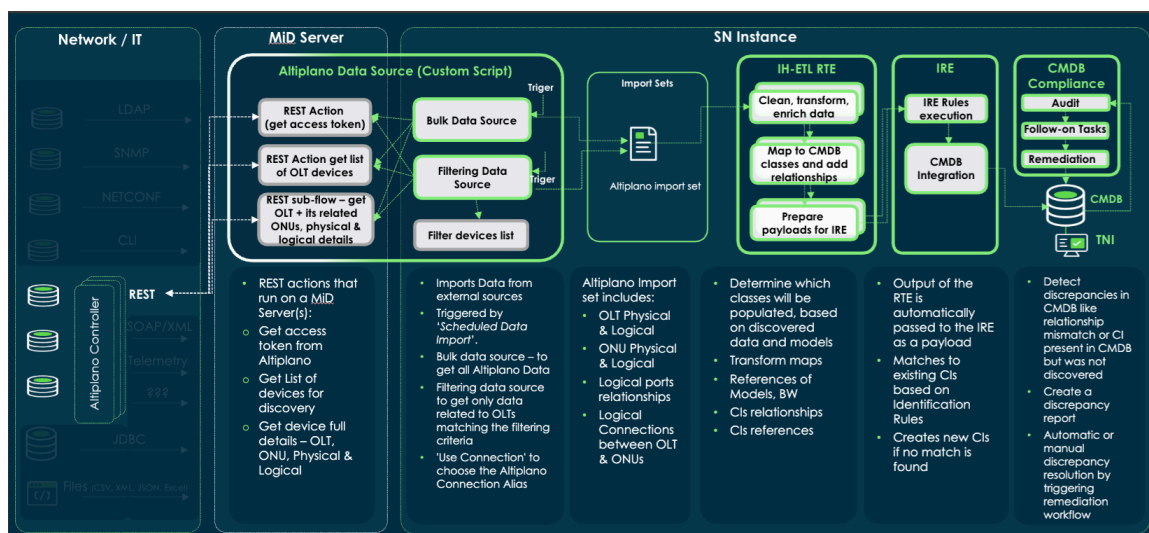
- 正確なインベントリ同期: 安全な REST API ベースの取り込みを使用して、Nokia Altiplano からの最新の物理インベントリと論理インベントリを CMDB に自動的に入力します。これには、OLT、ONU/ONT、インターフェイスカード、ポート、スロット、および論理接続が含まれます。
- 電気通信対応 CI モデリング: 専用の CI クラスと関係を使用して、ネットワークインフラストラクチャを電気通信に合わせた形式でモデル化します。コネクタは、デバイスとその依存関係の正確な表現を保証し、実際のネットワークトポロジを反映します。
- 簡素化されたガイド付きセットアップ: 接続構成、データソース管理、およびインポートスケジューリングを簡単に説明する組み込みのガイド付きセットアップにより、価値実現までの時間を短縮します。

- 柔軟なディスカバリーオプション:完全な一括ロード、ターゲットを絞ったフィルター済みディスカバリー、段階的な OLT だけのインポートなど、ニーズに基づいてディスカバリーを実行する方法とタイミングを選択します。デバイスの IP または名前でカスタムフィルターを適用します。
- マルチインスタンスのサポート:複数の Altiplano インスタンスを個別にオンボーディングすることで、簡単に拡張できます。接続エイリアスを構成し、インスタンスごとにインポートスケジュールを定義することで、運用の柔軟性を最大限に高めることができます。

i 注: サポートされている Nokia Altiplano Controller の最小バージョンは 24.6 です。サービスグラフコネクタテクノロジーの一般的な概要については、「[サービスグラフコネクタの開始](#)」を参照してください。

Nokia Altiplano SGC アーキテクチャ

次のインフォグラフィックは、Nokia Altiplano サービスグラフコネクタのアーキテクチャを理解するのに役立ちます。



自動翻訳

ユースケース

Nokia Altiplano サービスグラフコネクタの使用法の例を次に示します。

- 物理ネットワーク要素と論理ネットワーク要素の両方について、ネットワークデータを ServiceNow の CMDB に自動的に取り込み、構造化します。これにより、ライブネットワーク環境を反映した通信モデルに合わせた CMDB ビューが作成されます。
 - 物理コンポーネント: OLT、ONU/ONT、スロット、サブスロット、カード、ポート
 - 論理コンポーネント: 論理ポート、VLAN、リンクアグリゲーショングループ(LAG)、およびデバイス間の論理パス
- スケジュールされた自動調整により、ネットワークデータの完全性を維持します。
 - リアルタイムの変更を検出して応答
 - データドリフトや古いレコードを防止
 - サービス保証、受注処理、ネットワーク計画などの運用プロセスをサポート
- ServiceNow 内の論理接続をキャプチャして管理することで、物理インフラストラクチャを超えて検出します。

- 論理ポート、LAG、および PON や VLAN 接続などの論理パスを特定してモデル化します
- 親子関係およびメンバー関係を使用した論理 CI と物理コンポーネントのリンク
- エンドツーエンドの論理トポロジを可視化することで、診断、影響分析、サービスモデリングを強化
- ServiceNow に組み込まれた抽出、変換、ロード (ETL) フレームワークを活用して、統合を簡素化および促進します。
 - 事前定義された変換マップと CI クラス定義により、開発のオーバーヘッドが削減されます
 - 既存の CMDB 構造を再利用してカスタマイズを最小限に抑える
 - 新しい Altiplano インスタンスを迅速にオンボーディングし、ネットワークインフラストラクチャ全体に拡張します

主な機能とコンポーネント

主な機能とコンポーネント

機能	説明	サポートコンポーネント
電気通信認識型 CMDB モデリング	Altiplano の物理インベントリおよび論理インベントリ (OLT、ONU/ONT、ポート、スロット、カード、インターフェイス) を、電気通信固有の CI クラスと関係とともに CMDB にマッピングします。	RTE、IRE、CMDB テーブル (cmdb_ci_optical_line_terminal、cmdb_ci_optical_network_terminal など)
自動データ取り込み	安全でスケジュールされた安全なインポートを使用して、REST API を介して Nokia Altiplano からデバイスインベントリを取得します。	MID サーバー、データソース (SGC-Nokia Altiplano 一括/フィルター済みディスカバリー)
カスタムディスカバリーコントロール	OLT のみを選択するか、ONU データを含めて、IP または名前前でフィルターを適用し、インスタンスごとにジョブをスケジュールします。	スケジュール、システムプロパティ (sn_sgc_altiplano.enable_onu_discovery など) をインポートします。
マルチインスタンスのサポート	複数の Altiplano コントローラーに対してディスカバリーを個別に構成および管理します。	接続エイリアス、認証情報エイリアス
並列データ処理	大規模なデータセットに対してデータソースジョブを同時に実行することで、パフォーマンスを向上させます。	システムプロパティ: sn_sgc_altiplano.parallel_number_of_data_source、並列ロードの有効化
ガイド付き構成	接続、認証情報、およびジョブを作成するためのステップバイステップのインターフェイスにより、セットアップを簡素化します。	ガイド付きセットアップ UI (ナビゲーション: Nokia Altiplano >セットアップ>すべての>サービスグラフコネクタ)
モデル駆動型 CI 分類	検出されたデバイスをモデルに一致させ、正しい CI クラス (OLT、ONU、ONT) を割り当	モデルテーブル、システムプロパティ: sn_sgc_altiplano.onu_ci_class

主な機能とコンポーネント (続く)

機能	説明	サポートコンポーネント
	てるか、ネットワーク機器にフォールバックします。	
関係性	電気通信固有の CI 関係 (メンバー、包含、論理パスなど) を確立します。	IRE、論理接続 CI (cmdb_ci_ni_logical_path)
ダッシュボードとモニタリング	各実行のステータス、結果、およびエラーを表示します。コネクタまたは期間でフィルタリングします。	CMDB ダッシュボード共通統合
スケーラブルで再利用可能なアーキテクチャ	データソース、変換、および CI 調整を切り離し、スケーリングとカスタマイズを容易にします。	インポートセット (sn_sgc_altiplano_tsom_inventory)、変換マップ、システムプロパティ

CMDB 統合ダッシュボード

CMDB 共通統合ストアアプリは、インストールされているすべてのサービスグラフコネクタのステータス、処理結果、および処理エラーを一元的に表示するダッシュボードを提供します。すべての統合実行のメトリクスを確認することができます。特定の統合、特定の期間、または特定の統合実行に関してビューをフィルタリングすることもできます。CMDB 統合ダッシュボードでの統合の監視の詳細については、「[CMDB 共通統合](#)」を参照してください。

関連トピック

[Nokia Altiplano サービスグラフコネクタの構成](#)

[Nokia Altiplano とともにインストールされるシステムコンポーネント](#)

電気通信不一致の識別と調整

電気通信不一致識別および調整機能を使用して、CMDB または通信ネットワークインベントリ (TNI) とライブ通信ネットワークとの同期を維持します。このソリューションは、検出されたデータを継続的に監査し、インベントリレコードと比較することで、サービス品質、保証、または履行プロセスに影響を与える前に、不整合を検出、分類、および自動的に修正するのに役立ちます。

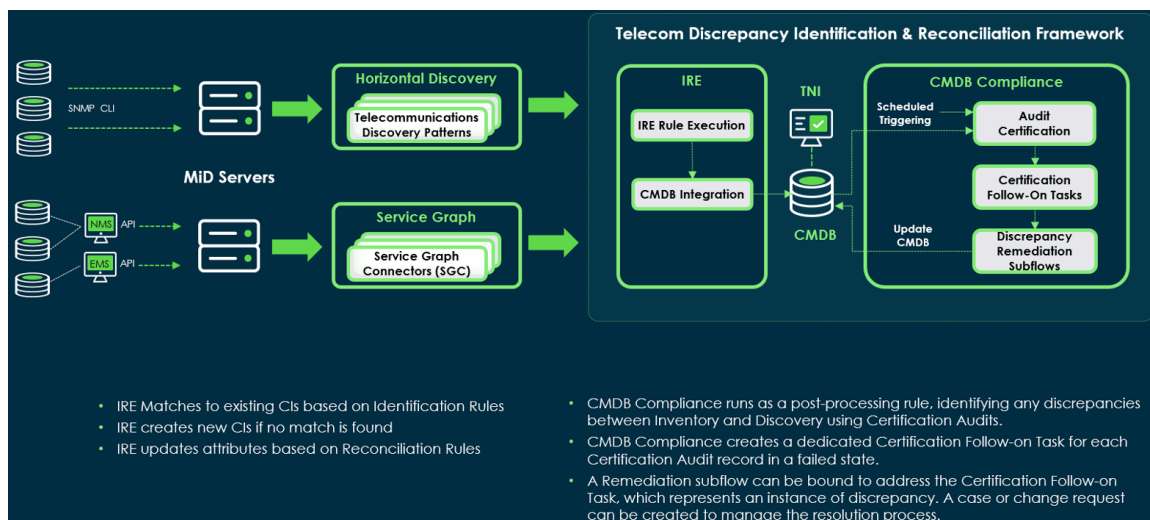
電気通信不一致識別および調整は、TSOM Visibility プラグインに含まれる電気通信固有の機能です。リアルタイムのネットワークデータ (ディスカバリーまたは外部システムからの) と CMDB または TNI に格納されているインベントリレコードの間の不一致を特定することで、ネットワークインベントリの完全性を確認するのに役立ちます。

このソリューションでは、以下を使用します。

- 例外を検出するための CMDB コンプライアンス認定監査。
- 問題をログに記録および追跡するための後続タスク。
- 不一致を解決するための自動修正サブフロー。

i 注: インベントリの正確性を維持することで、自動化が可能になり、サービスエラーが削減され、規制コンプライアンスがサポートされます。また、TMフォーラムの自律ネットワーク運用 (ANO) フレームワークの基盤コンポーネントでもあります。

電気通信不一致識別および調整アーキテクチャ



主な機能

電気通信不一致の識別および調整の主な機能

機能	説明
継続的な監査チェック	認定監査を使用して、検出されたネットワークデータとインベントリレコードを比較します。
不整合の検出	階層または属性値に関して欠落している CI、構成が間違っている、または調整が間違っている CI を特定します。
後続タスクの生成	各不一致に対してタスクを自動的に作成し、追跡と修復を可能にします。
自動修復	フローデザイナーのサブフローを使用して、古い CI、無効な関係性、ディスカバリーデータの欠落などの問題を解決します。

自動翻訳

Telecom Discrepancy Identification & Reconciliation (電気通信不一致識別および調整) の仕組み

ディスカバリープロセスが完了すると、CMDB コンプライアンスフレームワークは、スケジュール設定済みまたはオンデマンドのコンプライアンス監査を使用して、検出されたデータとインベントリレコードの間の不一致を特定します。

- **コンプライアンス監査:** 検出されたネットワークデータを CMDB インベントリレコードと比較して、不整合を検出します。
- **後続タスク:** 失敗した監査ごとに自動的に生成されます。これらのタスクは、識別された不一致を文書化して分類します。
- **修正サブフロー:** 後続タスクから起動され、構成アイテム (CI) を更新、廃止、または再調整することで不整合を解決します。

i 注: 詳細については、「[不一致の識別 - 不一致の種類](#)」を参照してください。

CMDB コンプライアンスおよび電気通信不一致の識別および調整

CMDB コンプライアンスは、アドミニストレーターが CMDB データの正確性を認定し、コンプライアンス監査中に検出された不一致を特定できるようにするツールセットです。また、失敗した監査

レコードの後続タスクを自動的に生成してアサインすることもできます。後続タスクは、適切な修正サブフローをトリガーして不整合を修正するタスクとして機能します。CMDB コンプライアンス監査は、電気通信不一致の識別および調整の基盤を形成します。

- CMDB コンプライアンスは後処理ルールとして監査を実行し、CMDB 内の例外 (不一致) を特定します。
- CMDB コンプライアンスは、失敗した状態の各監査レコードに対して後続タスクを作成します (失敗状態は、監査により CMDB の異常または不一致が見つかった結果です)。修正フローを設計してトリガーし、不一致に対処して解決するために、後続タスクごとにトリガーできます。

電気通信不一致識別および調整のロジックと修正サブフローの例は、TSOM ヴィジビリティプラグインによって自動的に設定されます。一般的な CMDB コンプライアンスツールセットの詳細については、「」を参照してください。

不一致識別シナリオ (コンプライアンス監査を使用)

TSOM ヴィジビリティでの不一致の識別は、CMDB コンプライアンス (認定監査) の使用に依存しており、モデルの関係と情報を使用して不一致を識別する特定のロジックを追加することで拡張されています。修正をサポートするために、見つかった問題ごとに次のような特定の調整タスク タイプが生成されます。

- 占有スロットの不一致
- 最新検出日が設定されていません
- 最新の検出日が設定されたしきい値内にありません
- モデル関係が定義されていません
- CI モデルが見つかりません
- 関係の数が正しくありません
- 論理インターフェイスへの参照が見つかりません

次の監査を使用して、検出された物理エンティティと論理エンティティの不一致を特定できます

- 電気通信不一致の監査
- 電気通信論理接続の不整合監査
- 電気通信ネットワークポロジの不整合監査

i 注: 一般的なコンプライアンス監査の詳細については、「」を参照してください。

自動化と UI の統合

- 監査が失敗するたびに、自動的に後続タスクが作成されます。
- タスクは、フローデザイナーを使用してビルド済みサブフローまたはカスタムサブフローをトリガーできます。
- 手動修正は、タスクフォームの [修正] UI アクションボタンを使用して開始できます。
- 修復手順は、可視化と監査のために作業メモに記録されます。

i 注: サブフローをビルドする方法の詳細については、「サブフローのビルド」を参照してください。

失敗した監査結果レコードに対して作成された後続タスクタイプ

条件に一致する CI 関係テーブル (cmdb_rel_ci) 内の各関係レコードの親 CI と子 CI について、次の不一致タイプ (監査結果) を見つけることができます。失敗した監査結果ごとに、次の後続タスクを作成できます。

1. 最新の検出日が設定されていません。CI に [最新検出日] フィールドがない場合に生成されます。
2. 親 CI と子 CI の [最新検出日] フィールド値の差が 2.5 日を超える場合に生成される、設定されたしきい値内にはない最新の検出日。たとえば、デフォルトでは、sn_tsom_core.discovered_date.diff.threshold.in.days システムプロパティで 2.5 日に設定されていますが、変更できます。
3. CI モデルが見つかりません:([モデル ID] フィールドが設定されていないか、データが無効です)。対応する CI モデルが見つからない場合に生成されます。CI モデルが見つからない場合、次の検証 (4 ~ 6) は CI モデルに依存しているため、無関係です。CI モデルが見つかった場合、監査は次の検証 (4 ~ 6) に進みます。
4. 占有されているスロットの不一致:カードが誤った数のスロットを占有している場合に生成されます。
5. モデル関係が未定義:TNI がインストールされている場合にのみ関連します。監査で [ネットワークモデルの関係] テーブルで親と子の CI モデル間の関係を見つけない場合に生成されます。
6. 関係の数が正しくありません。TNI がインストールされている場合にのみ該当します。検出された子 CI レコードの数が、ネットワークモデル関係テーブルのモデル関係カウントフィールドで対応する親 CI レコードの最大数を超過していることが監査で検出された場合に生成されます。
7. 関係の数が正しくありません:論理インターフェイスが複数の論理接続に関連付けられており、想定される 1 対 1 のマッピングに違反している場合に、論理接続の不整合の監査中に生成されます。
8. 関係の数が正しくありません。ネットワークトポロジレコードが必要な関係性の基準を満たしていない場合に、ネットワークトポロジの不整合監査中に生成されます。具体的には次のとおりです。
 - レコードには、機器 CI との「Contains:Contained By」関係が少なくとも 1 つ必要です。
 - レコードには、論理接続 CI との「メンバー:メンバー」関係も少なくとも 1 つ必要です。
9. 論理インターフェイスへの参照が見つかりません - 論理接続の不整合監査中に、論理接続にエンドポイント (ポート A またはポート z、あるいはその両方) の 1 つがない場合に生成されます。

電気通信調整

ネットワークインベントリの不一致の調整を自動化し、運用効率を向上させます。Telecom Reconciliationは、次のことに役立ちます。

- ライブネットワークインベントリと CMDB インベントリの不一致を特定して解決し、整合性を確保し、生産性を向上させます
- 不一致のタイプを詳細に説明する自動生成の不一致レポートをユーザーに提供します
- 推奨される是正処置と、矛盾を解決するための手動または自動の方法を選択するための柔軟性と制御をユーザーに提供します
- ネットワークリソースの運用ステータスをインベントリ内の同じステータスに自動的に位置合わせすることで運用を改善します CMDB

関連トピック

[不一致の識別 - 不一致の種類](#)

[電気通信不一致の識別と調整のアクティブ化](#)

電気通信不一致監査の実行

Telecom Discrepancy Identification & Reconciliation とともにインストールされるシステムコンポーネント

不一致の識別 - 不一致の種類

電気通信不一致識別および調整機能は、(TSOM ディスカバリーまたはサービスグラフコネクタを介して検出された) ネットワーク状態と、CMDB または TNI に格納されているインベントリデータとの間の不一致を識別して分類します。

不一致の識別は、CMDB コンプライアンス認定監査によって強化されます。

- CI および関係データで実行されます。
- 検出されたレコードとインベントリされたレコードを比較します。
- 不一致が検出されると、後続タスクを生成します。

不一致タイプ

監査プロセスでは、検出されたネットワークデータを CMDB/TNI 内の既存のインベントリと比較することで、不一致を特定します。不一致は、次の 2 つの主要なカテゴリに分類されます。

ネットワークにない - インベントリには存在するがネットワークに存在しないエンティティ

定義:CI はディスカバリーによって検出されましたが、CMDB/TNI に存在しないか、正しく表示されていません。たとえば、ディスカバリーは Card05 が Slot04 にインストールされていることを検出しても、CMDB は Card04 をまだリストします。さらに悪いことに、Card04 と Card05 の両方が同じスロットに表示され、カーディナリティまたはモデルの制約に違反しています。

影響度：

- データの競合を強調表示するために不一致タスクが生成されます。
- オプションの修正サブフローをトリガーして、古いレコードを廃止するかスロット割り当てを更新することで、データを調整することができます。

不一致構成アイテム (CI):インベントリとネットワークに存在するが、属性値と階層関係が異なるエンティティ

定義:CI はディスカバリーと CMDB/TNI の両方に存在しますが、関係、階層、または属性値に不一致が存在します。サブタイプは次のとおりです。

- 階層の不一致:CI 間の構造的関係 (親子の関連付けなど) に一貫性がない場合に発生します。以下に例を示します。
 - シャーシには、モデル定義で許可されている数よりも多くの子カードが含まれています。
 - カードがディスカバリーデータと一致しない CMDB のスロットに誤って関連付けられています。検証ソースは次のとおりです。
 - 関係によって Contains::Contained の `cmdb_rel_ci` レコード。
 - モデル固有の制約を適用するための `sn_ni_core_network_model_relationship` テーブル。
- 属性値の不一致:CI フィールドレベルのプロパティの不一致を伴います。一般的な問題は次のとおりです。

- ディスカバリー日が古いか、正しくありません。
- 不正確なモデル構成。
- モデルルールに違反する無効なスロット割り当て。

影響として、影響を受けるレコードは監査レポートで失敗としてマークされます。後続タスクは、コンテキスト固有の修正サブフローを呼び出して、インベントリデータを実際のネットワークステータスに再調整することができます。

i 注: 詳細については、「[CMDB 360 での属性値の不一致の構成](#)」を参照してください。

関連トピック

[電気通信不一致の識別と調整のアクティブ化](#)

[電気通信不一致監査の実行](#)

通信事業サービスオペレーション管理の構成

外部ネットワーク監視およびディスカバリーシステムと統合することで、リアルタイムのイベントの取り込み、相関、および自動修復を有効にするように 通信事業サービスオペレーション管理 (TSOM) を構成します。

TSOM をセットアップして、アラームの取り込み、CMDB の入力、不一致検出、サービス影響度のヴィジビリティなど、エンドツーエンドの通信サービスオペレーションを有効にします。この構成には、電気通信 API 通知のアクティブ化、ディスカバリーとサービスグラフコネクタの設定、可視化の有効化、監査および調整フレームワークの構成が含まれます。

構成の概要

通信サービスオペレーション管理 (TSOM) では、エンドツーエンドの通信オペレーションをサポートするために、アラームの取り込み、ディスカバリー、データ正規化、および調整全体で複数のコンポーネントを構成する必要があります。通常、構成フローには次のものが含まれます。

1. TSOM Core をシステムに組み込むには、次のプラグインをアクティブ化します
 - パターンデザイナー (com.snc.pattern.designer)
 - ServiceNow 統合ハブ開始済みパックインストーラー (com.glide.hub.integrations)
 - ディスカバリー (com.snc.discovery)
 - CMDB CI クラスモデル (sn_cmdb_ci_class) 1.69.0
 - ヴィジビリティコンテンツ (sn_pattern_design) 6.23.0
 - CMDB 共通統合 (sn_cmdb_int_util) 2.19.0
2. アラームの取り込みを有効にする: Telecommunications API 通知を有効にし、外部システムからアラームを受信するようにトピックサブスクリプションを設定します。詳細については、「[電気通信 API 通知の構成](#)」を参照してください。
3. 可視化の設定: Telecom Visibility を設定して、サービスからインフラストラクチャへのマッピングを表示し、ネットワーク健全性を監視します。詳細については、「[Telecom Visibility を設定](#)」を参照してください。

- a. 電気通信認識型 CMDB に入力します。
 - 水平ディスカバリーをインストールして構成します。詳細については、「[水平ディスカバリーのインストールとディスカバリーパターンの設定](#)」を参照してください。
 - コネクタで Telecom Discovery Builder (TDB) ETL フローを設定します。詳細については、「[コネクタでの Telecom Discovery Builder フレームワーク ETL の構成](#)」を参照してください。
 - サービスグラフコネクタ (Nokia Altiplano など) を使用して、トポロジと構成データをインポートします。詳細については、「[Nokia Altiplano サービスグラフコネクタの構成](#)」を参照してください。
- b. 不整合検出を有効化:
 - 電気通信不一致の識別および調整をアクティブ化します。詳細については、「[電気通信不一致の識別と調整のアクティブ化](#)」を参照してください。
 - 電気通信監査のフィルターを定義し、CMDB 360 で属性値の不一致チェックを構成します。

各ステップはモジュール式で、環境と利用可能な統合に基づいて構成できます。

電気通信 API 通知の構成

ServiceNow インスタンスで電気通信 API 通知を構成します。

Telecommunications API 通知ワークフローのモデル化

次の手順は、ServiceNow インスタンスで電気通信 API 通知を構成するのに役立ちます。

1. **トピックを作成:**外部メッセージの詳細を手動で入力するか、利用可能なトピックを外部システムから自動的に収集することで、トピックを作成できます。
2. **トピックのサブスクリプションを作成:**顧客設定に基づいて、外部システムからの受信通知に利用可能なトピックをサブスクライブします。さらに、コールバック URL を生成してサブスクリプションを登録します。
3. **電気通信アラーム管理オープン API 接続のエンドポイントをアクティブ化します:**外部システムから応答を受信するには、フローデザイナーで電気通信アラーム管理オープン API 接続の登録済みエンドポイントをアクティブ化します。
4. **通知を受信するための外部システムへのコールバック URL を提供します。**顧客はコールバック URL を再使用することもできます。TMF 688 からの要求がコールバック URL にヒットすると、デフォルトアラームイベント通知トリガー フローが開始されてイベントが作成されます。

イベントを作成、更新、削除するために外部トリガー定義によってトリガーされるイベント通知管理オープン API 要求を処理する関数の詳細については、「[イベント通知管理オープン API](#)」と「[TMFTopicEventAPIUtilOOB: スコープ指定](#)」を参照してください。

このワークフローは、イベント管理 アプリケーションでイベントを作成します。イベント管理の使用の詳細については、「[イベント管理](#)」を参照してください。

関連トピック

[通信事業 API 通知とともにインストールされるシステムコンポーネント](#)

[電気通信 API 通知を介した外部イベント管理](#)

トピックを作成

トピックを作成し、外部システムからの受信通知をトピックに公開します。トピックを作成することで、登録者は登録するトピックを選択できます。

始める前に

Telecommunications Alarm Management Open API (sn_ind_tmf642) アプリケーションが ServiceNow AI Platform とともにインストールされていることを確認します。

必要なロール: admin、sn_api_notif_mgmt.topic_creator

このタスクについて

トピックを作成するには、外部メッセージの詳細を手動で入力するか、利用可能なトピックを外部システムから自動的に収集します。トピックを作成すると、トピック [sn_api_notif_mgmt_topic] テーブルにレコードが作成されます。

手順

1. **すべて > Telecom API 通知 > トピック.**
2. **[New (新規)]** を選択します。
外部システムと統合している場合は、[トピックを取得] を選択して、利用可能なトピックを自動的に取得できます。このアクションにより、イベントアラーム通知 API サブフローがトリガーされます。トピック内のレコードのクエリと操作を可能にする関数の詳細については、「[TopicUtilOOB: スコープ指定](#)」を参照してください。
3. フォームで、フィールドに入力します。

トピックフォーム

フィールド	説明
トピック ID	一意のトピック ID。
トピック名	トピックの名前。
タイプ	トピックのタイプ。次のいずれかを選択します。 <ul style="list-style-type: none"> ○ 入口:受信通知のオプション。 ○ エグレス:送信通知のオプション。
ヘッダークエリ	エンコードされたヘッダークエリパラメーター。 TMF 688 標準に従うクエリパラメーターの詳細については TM フォーラム を参照してください。
コンテンツクエリ	エンコードされたコンテンツのクエリパラメーター。 TMF 688 標準に従うクエリパラメーターの詳細については TMF 688 を参照してください。
説明	トピックに関する簡単な説明。

4. **[Submit (送信)]** を選択します。

結果

トピックが作成されます。

次のタスク

トピックサブスクリプションは、顧客の要件に従って作成できます。詳細については、「[トピックのサブスクリプションを作成](#)」を参照してください。

トピックのサブスクリプションを作成

外部システムからの受信通知にตอบสนองする ServiceNow AI Platform のトピックに登録します。トピックをサブスクライブすると、登録者は、サブスクライブしたトピックに基づいて通知を受信します。

始める前に

- Telecommunications Alarm Management Open API (sn_ind_tmf642) アプリケーションが ServiceNow AI Platform とともにインストールされていることを確認します。
- 受信通知のトピックを作成します。

必要なロール:admin、sn_api_notif_mgmt.subscription_creator

このタスクについて

顧客の設定に基づいて、外部システムからの受信通知に利用可能なトピックをサブスクライブします。顧客と共有するコールバック URL を生成します。外部システムからの要求がコールバック URL にヒットすると、イベント管理 アプリケーションでイベントの作成が開始されます。

さらに、トピックサブスクリプションに登録して、受信通知の受信を開始します。トピックサブスクリプションを作成すると、トピックサブスクリプション [sn_api_notif_mgmt_subscription] テーブルにレコードが作成されます。トピックサブスクリプションでレコードをクエリおよび操作する方法の詳細については、「[TopicSubscriptionUtilOOB: スコープ指定](#)」を参照してください。

手順

1. [すべて](#) > **Telecom API 通知** > サブスクリプション.
2. **[New (新規)]** を選択します。
3. フォームで、フィールドに入力します。

トピックのサブスクリプションフォーム

フィールド	説明
トピック	登録するトピック。
CallbackURL	受信通知をキャプチャするために外部システムと共有しているコールバック URL。この URL は、[CallbackURL の生成] を選択すると自動的に生成されます。
フィルタークエリ	トピックからのエンコードされたコンテンツのクエリパラメーター。フィルタークエリを変更することもできます。 TMF 688 標準に準拠するクエリパラメーターの詳細については、 TM フォーラム を参照してください。
登録ステータス	外部システムとのトピック登録のステータス。デフォルトでは [未登録] です。プロセスが成功すると、フィールドの値が [登録済み] に変わります。それ以外の場合は [エラー] です。

フィールド	説明
登録メッセージ	外部システムからの登録ステータスメッセージ。
サブスクリプション ID	外部システムからの一意のサブスクリプション ID。

4. [**CallbackURL** の生成] を選択して、コールバック URL を取得します。

5. [登録] を選択してサブスクリプションを登録します。

結果

コールバック URL のトリガー定義が作成され、トピックが外部システムに登録されます。

次のタスク

フローデザイナーでは、通信アラーム管理オープン API 接続のエンドポイントをアクティブ化します。詳細については、「[電気通信アラーム管理オープン API 接続のエンドポイントをアクティブ化します](#)」を参照してください。

電気通信アラーム管理オープン **API** 接続のエンドポイントをアクティブ化します

通信アラーム管理オープン API 接続のエンドポイントをアクティブ化します。エンドポイントをアクティブ化すると、登録したトピックの受信通知が外部システムから受信されます。

始める前に

- トピックを作成してサブスクライブし、受信通知を受信します。
- コールバック URL を生成し、トピックサブスクリプションを登録します。

必要なロール：admin

このタスクについて

外部システムから応答を受信するには、フローデザイナー で電気通信アラーム管理オープン API 接続の登録済みエンドポイントをアクティブ化します。

手順

1. 移動先 [すべて](#) > [プロセス自動化](#) > [フローデザイナー](#).
2. [接続] タブで、[通信アラーム管理オープン **API**] を選択します。
3. アクティブ化するエンドポイントレコードを開きます。
4. [Activate (アクティブ化)] を選択します。

Telecom Visibility を設定

Telecom Visibility は、Telecom Discovery と Telecom Dysrepancy Identification & Reconciliation の両方をサポートする基本機能を提供します。これには、共有ロジック、拡張 CI クラスモデル、および電気通信ネットワーク要素に合わせて調整された識別および調整エンジン (IRE) の更新が含まれています。

始める前に

必要なロール：admin

次のことを確認します。

- ServiceNow インスタンスに TSOM のライセンスが付与されている。
- MID サーバーが動作しており、ディスカバリーのために検証されています。
- 電気通信 CI に適用されるカスタム IRE ルールを確認して、アップグレード中に上書きされないようにします。

このタスクについて

Telecom Visibility を設定するには、必要なプラグインをインストールし、CMDB CI クラスモデルをバージョン 1.69.0 に更新する必要があります。これにより、電気通信固有の IRE 識別ルールが導入されます。これにより、電気通信ドメイン全体で正確な CI 識別と調整を確認できます。

- i** 注: 影響を受ける電気通信 CI のいずれかに対して IRE 識別ルールをカスタマイズしている場合、バージョン 1.69.0 にアップグレードすると、それらのルールが上書きされたり、影響を受ける可能性があります。

Telecom Visibility サブスクリプションには、次の機能が含まれています。

プラグイン名	説明	ストアアプリ
サービスグラフコネクタ sn_sgc_altiplano_connector	Nokia Altiplano のサービスグラフコネクタ	はい
電気通信ディスカバリーパターン sn_tsom_patterns	電気通信ディスカバリーパターン	はい
Telecom Visibility プラグイン (コアロジック) sn_tsom_core	電気通信の可視化と不一致の調整のコアロジック	いいえ

- i** 注: これらのプラグインは、CMDB CI クラスモデルアプリケーションのインストールまたはバージョン 1.69.0 への更新を自動的にトリガーします。

手順

1. 移動先 **すべて > プラグイン**.
2. 以下を検索してインストールします。
 - a. Nokia Altiplano のサービスグラフコネクタ (sn_sgc_altiplano_connector)
 - b. 電気通信ディスカバリーパターン (sn_tsom_patterns)
 - c. Telecom Visibility (プラグイン) (sn_tsom_core)
3. オプション: CMDB CI クラスモデルを更新します。
 - a. 移動先 **すべて > 利用可能なアプリケーション ServiceNow Store** のリスト。
 - b. CMDB CI クラスモデル (sn_cmdb_ci_class) を検索します。
 - c. バージョン 1.69.0 をインストールするか、アップグレードします。

- i** 注: 上記の TSOM プラグインのいずれかをインストールすると、CMDB CI クラスモデルアプリのバージョン 1.69.0 が自動的に更新またはインストールされます。インスタンスに TSOM プラグインが含まれていない場合、または Yokohama より前のリリース (Washington DC や Xanadu など) を使用している場合は、ストアアプリを手動でインストールまたはアップグレードできます。

4. 移動先 **CMDB > 識別ルール**.

a. 以下に関連するレビュールール:

- `cmdb_ci_interface_card`
- `cmdb_ci_slot`
- `cmdb_ci_subslot`
- `cmdb_ci_network_adapter`

b. カスタムロジックが引き続き機能することを検証します。**c.** 必要に応じてカスタムルールを更新または結合します。**結果**

以下がアクティブとしてリストされていることを確認します。

- 電気通信の可視化 (`sn_tsom_core`)
- 電気通信ディスカバリーパターン (`sn_tsom_patterns`)
- Nokia Altiplano のサービスグラフコネクタ (`sn_sgc_altiplano_connector`)
- CMDB CI クラスモデルバージョン 1.69.0

関連トピック[電気通信のヴィジビリティ](#)**水平ディスカバリーのインストールとディスカバリーパターンの設定**

水平ディスカバリーパターンのインストール 依存関係と要件を理解します。

始める前に

必要なロール: `admin`

TSOM のサブスクリプションがあることを確認します。

このタスクについて

[ServiceNow Store](#) Web サイトにアクセスして利用可能なすべてのアプリを表示し、ストアにリクエストを送信する方法について確認してください。リリースされたすべてのアプリのリリースメモ情報については、「[ServiceNow Store バージョンの履歴リリースノート](#)」を参照してください。

依存関係と要件:


- テレコムコア (`sn_tsom_core`)
- ディスカバリーによって自動的にインストールされるディスカバリーコアプラグイン (`com.snc.discovery.core`)。
- ITOM Discovery ライセンスプラグイン (`com.snc.itom.discovery.license`)。このプラグインを有効にする必要があります。
- ITOM Licensing プラグイン (`com.snc.itom.license`)。詳細については、「[ディスカバリーの要求](#)」を参照してください。

パターン実行ロジック:デフォルトでは、TSOM パターンが実行されると、TSOM 固有のパターンと対応する ITOM パターンの両方が実行されます (Telco Router がルーターを実行するなど)。これにより、必要に応じて共有 ITOM ライブラリが再利用されます。この動作を上書きするには:

- システムプロパティ `sn_tsom_patterns.itom_pattern_enabled` を使用します。
- このプロパティを `false` に設定すると、TSOM 固有のパターンのみが実行されるようになります。


手順

1. 水平ディスカバリーアプリケーションをインストールします。
電気通信ディスカバリーパターンを実行するための基礎となるため、「[ディスカバリーのセットアップ](#)」を参照してください。
2. 電気通信ディスカバリーパターンを取得してインストールします。
 - a. ServiceNow ストアから電気通信ディスカバリーパターン (`sn_tsom_patterns`) をインストールします。
3. MID サーバーと同期の設定パターン:
 - a. インストールされているパターンを適切な MID サーバーと同期して、使用する準備ができていることを確認します。
 - i. 移動先 [ディスカバリー](#) > **MID** サーバー.
 - ii. **[MID にパターン同期]** を選択します。

 注: このアクションは、TSOM と ITOM の両方のパターンを同期します。

MID サーバーの構成方法の詳細については、「[MID サーバーの構成](#)」を参照してください。
4. TSOM システムプロパティを構成します。
 - a. システムプロパティ `sn_tsom_patterns.itom_pattern_enabled` を設定して、TSOM パターンのみを使用するか、ITOM と TSOM パターンの組み合わせを使用するかのロジックを定義します。
 - i. 移動先 [すべて](#) > システムプロパティ > すべてのプロパティ.
 - ii. **`sn_tsom_patterns.itom_pattern_enabled`** を選択します。
 - iii. 値が **true** (デフォルト) に設定されていることを確認します。

TSOM で TSOM パターンのみを実行し、ITOM パターンを除外する場合は、値を **false** に設定します。

 注: デフォルト設定は、TSOM パターンと ITOM パターンの両方を使用するように構成されています。

5. 特定の MID サーバー上のさまざまな ITOM パターンを TSOM パターンに置き換えることができます。

例:特定の MID サーバーに対して **`mid.telecom.discovery.patterns.enabled`** が `true` に設定されている場合、特定の MID サーバーについて、ネットワークルーターパターンが電気通信ルーターパターンに置き換わります。

- a. フィルターナビゲーターに移動し、「ecc_agent_config.list」と入力します。
- b. **mid.telecom.discovery.patterns.enabled** を選択します (各 MID サーバーにこのパラメーターがあります)。
- c. 値が **true** に設定されていることを確認します。

T SOM パターンの実行に使用する各 MID サーバーについて、この構成を繰り返します。

関連トピック

[ディスカバリーパターンを使用した直接ディスカバリー](#)

コネクタでの **Telecom Discovery Builder** フレームワーク **ETL** の構成

事前に構築された Telecom Discovery Builder フレームワーク ETL をコネクタスコープに複製し、適切なデータソースを割り当てて、新しいサービスグラフコネクタの一部として展開することで活用します。

通信サービスオペレーション管理 (T SOM) Core で提供される Telecom Discovery Builder フレームワーク ETL は、通信コネクタ全体で CMDB へのデータの取り込みを簡素化するように設計された、すぐに使用できるフレームワークです。アドミニストレーターは、直接構成するのではなく、ETL をコネクタのアプリケーションスコープに複製し、コネクタのディスカバリーペイロードに合わせてデータソースを更新できます。

コネクタで **Telecom Discovery Builder** フレームワーク **ETL** を使用する手順

1. 一時的なデータソースを作成: T SOM Core の汎用スキーママルチソースデータソースに基づいて、コネクタのアプリケーションスコープにデータソースを作成します。これにより、スキーマ整理されたプレースホルダーデータを使用して、重複した ETL を実行してテストできます。詳細については、「[電気通信コアデータソースに類似したデータソースを作成する](#)」を参照してください。
2. Telecom Discovery Builder フレームワーク ETL の複製: ETL Studio にアクセスし、Telco 汎用スキーマ ETL を見つけて、コネクタのスコープに複製します。複製中: 新しい名前を指定します。一時データソースを割り当てます。ディスカバリーソースとして importSet を使用します。詳細については、「[Telecom Discovery Builder フレームワーク ETL をコネクタスコープに複製します](#)」を参照してください。
3. 複製後、新しい ETL を開き、一時データソースをコネクタの実際のディスカバリーデータソースに置き換えます。詳細については、「[コネクタのデータソースを更新します](#)」を参照してください。
4. サービスグラフコネクタをテストまたは展開します。詳細については、「[既存の ETL を使用した新しいサービスグラフコネクタの展開](#)」を参照してください。

電気通信コアデータソースに類似したデータソースを作成する

コネクタのアプリケーションスコープにスキーマ準拠のデータソースを設定して電気通信データをシミュレートし、ライブデバイスデータと統合する前に電気通信ディスカバリービルダーフレームワーク ETL のテストと検証が正常に行われるようにします。

始める前に

必要なロール: admin

次のことを確認します。

- TSOM コアアプリケーションとそのデータソースへのアクセス。
- コネクタのアプリケーションスコープに切り替えます。
- Telco 汎用スキーマに準拠したサンプルペイロードを用意します (オプションですが、テストに推奨します)。

このタスクについて

Telecom Discovery Builder フレームワーク ETL をサービスグラフコネクタ (SGC) アプリケーションスコープに複製する場合は、まず TSOM Core データソースの正確な構造を複製するデータソースを作成する必要があります。この重複データソースは、ETL をライブ通信データに接続する前に ETL をテストおよび検証するために必要なスキーマと構造を提供します。

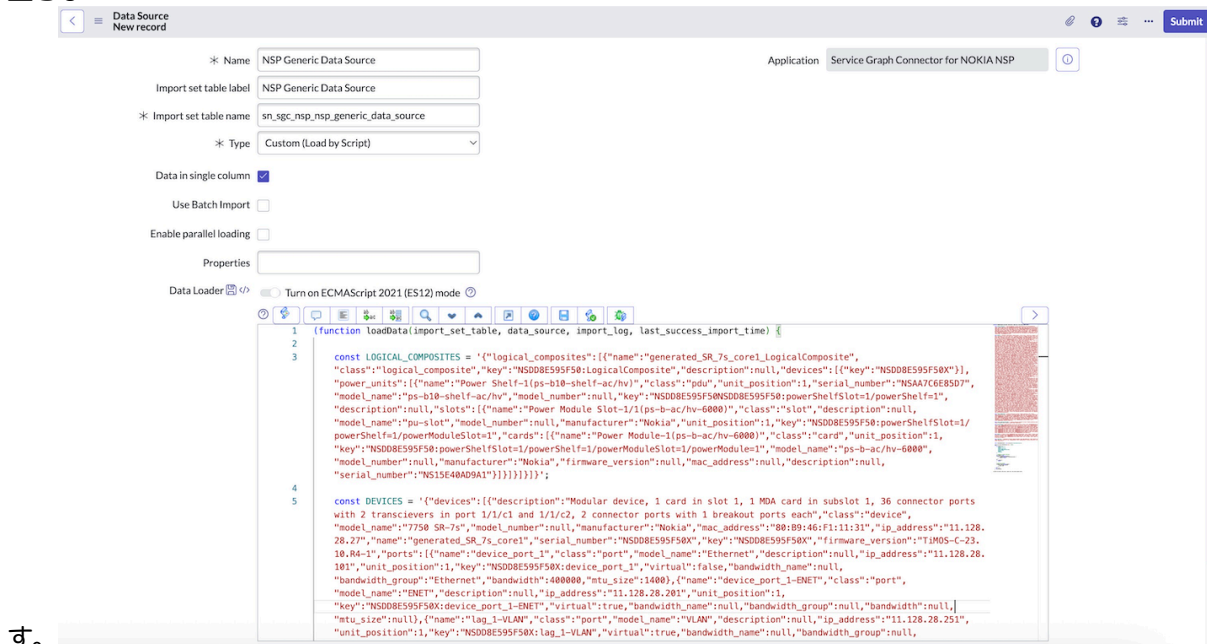
Telecom Discovery Builder フレームワークは、Telecom Core で定義されている Telco 汎用スキーマに準拠するデータを想定しています。元の TSOM データ・ソースを複製すると、次のことが保証されます。

- ETL 複製中のスキーマアライメント。
- テスト用の有効なインポートセット構造。
- カスタマイズとアップグレードの安全性のためのコアとコネクタのスコープの分離。

TSOM Core データ・ソースに類似したデータ・ソースを作成する場合:

- Telecom Discovery Builder フレームワーク ETL を複製する前に。
- シミュレートされた通信データまたはプレースホルダー通信データを使用してテストロードを実行する場合。
- 取り込み構成のためにコネクタのアプリケーションスコープを準備するとき。

次のスクリーンショットは、データソースの作成時にフィールド値を入力する方法を理解するのに役立ちます。



手順

1. 移動先 [すべて](#) > [システムインポートセット](#) > [アドミニストレーション](#) > [データソース](#).
2. 汎用スキーママルチソース v2 または TSOM コアに付属する同様のベースラインソースを検索します。
3. TSOM Core データソースレコードを開き、[データローダー] 項目からスクリプトをコピーします。
4. [新規] をクリックしてデータソースを作成します。
5. フォームでフィールドに入力します。
詳細については、「[カスタム \(スクリプトでロード\) タイプのデータソースを作成する](#)」を参照してください。
6. [タイプ] フィールドで、[カスタム (スクリプトでロード)] を選択します。
7. [単一列のデータ] フィールドを選択します。
8. [データローダー] 項目にコピーしたスクリプトを貼り付けます。
9. [Submit (送信)] を選択します。
データソースが作成されます。
10. オプション: データソースをテストロードするには:
 - a. [20 件のレコードのテストロード] (または同様のレコード) をクリックして、インポートセットを生成します。
 - b. レコードがエラーなしで作成されていることを確認します。
 - c. 論理コンポジット、ネットワーク機器、ポートなどのクラスがステージングテーブルに表示されていることを確認します。

次のタスク

データソースを作成してテストした後:

- Telco 汎用 ETL を複製するときにインポートソースとして使用します。
- シミュレーションと検証が完了したら、実際のコネクタ固有のデータソースに置き換えます。

関連トピック

[すべてのサービスグラフコネクタをサポートする標準化された JSON 共通データセット](#)
[Telecom Discovery Builder フレームワーク ETL をコネクタスコープに複製します](#)

Telecom Discovery Builder フレームワーク ETL をコネクタスコープに複製します

Telecom Discovery Builder フレームワーク ETL を使用すると、サービスグラフコネクタ (SGC) チームは、ETL ロジックをゼロからビルドすることなく、標準化されたスキーマ準拠のデータ取り込みパイプラインを迅速に採用できます。

始める前に

必要なロール: admin

次のことを確認します。

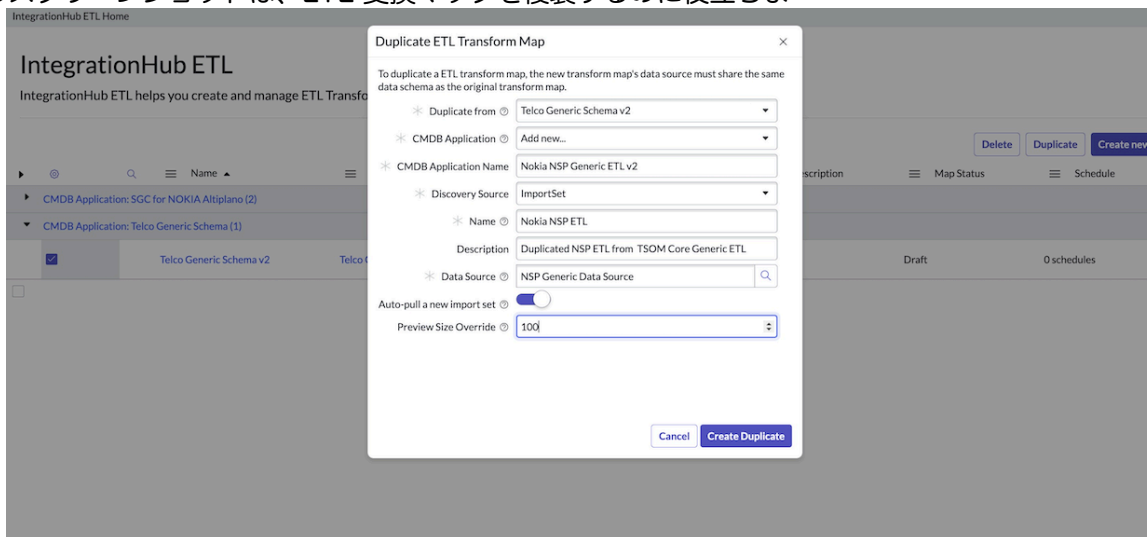
- Telecom Core プラグインが有効になり、Telco Generic ETL v2 (自動的にインストールされる) が利用可能になります。
- 統合ハブ ETL Studio へのアドミンアクセス権がある。
- コネクタのアプリケーションスコープに一時データソースが作成されました。一時データソースを作成した後、汎用 ETL をコネクタのアプリケーションスコープに複製して、特定のユースケースに合わせてカスタマイズおよび拡張できます。

このタスクについて

Telco 汎用 ETL を複製すると、次のことができます。

- 標準化された ETL マッピングを複数のコネクタで再利用します。
- 通信サービスオペレーション管理 (TSOM) コアによって提供される元のベースラインを変更することなく、ETL の動作をカスタマイズします。
- 一貫性と TNI コンプライアンスのために、Telco 汎用スキーマに合わせて調整します。
- テスト済みで実績のある ETL フレームワークを使用することで、時間を節約し、エラーを減らします。

次のスクリーンショットは、ETL 変換マップを複製するのに役立ちま



す。

手順

1. 移動先 **すべて > 統合ハブ ETL**.
2. [**統合ハブ ETL**] ページで、コネクタのアプリケーションスコープを選択します。
 - ❗ **注:** 例:Nokia **NSP** のサービスグラフコネクタ。
3. CMDB アプリケーション **Telco 汎用スキーマ**を展開し、 **Telco 汎用スキーマ** レコードを選択します。
4. [複製] をクリックします。
[重複 ETL 変換マップ] ページが表示されます。
5. 重複した ETL 変換マップの構成
 - a. [複製元] フィールドで、[Telco 汎用スキーマ] が選択されていることを確認します。
 - b. 既存の CMDB アプリケーションを選択するか、[新しく追加...] をクリックして新しい CMDB アプリケーションを追加します。
 - c. 新しい CMDB アプリケーションの場合は、CMDB アプリケーション名を入力します (例:Nokia NSP Generic ETL v2)
 - d. [ディスカバリーソース] フィールドで、インポートセットを選択します。
 - e. [名前] フィールドに、重複する ETL の名前を入力します。

- f. [説明] フィールドに、重複する ETL の説明を入力します。
 - g. 重複する ETL 変換マップに使用するデータソースを選択します。
 - ① 注: これは、Telco 汎用スキーマ ETL に添付されている既存のデフォルトのデータソースとは異なる必要があります。詳細については、「[電気通信コアデータソースに類似したデータソースを作成する](#)」を参照してください。
 - h. [新しいインポートセットを自動プル] オプションを有効にして、データを新しいインポートセットに自動的にプルします。
 - i. [プレビューサイズの上書き] フィールドで、テストと検証のためのカスタムプレビューサイズを設定します。
6. [複製を作成] を選択します。
Telco 汎用 ETL の重複 ETL が作成されます。
7. オプション: 複製された ETL レコードを開き、マッピングと設定を確認します。

次のタスク

1. 重複した ETL をテストします (オプション、ただし推奨):
 - 一時データソースを使用してテストロードまたはシミュレーションを実行します。
 - インポートセットが正常に処理されていること、Telco 汎用スキーマに従って CI が作成されていること、および関係が適切に確立されていることを確認します。
2. 複製とテストが成功した後:
 - 実際の本番データソース (実際のデバイスデータの場合) を指すように ETL のデータソース構成を更新します。
 - コネクタ統合をテスト環境または本番環境に展開します。
 - インポート実行を監視して、インベントリデータが CMDB に正しく取り込まれていることを検証します。

関連トピック

[コネクタのデータソースを更新します](#)

例 - Telco 汎用 ETL スキーマの複製

この例では、Telco 汎用スキーマ ETL を複製して、電気通信統合用にカスタマイズされたサービスグラフコネクタ (SGC) ETL を設定する方法について説明します。標準化された Telco 汎用スキーマに基づいてコネクタのアプリケーションスコープにベースライン ETL を作成し、スキーマの整合、一貫性、および迅速な展開を確保する場合は、この手順を使用します。

シナリオ

新しいサービスグラフコネクタを展開しており、Telco 汎用スキーマ ETL をコネクタのアプリケーションスコープに複製する必要があります。

一時的なデータソースを作成し、ETL を複製して、コネクタのデバイスペイロードで動作するように構成します。

Telco 汎用スキーマ ETL を複製する手順

1. ETL を複製するコネクタのアプリケーションスコープ (Nokia NSP Connector など) に切り替えます。
2. データソースを作成します。
 - [System Import Sets] > [Administration] > [Data Sources] に移動します。
 - TSOM Core アプリケーションによって提供される汎用スキーママルチソースデータソースを見つけます。
 - このデータソースをコネクタのアプリケーションスコープにコピーします。
 - サンプルインポートセットを作成するには、[20 件のレコードのテストロード] を実行して、コピーしたデータソースをテストします。
3. [重複 ETL 変換マップ] ダイアログを開きます。ETL Studio で、[重複 ETL] を選択して複製プロセスを開始します。
4. [複製元(Duplicate from)] リストで、[Telco 汎用スキーマ(Telco Generic Schema)] を選択します。
5. [新しく追加] を選択し、複製した ETL の名前を入力します。
6. 複製した ETL のディスカバリーソースとして importSet を選択します。
7. 複製された変換マップの新しい名前を入力します。
8. 汎用スキーママルチソースからコピーした新しく作成された一時データソースを指定します。
9. 必要に応じて、[新しいインポートセットを自動プル] を有効にして、複製後に新しいレコードを自動的にロードします。
10. [複製を作成] をクリックして、複製を完了します。
11. 基本的な詳細を更新:
 - 複製後、新しく作成された ETL を開きます。
 - ステップ 1: [基本的な詳細] を指定するで、一時データソースをコネクタの本番データソースに置き換えます。
12. ETL 構成の保存: [保存] をクリックして変更を確定します。
13. データソースの実行:[スケジュールをインポート] から、コネクタのデータソースを実行して実際のデバイスデータを取り込みます。

結果:Telco 汎用スキーマ設定に基づいて新しい ETL が作成され、指定されたコネクタデータソースからのペイロードが処理されます。インポートされたデータは構成アイテム (CI) に変換され、想定される関係と構造で ServiceNow CMDB に挿入されます。

コネクタのデータソースを更新します

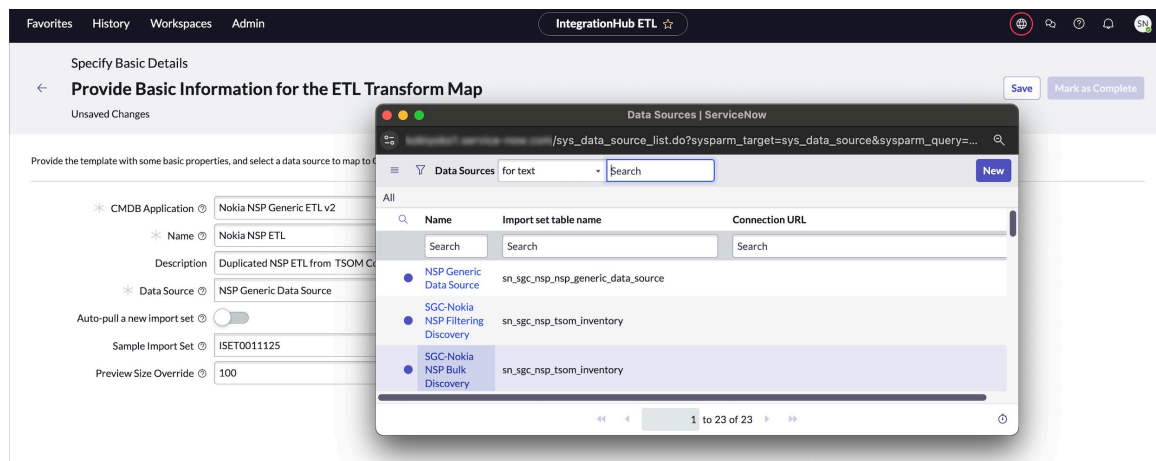
重複した ETL を特定のコネクタの有効なデータソースにリンクします。

始める前に

必要なロール : admin

このタスクについて

次のスクリーンショットは、デフォルトのソースをコネクタのデータソースに置き換えるのに役立ちます。



す。

手順

1. 移動先 **すべて > 統合ハブ ETL**.
2. [基本的な詳細を指定] で、デフォルトのデータソースをコネクタのデータソースに置き換えます。
3. [**Save** (保存)] を選択します。
4. [スケジュールのインポート] を開き、データソースを実行します。
選択した設定に基づいて新しい変換マップが作成され、ペイロードから生成された CI が作成されます。

関連トピック

[既存の ETL を使用した新しいサービスグラフコネクタの展開](#)

既存の **ETL** を使用した新しいサービスグラフコネクタの展開

新しいサービスグラフコネクタ (SGC) を既存の ETL に関連付けます。

始める前に

必要なロール：admin

必要なサービスグラフコネクタプラグインがアクティブ化されていることを確認します。

手順

1. 移動先 **すべて > 統合ハブ ETL**.
2. 複製した ETL 設定を開きます。
3. サービスグラフコネクタのデータソースを選択し、[保存] をクリックします。
4. [スケジュールのインポート] を選択して ETL を実行し、新しい SGC からデータを変換してロードします。
5. 実行状況を監視します。

結果

CMDB マップまたはリストビューを使用して、CI レコードと関係が CMDB に作成されていることを確認します。

重複した ETL に対する TNI エンティティサポートの拡張

テレコムネットワークインベントリ (TNI) が有効になっている場合、複製された ETL によって作成されたすべての構成アイテム (CI) には、対応する TNI エンティティが必要です。

始める前に

必要なロール：admin

- TNI がインスタンスにインストールされ、アクティブになっていることを確認します。
- Telecom Discovery Builder フレームワーク ETL のターゲットアプリケーションスコープへの複製を完了します。

手順

1. 移動先 **すべて > 統合ハブ ETL**.
2. 複製された ETL に関連付けられている CMDB アプリケーションを選択します。
新しいページの CMDB Integration Studio アプリケーション
3. ETL 設定ページで、[スクリプトの前に実行] オプションをオンにします。
4. デフォルトのスクリプトを次のように置き換えます。

```
(function(input, runId) {
  new
  sn_tsom_core.TelcoGenericMappingHelper().checkAndUpdatePayloadForTni(input);
})(input, runId);
```

5. [Update (更新)] をクリックして変更を保存します。
重複した ETL は、検出された CI の TNI エンティティにリンクします。

すべてのサービスグラフコネクタをサポートする標準化された **JSON** 共通データセット

TSOM アーキテクチャを使用して、統一されたスキーマと再利用可能な ETL ロジックを使用して、標準化されたサービスグラフコネクタをサポートします。これにより、新しいコネクタのオンボーディング時間が短縮され、CMDB との統合が簡素化されます。

1. 共通の JSON スキーマを定義する

TNI スキーマに準拠する単一の JSON 形式に合わせて、すべてのサービスグラフコネクタの出力を標準化します。

次の点を確認してください。

1. コレクターまたはアダプターに変換ロジックを実装して、共通スキーマでデータを出力します。
2. 必ず次のことを実行します。
 - スロットオンスロットまたはカードオンカード階層は除外されます。
 - 論理インターフェイスは、`virtual=true` で明確にマークされます。
 - 機器タイプはモデルクラスのマッピングに合わせて調整されます。

- i** 注：スキーマは、TNI 変更への実行時の適応性をサポートする必要があります (利用可能な場合)。

すべての TSOM サービスグラフコネクタをサポートする JSON スキーマ共通データセットは次のとおりです。

```

{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "Telco Generic Schema",
  "version": "2.0.1",
  "oneOf": [
    {
      "type": "object",
      "properties": {
        "logical_composites": {
          "type": "array",
          "items": {
            "$ref": "#/$defs/logical_composite"
          }
        }
      },
      "required": [
        "logical_composites"
      ],
      "additionalProperties": false
    },
    {
      "type": "object",
      "properties": {
        "devices": {
          "type": "array",
          "items": {
            "$ref": "#/$defs/device"
          }
        }
      },
      "required": [
        "devices"
      ],
      "additionalProperties": false
    },
    {
      "type": "object",
      "properties": {
        "logical_connections": {
          "type": "array",
          "items": {
            "$ref": "#/$defs/logical_connection"
          }
        }
      },
      "required": [
        "logical_connections"
      ],
      "additionalProperties": false
    },
    {
      "type": "object",
      "properties": {
        "port_relations": {
          "type": "array",
          "items": {

```

```

    "$ref": "#/$defs/port_relation"
  }
},
"required": [
  "port_relations"
],
"additionalProperties": false
},
{
  "type": "object",
  "properties": {
    "logical_connection_relations": {
      "type": "array",
      "items": {
        "$ref": "#/$defs/logical_connection_relation"
      }
    }
  },
  "required": [
    "logical_connection_relations"
  ],
  "additionalProperties": false
},
{
  "type": "object",
  "properties": {
    "numbers": {
      "type": "array",
      "items": {
        "$ref": "#/$defs/number"
      }
    }
  },
  "required": [
    "numbers"
  ],
  "additionalProperties": false
},
{
  "type": "object",
  "properties": {
    "topologies": {
      "type": "array",
      "items": {
        "$ref": "#/$defs/network_topology"
      }
    }
  },
  "required": [
    "topologies"
  ],
  "additionalProperties": false
},
{
  "type": "object",
  "properties": {

```

```

"topology_relations": {
  "type": "array",
  "items": {
    "$ref": "#/$defs/network_topology_relation"
  }
},
"required": [
  "topology_relations"
],
"additionalProperties": false
},
"$defs": {
  "keyRef": {
    "type": "object",
    "properties": { "key": { "type": "string" } },
    "required": [ "key" ],
    "additionalProperties": false
  },
  "optionalKeyRef": { "type": [ "object", "null" ],
    "properties": {
      "key": { "type": "string" }
    },
    "additionalProperties": false
  },
  "value": {
    "type": "object",
    "properties": {
      "from": { "type": [ "integer" ], "default": 0, "minimum": 0 },
      "to": { "type": [ "integer" ], "default": 0, "minimum": 0 }
    },
    "required": [ "from", "to" ],
    "additionalProperties": false
  },
  "logical_composite": {
    "type": "object",
    "properties": {
      "class": { "type": "string", "enum": [ "logical_composite" ] },
      "key": { "type": "string" },
      "name": { "type": [ "string", "null" ] },
      "description": { "type": [ "string", "null" ] },
      "devices": { "type": "array", "items": { "$ref": "#/$defs/keyRef" } },
      "power_units": { "type": "array", "items": { "$ref": "#/$defs/pdu" } },
      "fan_shelves": { "type": "array", "items": { "$ref": "#/$defs/fan_shelf" } } },
    "required": [ "key", "name" ]
  },
  "pdu": {
    "type": "object",
    "properties": {
      "class": { "type": "string", "enum": [ "pdu" ] },
      "key": { "type": "string" },
      "name": { "type": [ "string", "null" ] },
      "description": { "type": [ "string", "null" ] },
      "model_name": { "type": [ "string", "null" ] },
      "model_number": { "type": [ "string", "null" ] },

```

```

    "unit_position": { "type": [ "integer", "null" ], "minimum": 1 },
    "slots": { "type": "array", "items": { "$ref": "#/$defs/slot" } }
  },
  "required": [ "key", "name" ]
},
"fan_shelf": {
  "type": "object",
  "properties": {
    "class": { "type": "string", "enum": [ "fan_shelf" ] },
    "key": { "type": "string" },
    "name": { "type": [ "string", "null" ] },
    "description": { "type": [ "string", "null" ] },
    "slots": { "type": "array", "items": { "$ref": "#/$defs/slot" } }
  },
  "required": [ "key", "name" ]
},
"device": {
  "type": "object",
  "properties": {
    "class": { "type": "string", "enum": [ "device" ] },
    "key": { "type": "string" },
    "name": { "type": [ "string", "null" ] },
    "description": { "type": [ "string", "null" ] },
    "ip_address": { "type": [ "string", "null" ] },
    "mac_address": { "type": [ "string", "null" ] },
    "serial_number": { "type": [ "string", "null" ] },
    "model_name": { "type": [ "string", "null" ] },
    "model_number": { "type": [ "string", "null" ] },
    "manufacturer": { "type": [ "string", "null" ] },
    "firmware_version": { "type": [ "string", "null" ] },
    "slots": { "type": "array", "items": { "$ref": "#/$defs/slot" } },
    "ports": { "type": "array", "items": { "$ref": "#/$defs/port" } } },
  "required": [ "key", "name", "serial_number" ]
},
"slot": {
  "type": "object",
  "properties": {
    "class": { "type": "string", "enum": [ "slot" ] },
    "key": { "type": "string" },
    "name": { "type": [ "string", "null" ] },
    "description": { "type": [ "string", "null" ] },
    "model_name": { "type": [ "string", "null" ] },
    "model_number": { "type": [ "string", "null" ] },
    "manufacturer": { "type": [ "string", "null" ] },
    "unit_position": { "type": [ "integer", "null" ], "minimum": 1 },
    "cards": { "type": "array", "items": { "$ref": "#/$defs/card" } }
  },
  "required": [ "key", "name" ]
},
"card": {
  "type": "object",
  "properties": {
    "class": { "type": "string", "enum": [ "card" ] },
    "key": { "type": "string" },
    "name": { "type": [ "string", "null" ] },
    "description": { "type": [ "string", "null" ] },
    "mac_address": { "type": [ "string", "null" ] },

```

```

"serial_number": { "type": [ "string", "null" ] },
"firmware_version": { "type": [ "string", "null" ] },
"model_name": { "type": [ "string", "null" ] },
"model_number": { "type": [ "string", "null" ] },
"manufacturer": { "type": [ "string", "null" ] },
"unit_position": { "type": [ "integer", "null" ], "minimum": 1 },
"slots": { "type": "array", "items": { "$ref": "#/$defs/slot" } },
"ports": { "type": "array", "items": { "$ref": "#/$defs/port" } }
},
"required": [ "key", "name" ]
},
"port": {
  "type": "object",
  "properties": {
    "class": { "type": "string", "enum": [ "port" ] },
    "key": { "type": "string" },
    "name": { "type": [ "string", "null" ] },
    "description": { "type": [ "string", "null" ] },
    "model_name": { "type": [ "string", "null" ] },
    "ip_address": { "type": [ "string", "null" ] },
    "virtual": { "type": "boolean", "default": false },
    "unit_position": { "type": [ "integer", "null" ], "minimum": 1 },
    "bandwidth_name": { "type": [ "string", "null" ] },
    "bandwidth_group": { "type": [ "string", "null" ] },
    "bandwidth": { "type": [ "integer", "null" ], "minimum": 0 },
    "mtu_size": { "type": [ "integer", "null" ], "minimum": 0 }
  },
  "required": [ "key", "name" ]
},
"number": {
  "type": "object",
  "properties": {
    "class": { "type": "string", "enum": [ "number" ] },
    "key": { "type": "string" },
    "name": { "type": [ "string", "null" ] },
    "related_ci_type": { "type": "string", "enum": [ "Network Interface", "Physical Connection",
"Logical Connection", "Equipment", "Topology" ] },
    "related_ci": { "$ref": "#/$defs/keyRef" },
    "type": { "type": "string", "enum": [ "vlan_range", "vlan_subrange", "vlan", "lag_range",
"lag" ] },
    "vlan_type": { "type": "string", "enum": [ "inner", "outer" ] },
    "value": { "$ref": "#/$defs/value" }
  },
  "required": [ "key", "name", "type", "related_ci_type", "related_ci", "value" ] },
"logical_connection": {
  "type": "object",
  "properties": {
    "class": { "type": "string", "enum": [ "logical_connection" ] },
    "key": { "type": "string" },
    "name": { "type": [ "string", "null" ] },
    "description": { "type": [ "string", "null" ] },
    "model_name": { "type": [ "string", "null" ] },
    "bandwidth_group": { "type": [ "string", "null" ] },
    "bandwidth_name_a_to_z": { "type": [ "string", "null" ] },
    "bandwidth_name_z_to_a": { "type": [ "string", "null" ] },
    "bandwidth_a_to_z": { "type": [ "integer", "null" ], "minimum": 0 },
    "bandwidth_z_to_a": { "type": [ "integer", "null" ], "minimum": 0 },

```

```

"equipment_a": { "$ref": "#/$defs/optionalKeyRef" },
"equipment_z": { "$ref": "#/$defs/optionalKeyRef" },
"port_a": { "$ref": "#/$defs/keyRef" },
"port_z": { "$ref": "#/$defs/keyRef" }
},
"required": [ "key", "name", "equipment_a", "equipment_z", "port_a", "port_z" ]
},
"port_relation": {
  "type": "object",
  "properties": {
    "class": { "type": "string", "enum": [ "port_relation" ] },
    "parent": { "$ref": "#/$defs/keyRef" },
    "child": { "$ref": "#/$defs/keyRef" }
  },
  "required": [ "parent", "child" ]
},
"logical_connection_relation": {
  "type": "object",
  "properties": {
    "class": { "type": "string", "enum": [ "logical_connection_relation" ] },
    "sequence": { "type": [ "integer", "null" ], "minimum": 1 },
    "route": { "type": [ "integer", "null" ], "minimum": 1 },
    "parent": { "$ref": "#/$defs/keyRef" },
    "child": { "$ref": "#/$defs/keyRef" }
  },
  "required": [ "parent", "child" ]
},
"network_topology": {
  "type": "object",
  "properties": {
    "class": { "type": "string", "enum": [ "network_topology" ] },
    "key": { "type": "string" },
    "name": { "type": "string" },
    "model_name": { "type": [ "string", "null" ] },
    "devices": { "type": "array", "items": { "key": "#/$defs/optionalKeyRef" } },
    "logical_connections": { "type": "array", "items": { "key": "#/$defs/optionalKeyRef" } }
  },
  "required": [ "key", "name", "devices", "logical_connections" ]
}
}
}
}

```

2. ETLから接続性を切り離す

デバイスインタラクションロジックを変換ロジック (ETL 取り込み) から分離することで、柔軟性を確保します。ユーザーは、ETL ロジックとは無関係に独自のコレクターを開発できます。

次の点を確認してください。

1. 統合スキーマへの接続とデータ変換にのみ焦点を当てるようにコレクターを設計します。
2. EMS / NMSアダプター(Atrinetなどのサードパーティアダプターを含む)を使用または再利用します。
3. 標準化されたデータを ServiceNow インポートセットにプッシュします。

3. CMDB 更新用の汎用 ETL を構成する

単一の再利用可能な ETL を使用して、すべての標準化されたインポートセットを処理し、CMDB を正確に更新します。

次の点を確認してください。

1. インポートセットに標準化された JSON 形式のデータが含まれていることを検証します。
2. TSOM 汎用 ETL は、以下の目的で使用します。
 - エンティティタイプ (スロット、カード、論理インターフェイスなど) を検出します。
 - モデルマッピングロジックに基づいて、各エンティティを適切な CI クラスにマッピングします。
 - 該当する場合は、[インベントリカテゴリ] を含むフィールドに入力します。

特別処理:

- 論理インターフェイス→ポートテーブル (virtual=true)
- 論理接続→論理接続テーブル
- PDU テーブル→ PDU カード
- 設備→ モデルベース CI クラス (複数存在する場合は最新のマッピング)

4. マルチシャーシおよび複合デバイスのサポートを設定する

マルチシャーシルーターなどの複雑なデバイスを適切な CMDB 関係でモデル化します。

次の点を確認してください。

1. 論理複合構成を使用して、ルーター + PDU などのグループ化されたエンティティを表します。
2. 個々のコンポーネント(ファン、管理、スロット)をそれぞれの階層で定義します。
3. TNI モデリングガイダンスに従って各エンティティをマッピングします。

例:7750-2s マルチシャーシ デバイスの場合:

- 論理コンポジット→にはルーターとPDUが含まれています
- PDU →には、スロット→カード→サブスロットが含まれています

5. インストールに基づいて TNI エンティティの作成を有効にする

不必要なレコードを作成することなく、TNI 標準との整合性を確保します。

次の点を確認してください。

- TNI がインストールされている場合:
 - 検出された CI ごとに TNI エンティティを自動的に作成します。
 - インベントリカテゴリを適切に設定します (例: 「論理接続」、「インターフェイス」)。
- TNI がインストールされていない場合:孤立したレコードや無効なレコードを避けるために、TNI エンティティの作成をスキップします。

Nokia Altiplano サービスグラフコネクタの構成

Nokia Altiplano Access Network SDN コントローラーから ServiceNow 構成管理データベース (CMDB) に物理インベントリデータと論理インベントリデータをインポートするように、Nokia Altiplano サービスグラフコネクタを構成します。

この統合では、(MID サーバー経由で) REST API を使用して、TM Forum ベースのデータモデルに沿った正確で最新の通信インベントリが CMDB に反映されるようにします。

- i** 注: このコネクタを使用するには、有効な 通信事業サービスオペレーション管理 サブスクリプションが必要です。

必須のプラグイン

プラグイン	ID
電気通信サービスオペレーションコア	sn_tsom_core
パターンデザイナー	com.snc.pattern.designer
ディスカバリーコア (ディスカバリーとともに自動的にインストールされます)	com.snc.discovery.core
ITOM Discovery ライセンス	com.snc.itom.discovery.license
ITOM ライセンス	com.snc.itom.license
統合ハブ ETL (開発環境のみ)	sn_int_studio

- i** 注: 外部要件:
 - REST ノースバウンド API にアクセスできる、実行中の Nokia Altiplano コントローラーインスタンス。
 - Altiplano インスタンスへの安全な接続を備えた MID サーバー。

構成タスクの概要

次のセクションは、Nokia Altiplano ナビゲーションペインで使用できます。次のテーブルをガイド付きセットアップに使用するか、手動で構成を実行します。

セクション	説明
セットアップ	MID サーバーを設定し、Altiplano 接続を定義し、インポートをスケジュールします。
データソース	一括ディスカバリーおよびフィルター済みディスカバリー用の事前定義されたデータソース (SGC-Nokia Altiplano Bulk Discovery、SGC-Nokia Altiplano Filtering Discovery)。必要に応じて並列ロードを有効にします。並列ロードの詳細については、「 Nokia Altiplano の同時インポートと並列ロードの構成 」を参照してください。
スケジュールをインポート	Altiplano の各接続エイリアスのスケジュールを管理します。手動または定義された間隔でジョブを実行します。
接続および資格	各 Altiplano インスタンスのエイリアスを定義します。接続メタデータと認証情報を格納します。

セクション	説明
格情報 エイリアス	
接続	URL、選択した MID サーバー、認証情報参照、接続エイリアス参照などの Altiplano インスタンスの詳細を定義します。
認証情報	基本認証を使用して Altiplano 認証情報を作成します。
フィルター	フィルター済みディスカバリーで使用されるフィルターパラメーターを設定します (デバイスの IP や名前など)。
プロパティ	コネクタ固有のプロパティを使用してシステムの動作を変更します。詳細については、「 Nokia Altiplano とともにインストールされるシステムコンポーネント 」を参照してください。

サポートされている **Nokia Altiplano** バージョン

- サポートされている最小バージョン:24.6
- 対応OLT:Lightspan MF-2

i 注: 詳細については、「[CMDB での Nokia Altiplano CI と関係のマッピング](#)」を参照してください。

ガイド付きセットアップへのアクセス

ガイド付きセットアップを使用して、構成プロセスを簡素化します。このセットアップでは、整理された一連のステップを使用して、統合を迅速かつ正しく完了することができます。ガイド付きセットアップにアクセスするには:

1. 移動先 [すべて](#) > [サービスグラフコネクタ](#) > **Nokia Altiplano** > [セットアップ](#)。
2. タスクシーケンスに従って、MID サーバー設定、Nokia Altiplano 接続、フィルター、およびインポートスケジュールを構成します。詳細については、「[Nokia Altiplano をインストールして構成する](#)」を参照してください。

関連トピック

- [Nokia Altiplano をインストールして構成する](#)
- [複数の Nokia Altiplano インスタンスの設定](#)
- [Nokia Altiplano の同時インポートと並列ロードの構成](#)
- [CMDB での Nokia Altiplano CI と関係のマッピング](#)
- [Nokia Altiplano サービスグラフコネクタのインポートの実行と検証](#)
- [Nokia Altiplano とともにインストールされるシステムコンポーネント](#)
- [Nokia Altiplano 経由の電気通信ディスカバリー](#)

Nokia Altiplano をインストールして構成する

Nokia Altiplano のサービスグラフコネクタをインストールして構成する方法 (オプションのデモデータ、接続のセットアップ、データ収集スケジュールなど) について説明します。

始める前に

必要なロール: admin

このタスクについて

Nokia Altiplano アプリケーションスコープのサービスグラフコネクタで操作していることを確認します。Nokia Altiplano のサービスグラフコネクタスコープに切り替えるには:

1. ヘッダーのアプリケーションピッカー (🌐) を選択します。
2. **Nokia Altiplano** を検索します。
3. リストから [Nokia Altiplano のサービスグラフコネクタ] を選択します。

手順

1. 移動先 **すべて** > サービスグラフコネクタ > **Nokia Altiplano** > セットアップ。
2. [はじめに] ページで、[開始] を選択します。
ガイド付きセットアップのホームページが新しいタブで開きます。
3. オプション: 環境に基づいて MID サーバー またはスキップを作成して構成します。
 - a. [構成] を選択してプロセスを完了します。
 - b. 画面の指示に従ってダウンロードし、インストールします。
 - c. 完了したら [完了としてマーク] を選択します。
 - 📘 注: MID サーバーをインストールして構成する方法の詳細については、「[MID サーバーの構成](#)」を参照してください。
 - d. 検証手順を設定またはスキップします。
 - e. 検証時に [完了としてマーク] を選択します。
すべての MID サーバーの手順が完了したら、接続の構成に進みます。
4. [接続の構成] を使用して、Nokia Altiplano インスタンスへの接続エイリアス、認証情報、および HTTP 接続を作成します
 - a. [Get Started (開始する)] を選択します。
 - b. 接続と資格情報のエイリアスを作成および構成します。
 - i. [Configure] を選択します。
 - ii. [名前] フィールドで、エイリアス名を指定します。
 - iii. 残りのフィールドはデフォルトのままにして、[送信]、[完了としてマーク] の順に選択します。
 - 📘 注: これにより、直接ではなく名前で接続を使用できるようになり、コレクターが CMDB からすべてのアクティブなエイリアスを抽出し、バインドされた HTTP 接続でデータ収集の実行を開始できるようになります。
 - c. [構成] を選択して、Nokia Altiplano コントローラーにアクセスするための認証情報を作成します。
 - i. [名前] フィールドで、エイリアス名を指定します。
 - ii. [ユーザー名] フィールドで、Nokia Altiplano インスタンスのユーザー名を指定します。

iii. [パスワード] フィールドで、Nokia Altiplano インスタンスのパスワードを指定します。

i 注: Nokia Altiplano インスタンスで使用されている認証方法によっては、他の認証フィールドが必要になる場合があります。デフォルトでは、[ベーシック認証情報](#) を使用します (ガイド付きセットアップの一部として)。

iv. 残りのフィールドはデフォルトのままにして、送信 を選択してから、完了としてマーク を選択します。

d. [構成] を選択して HTTP 接続を作成

i. [名前] フィールドで、接続名を指定します。

ii. 前に作成した 認証情報 と 接続エイリアス を選択します。

iii. Nokia Altiplano の 接続 URL を指定します。

iv. [MID サーバーを使用] をオンにし、次のいずれかを選択します。

- 自動選択
- 特定の MID サーバー
- 特定の MID クラスタ

v. 残りのフィールドはデフォルトのままにして、送信 を選択してから、完了としてマーク を選択します。

5. [開始] をクリックしてから [構成] をクリックして、一括ディスカバリーまたはフィルター済みディスカバリーのいずれかのデータ収集スケジュールを設定します。

移動先	必要な操作
<p>バルクディスカバリーのデータ収集をスケジュール (接続エイリアスのすべてのデバイスに対してバルクディスカバリーデータソースを使用)</p>	<ul style="list-style-type: none"> ○ スケジューラ ーの名前 を指定します。 ○ バルクデータソース SGC-Nokia Altipano Bulk Discovery が選択されていることを確認します。 ○ アクティブに設定すると、スケジューラーは [実行] フィールドと [時間] フィールドに記載されているとおりに自動的に実行されます。非アクティブな場合は、手動で実行する必要があります。 ○ [接続を使用] を選択します。 ○ [接続] フィールドで、接続エイリアスを選択します。 例:sn_sgc_altiplano。Atiplano_Alias。 ○ [実行] フィールドで、頻度を選択します。このスケジュールを実行するタイミングを指定します ([日次]、[週次]、[月次]、[定期的]、[親の実行後]、または [1 回])。 ○ [時間] フィールドに、時間、分、秒単位で時間を入力します。

移動先	必要な操作
<p>フィルタリングされたディスカバリーのデータ収集をスケジュールする (接続エイリアスごとの特定の OLT フィルター基準にフィルタリングディスカバリーデータソースを使用)</p>	<p>a. フィルタリングパラメーターを追加</p> <ul style="list-style-type: none"> i. [接続エイリアス] フィールドで、接続エイリアスを選択します。 ii. 必要に応じて、[フィルター済み IP] フィールドで、次のように定義されたさまざまな形式の IP フィルターを追加します。 <ul style="list-style-type: none"> ▪ 単一 IP: 10.10.10.10 ▪ リスト: 10.10.10.10、10.10.10.20 ▪ 範囲: 10.10.10.30-10.10.10.40 ▪ CIDR: 10.10.10.0/24 ▪ マスク: 10.10.10.0:255.255.255.0 iii. 必要に応じて、デバイス名または名前の一部で検索する名前フィルターを追加します。単一の文字列または名前のリストを指定して、結果をフィルタリングできます。 <p>b. フィルター済み収集をスケジュール</p> <ul style="list-style-type: none"> ▪ スケジューラ ーの名前 を指定します。 ▪ データソース [SGC-Nokia Altipano Filtering Discovery] が選択されていることを確認します。 ▪ アクティブに設定すると、スケジューラーは [実行] フィールドと [時間] フィールドに記載されているとおりに自動的に実行されます。非アクティブな場合は、手動で実行する必要があります。 ▪ [実行] フィールドで、頻度を選択します。このスケジュールを実行するタイミングを指定します ([日次]、[週次]、[月次]、[定期的]、[親の実行後]、または [1 回])。 ▪ [時間] フィールドに、時間、分、秒単位で時間を入力します。 ▪ [接続を使用] を選択します。 ▪ [接続] フィールドで、接続エイリアスを選択します。 例:sn_sgc_altiplano。Atiplano_Alias。

6. [送信] を選択し、[完了としてマーク] を選択します。

結果

構成された接続、エイリアス、認証情報、およびインポートスケジュールには、ナビゲーションからもアクセスできます 移動先 [すべて > サービスグラフコネクタ > Nokia Altiplano](#)。

次のスナップショットは、Nokia Altiplano サービスグラフコネクタの構成セットアップを確認するのに役立ちます。

100% Complete

Nokia Altiplano Integration with CMDB Filter Show all ▼

100%

Status: Completed

Edit

MID Server Skip

The MID Server runs as a Windows service or a UNIX daemon to facilitate communication and the movement of data between a ServiceNow instance and external applications, data sources, and services. Complete the activities in this category to create a user for the MID Server, download the installation package, and validate the MID Server after installation.

After completing each task, mark the step as complete or skip not relevant tasks.

3 / 3 Tasks completed

- ✓ Create MID User
- ✓ Download & Install MID
- ✓ Validate MID

100%

Status: Completed

Edit

Configure Connectivity

Configuration of information necessary for the Nokia Altiplano Collector to work properly.

After completing each task, mark the step as complete or skip not relevant tasks.

3 / 3 Tasks completed

- ✓ Create Connection Aliases
- ✓ Create Credentials
- ✓ Create HTTP Connection

100%

Status: Completed

Edit

Configure Data Collection schedule Skip

Configure Data collection schedule per Connection and Data Source for Nokia Altiplano inventory data integration.

Use Bulk discovery data source for all devices in connection alias.

Use Filtering discovery data source for specific OLT filter criteria for each connection alias.

After completing each task, mark the step as complete or skip not relevant tasks.

2 / 2 Tasks completed

- ✓ Schedule Data Collection for Bulk Discovery
- ✓ Schedule Data Collection for Filtered Discovery

関連トピック

[複数の Nokia Altiplano インスタンスの設定](#)

[Nokia Altiplano の同時インポートと並列ロードの構成](#)

[CMDB での Nokia Altiplano CI と関係のマッピング](#)

[Nokia Altiplano とともにインストールされるシステムコンポーネント](#)

複数の **Nokia Altiplano** インスタンスの設定

単一の ServiceNow 環境内で複数の Nokia Altiplano インスタンスを構成および管理する方法について説明します。これにより、アドミニストレーターとインテグレーターは、独立した接続エイリアスを作成し、特定のフィルタリングまたは頻度のニーズに合わせてカスタマイズされたスケジュールをインポートできます。

始める前に

必要なロール：admin

このタスクについて

以下が揃っていることを確認します。

- NOKIA Altiplano のサービスグラフコネクタアプリケーションスコープで操作しています。
- Nokia Altiplano のサービスグラフコネクタをインストールしました。
- 最初の Altiplano インスタンスのガイド付きセットアップが完了しました。
- MID サーバーのセットアップと検証。

追加の Nokia Altiplano インスタンスを構成することも、同じ Altiplano インスタンスを異なる接続エイリアスとインポートスケジュールで再利用することもできます。新しい Altiplano インスタンスを追加するには、ガイド付きセットアップを再実行して新しい接続エイリアスを構成します。新しいエイリアスごとに、セットアップのすべての接続ステージを実行します。

手順

1. 移動先 [すべて](#) > [サービスグラフコネクタ](#) > **Nokia Altiplano** > [セットアップ](#).
2. [はじめに] ページで、[開始] を選択します。

[構成済みの接続] セクションのすべての手順を繰り返します。Nokia Altiplano の新しいインスタンスの構成エントリを作成します。詳細については、「[Nokia Altiplano をインストールして構成する](#)」を参照してください。

3. 接続エイリアス、認証情報、および HTTP 接続を作成して、接続を構成します。
 - a. [接続の構成] セクションで、[開始] を選択します。
 - b. [構成] を選択して、一意のエイリアス名 (Altiplano_Prod_01 など) を入力して新しい接続エイリアスを作成します。
 - c. [送信] を選択し、ステップを完了としてマークします。
4. 新しい Altiplano インスタンスの認証情報を構成します。
 - a. Altiplano コントローラーのユーザー名とパスワードを指定します。
 - b. ステップを送信し、完了としてマークします。
5. HTTP 接続を構成します。
 - a. 接続名を入力します。
 - b. 新しく作成された認証情報と接続エイリアスを選択します。
 - c. Nokia Altiplano インスタンスの接続 URL を入力します。
 - d. [MID サーバーを使用] を有効にし、適切な MID オプションを選択します。
 - e. [送信] を選択し、ステップを完了としてマークします。
6. インポートスケジュールを構成して、一括ディスカバリーまたはフィルター済みディスカバリーを使用してデータインポートをスケジュールします。
 - a. フィールドに入力します。
詳細については、「[Nokia Altiplano をインストールして構成する](#)」を参照してください。
 - b. [接続を使用] フィールドで、新しい Altiplano インスタンスを選択します。
7. **[Submit (送信)]** を選択します。

結果

新しいインスタンスのセットアップが成功したことを確認するには、構成を確認します。

- [すべての>サービスグラフコネクタ] > Nokia Altiplano >接続エイリアスに移動し、新しいエイリアスが表示されていることを確認します。
- [すべての>サービスグラフコネクタ] > Nokia Altiplano >インポートスケジュール] に移動し、新しいインポートスケジュールがリストされていることを確認し、各エイリアスに対応するジョブが存在することを確認します。

同じ Nokia Altiplano インスタンスに複数の接続エイリアスを設定できます。この柔軟性により、異なる頻度でインポートを実行し、各エイリアスに異なるフィルターを適用できます。

例:Altiplano_Weekly_OLT_10.10.10.*:毎週実行され、10.10.10.10.* の範囲の IP を持つ OLT をフィルタリングします。Altiplano_Daily_Prod:毎日実行され、名前に「prod_olt」が含まれる OLT をフィルタリングします。

関連トピック

[Nokia Altiplano の同時インポートと並列ロードの構成](#)

[CMDB での Nokia Altiplano CI と関係のマッピング](#)

[Nokia Altiplano とともにインストールされるシステムコンポーネント](#)

Nokia Altiplano の同時インポートと並列ロードの構成

同時インポートを構成し、並列ロードを有効にすることで、Nokia Altiplano からの大規模なデータのインポートの効率を向上させます。この構成により、プラットフォームで複数のデータインポートジョブと変換ジョブを同時に実行できます。

始める前に

必要なロール：admin

このタスクについて

同時インポートと並列ロードは、大きなデータセットを小さなパーティションに分割し、複数の変換ジョブとコレクションジョブを並列に実行できるようにすることで、パフォーマンスを向上させます。この設定は、一括ディスカバリーまたはフィルター済みディスカバリー中に大量の機器、論理接続、およびネットワークポロジデータをインポートする場合に便利です。

パフォーマンスを最適化するには、次のガイダンスを使用します。

- 変換が遅い場合:インポートスケジュールで同時インポートを有効にします。
- データ収集が遅い場合:同時インポートを有効にすることに加えて、データソースで並列ロードを構成し、システムプロパティを更新します。

i 注: 並列ロードは、同時インポートに加えて機能します。それはそれにとって代わるものではありません。

手順

1. 移動先 **すべて** > サービスグラフコネクタ > **Nokia Altiplano** > スケジュールのインポート。
2. 同時インポートまたは同時インポートの構成

移動先	必要な操作
同時インポートを有効にするには	<ol style="list-style-type: none"> a. ディスカバリージョブのインポート スケジュール に移動します。 b. [同時インポート] チェックボックスをオンにします。 c. [パーティションメソッド] を [カスタムサイズ] に設定します。 d. [パーティションサイズe] フィールドに、パーティションあたりのレコード数を入力します (例:1000)。 <p>i 注: パーティションサイズに基づいてデータセットがインポートセットに分割されます。各インポートセットは並列で処理されるため、データ変換の速度が向上します。</p>
データソースでの並列ロードの構成	<ol style="list-style-type: none"> a. 対応するデータソースレコードに移動します。 b. [並列ロード] チェックボックスをオンにします。 c. Nokia Altiplano >のプロパティに移動します。

移動先	必要な操作
	<p>d. システムプロパティを開く <code>sn_sgc_altiplano.parallel_number_of_data_source</code></p> <p>e. 値を並列ジョブの目的の数 (3 など) に設定します。</p> <p>i 注: デフォルトでは、データソースジョブの数は 1 に設定されています。この値を増やすと、プラットフォームで複数のデータ収集ジョブを同時に実行できます。</p>

関連トピック

- Nokia Altiplano をインストールして構成する
- CMDB での Nokia Altiplano CI と関係のマッピング
- Nokia Altiplano とともにインストールされるシステムコンポーネント

CMDB での Nokia Altiplano CI と関係のマッピング

Nokia Altiplano のサービスグラフコネクタを使用して、検出された物理ネットワークリソースと論理ネットワークリソースを CMDB 内の電気通信向けの構成アイテム (CI) クラスにマッピングします。コネクタは、一貫したサービスモデリング、シャreshレベルのコンポーネントの可視化、および論理的および物理的な関係の自動化をサポートします。

正確な CI 分類と挿入を確認するために、コネクタは強力な変換エンジン (RTE) と識別および調整エンジン (IRE) を使用します。

コネクタは、デバイスタイプ、機能、およびシャresh構造に基づいて、電気通信固有のモデルを使用して、検出された CI を分類して関連付けます。これにより、ベンダー間でクリーンで正規化された CMDB を維持できます。Nokia Altiplano から検出されたモデル名は、スロットおよびサブスロットコンポーネントの ServiceNow 標準モデル識別子とカテゴリに自動的に変換されます。

CI マッピングと関係

次の表では、Altiplano CI が CMDB でどのように表され、物理レイヤーと論理レイヤー間でどのように相互に関連しているかについて説明します。

CMDB CI マッピングと関係 (物理レイヤー)

CI タイプ	CMDB テーブル	説明と関係
OLT CI	cmdb_ci_optical_line_terminal	OLT デバイスを表します。スロット CI と論理ネットワークインターフェイス CI が含まれています。
ONU/ONT CI	cmdb_ci_optical_network_terminal または cmdb_ci_optical_network_unit	ONU または ONT デバイスを表します。クラスは、システムプロパティ <code>sn_sgc_altiplano.onu_ci_class</code> によって決定されます。ネットワークインターフェイス CI が含まれています。
スロット CI	cmdb_ci_container_slot	メインシャreshスロットを表します。OLT CI に含まれます。インターフェイスカード CI (LT/NT、PSU、ファンなど) が含まれています。モデル変換はデータソースに適用されます。詳細については、上記のス

CMDB CI マッピングと関係 (物理レイヤー) (続く)

CI タイプ	CMDB テーブル	説明と関係
		ロットおよびサブスロット CI のモデル変換の表を参照してください。
サブスロット CI	cmdb_ci_container_subslot	インターフェイスカード内のサブコンポーネント (SFP のケージなど) を表します。LT/NT カードに含まれています。トランシーバカード CI が含まれています。詳細については、上記のロットおよびサブスロット CI のモデル変換の表を参照してください。
インターフェイスカード CI	cmdb_ci_interface_card	LT/NT カード、トランシーバー、および制御装置を表します。サブスロットとネットワークインターフェイスを含めることができます。
ネットワークインターフェイス CI	cmdb_ci_ni_interface	物理ポート (PON、イーサネットなど) と論理ポート (VLAN など) の両方表します。インターフェイスカードまたは ONU/ONT CI に含まれています。論理ポートは、Members::Member of を使用して物理ポートに関連付けられます。
論理接続 CI	cmdb_ci_ni_logical_path	OLT と ONU の間の PON や VLAN などの論理パスを表します。ポート A 属性とポート Z 属性で定義され、終端ネットワークインターフェイス CI を参照します。VLAN パスは PON パスを消費します。
IP アドレス CI	cmdb_ci_ip_address	OLT の検出された IP アドレスを表します。対応する OLT CI によって所有されます。

重要な関係の例

- 格納関係
 - スロット CI を含む OLT CI →
 - スロット CI → にインターフェイスカード CI が含まれています
 - インターフェイスカード CI → にサブスロット CI が含まれています
 - サブスロット CI → にはトランシーバーインターフェイスカード CI が含まれています
 - ONU/ONT CI → にはネットワークインターフェイス CI が含まれています
- インターフェイスの関係:
 - 論理接続: ネットワークインターフェイスによって > 終了します
 - ネットワークインターフェイス: ネットワークインターフェイスのメンバー >
- 論理パスの関係:

- VLAN パス (親) → は → PON パス (子) を消費します
- 論理パス→ポート A とポート Z を介して→ネットワークインターフェイス CI で終了
- 所有権:OLT デバイスが所有する IP アドレス CI の→

サポートされているモデル

1. ネットワーク装置モデル (sn_ent_nw_equipment_model)

- サポートされているOLTはNokia Lightspan MF-2で、デフォルトではモデル名は「Nokia MF-2」です。
- ONU / ONTモデルはメーカー+ ONU / ONTです。システムプロパティ sn_sgc_altiplano.onu_ci_class は、ONU と ONT のどちらを使用するかを定義します。
- モデルがモデルテーブルで見つからなかった場合は、CI に新しいモデルが作成されます。CI は「ネットワーク機器」として作成されます

2. 機器ホルダーモデル:(sn_ent_nw_holder_model)

- スロットモデル:「トラフィックスロット」、「ファンスロット」、「パワースロット」
- サブスロットモデル:「SFPサブスロット」
- 使用するモデル名は、Altiplano 拡張ポイント (sn_sgc_altiplano を使用してカスタマイズできます。AltiplanoCustomizedModels)
- モデルがモデルテーブルで見つからなかった場合は、CI に新しいモデルが作成されます。

3. ネットワークカードモデル (sn_ent_nw_card_model)

- カードモデルは、Altiplano API から検出されたモデル名、メーカー、およびモデル番号によって検出されます
- モデルがモデルテーブルで見つからなかった場合は、CI に新しいモデルが作成されます。

4. ネットワークインターフェイスモデル:(sn_ent_nw_interface_model)。

- イーサネットポートモデルは、ネットワークインターフェイステーブル (sn_ent_nw_interface_model) の「ポート帯域幅」列で確認できます。ポート CI のポート帯域幅は、帯域幅テーブルで検出されたポート速度 (帯域幅) によって特定されます
- PON 物理ポートモデル:「PON アクセスインターフェイス」、「PON ネットワークインターフェイス」
- 論理ポートモデル:「ENETインターフェイス」、「VLANインターフェイス」、「LAGインターフェイス」、PON論理インターフェイス
- モデルがモデルテーブルで見つからなかった場合、「モデル ID」への参照は空のままになります。

5. 論理ネットワーク接続モデル (sn_ent_logical_nw_connection_model)

- PON アクセスパス
- VLAN パス
- モデルがモデルテーブルで見つからなかった場合、「モデル ID」への参照は空のままになります。

i 注:

- コネクタが検出された機器を製品モデルテーブルの既存のモデルと一致させることができない場合、CI はデフォルトで ネットワーク機器 として作成されます。
- デモデータがインストールされている場合、OLT、ONU、ONT、スロット、サブスロット、カード、ネットワークインターフェイス、および論理接続のデフォルトモデルが作成されません。
- 機器および機器所有者モデル名は、拡張ポイントを使用してカスタマイズできます

スロットおよびサブスロット CI のモデル変換

取り込み時に、検出された特定のモデル名が事前定義された CMDB モデル識別子にマッピングされ、スロットのカテゴリ設定が一貫していることが確認されます。変換ロジックは SGC データソーススクリプトに埋め込まれており、Nokia Altiplano ソースに適用されます。

ファン、電源、トラフィックスロットなどのスロットコンポーネントは、CMDB の **スロット** カテゴリにマッピングされます。SFP ケージや合成サブスロットなどのサブスロットコンポーネントは、CMDB の **サブスロット** カテゴリにマッピングされます。

例:Nokia **Altiplano** スロットとサブスロットモデルのマッピング

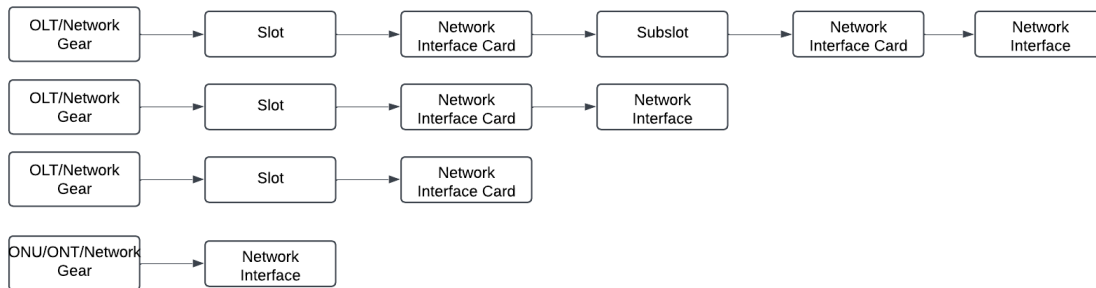
ソース	検出されたモデル名	ターゲット CMDB モデル ID	モデルカテゴリ
アルティプラーノ	slot-fan	ファンスロット	スロット
アルティプラーノ	slot-lt	トラフィックスロット	スロット
アルティプラーノ	籠	SFP サブスロット	サブスロット
アルティプラーノ	slot-nt	トラフィックスロット	スロット
アルティプラーノ	合成 nt-slot	SFP サブスロット	サブスロット
アルティプラーノ	slot-psu	電源スロット	スロット

CI 関係構造

次のインフォグラフィックでは、CI の関係について説明します。

CIs Relationships

Contains::Contained by:



Owns::Owned by (Dependent Relationship)



Members::Member of



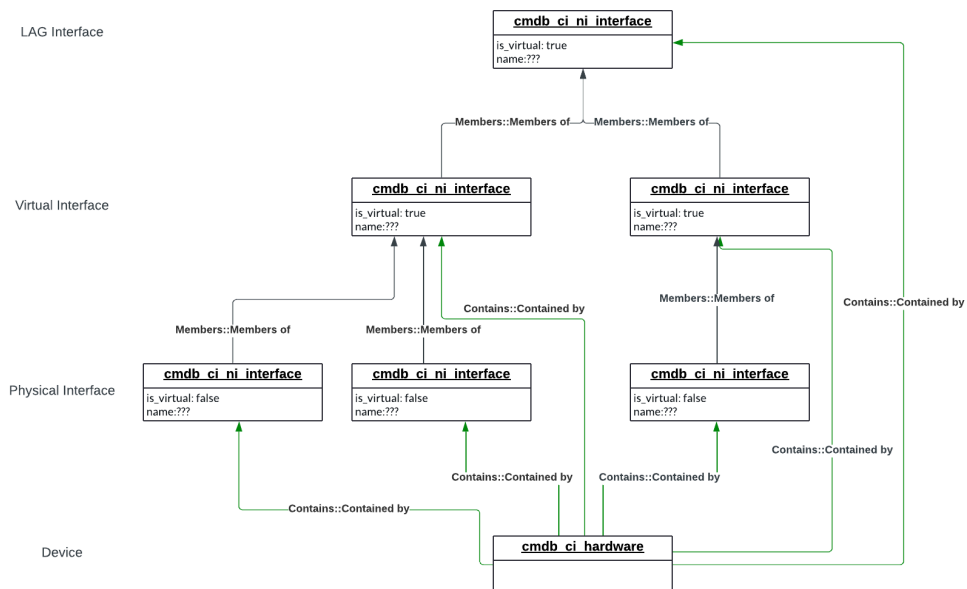
Consumes::Consumed by



Terminates::Terminated by

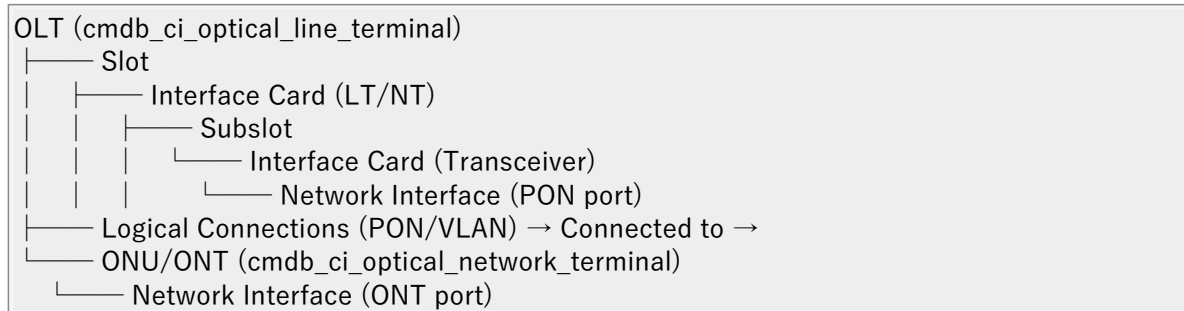


Dependent Relationship →



例:OLTからONUへの構造

次の構造により、OLT からハードウェア レイヤーを介して、接続された ONU および関連する論理パスまでのエンドツーエンドのトレーサビリティが可能になります。



電気通信不一致の識別と調整のアクティブ化

Telecom Discrepancy Identification and Reconciliation 機能 (Telecom Visibility プラグインの一部) をアクティブ化して、ライブネットワークで検出された電気通信ネットワークリソースと、CMDB または電気通信ネットワークインベントリ (TNI) で表されるデータとの一貫性を確保します。この機能は、不一致を自動的に検出して修正するのに役立ち、電気通信環境全体でサービスの正確性とデータの整合性をサポートします。

始める前に

必要なロール：admin

この機能を使用するには、組織に TSOM のアクティブなサブスクリプションが必要です。Telecom Discovery、Telecom Visibility、およびこの機能は、一緒にライセンス供与されます。

このタスクについて

プラグインの依存関係:次のプラグインがインストールされ、アクティブ化されていることを確認します。

プラグイン	ID/アプリ ID	タイプ
電気通信サービスオペレーションコア	sn_tsom_core	ストア
CMDB CI Class Models	sn_cmdb_ci_class	ストア
展開されたモデルと資産クラス	sn_ent	ストア
可視化コンテンツ	sn_pattern_design	ストア
CMDB 共通統合	sn_cmdb_int_util	ストア
ディスカバリーコア	com.snc.discovery.core	ファミリー
ITOM Discovery ライセンス	com.snc.itom.discovery.license	ファミリー
ITOM ライセンス	com.snc.itom.license	ファミリー

手順

1. 電気通信ディスカバリーパターンまたは Nokia Altiplano サービスグラフコネクタをインストールするときに、sn_tsom_core プラグインが自動的にインストールされていることを確認します。
2. 前述の依存プラグインがアクティブになっていることを確認します。

関連トピック

[不一致の識別 – 不一致の種類](#)

[Telecom Discrepancy Identification & Reconciliation とともにインストールされるシステムコンポーネント](#)

[電気通信不一致の識別と調整](#)

[電気通信不一致監査の実行](#)

[調整ルールを使用した CI 属性の更新の制御](#)

監査のフィルターの構成

電気通信不一致監査の範囲を制御するフィルタリング条件を定義します。これらのフィルターにより、必要な CI のセットでのみ監査が実行されることが保証され、パフォーマンスとターゲット設定の精度が向上します。

始める前に

必要なロール：admin

このタスクについて

電気通信不一致監査のフィルタリング条件について

- 監査では、フィルタリング条件を使用して、評価対象の CI の範囲を絞り込みます。
- フィルタリング条件は、監査対象を絞り込むために不可欠です。
- 特定の監査要件を満たすために、監査インスタンスごとに条件を定義および変更できます。
- `discovery_source`によるフィルタリングは、特定の統合ソース (TSOM や Altiplano など) に関連付けられたレコードをフィルタリングするための一般的な条件です。さらに、監査要件に従ってフィルターをカスタマイズできます。
- 監査の各バージョンには、異なるフィルタリング条件を設定できます。

監査フィルターは次のとおりです。

- 1. 電気通信論理接続の不整合監査:**この監査は論理接続テーブルで実行されます。デフォルトのフィルター条件は、TSOM のような `discovery_source` として定義されています。この条件は、論理接続テーブルの各 CI に適用されます。
- 2. 電気通信ネットワークポロジの不整合の監査:**この監査はネットワークポロジテーブルで実行されます。デフォルトのフィルター条件は、TSOM のような `discovery_source` として定義されています。この条件は、ネットワークポロジテーブル内の各 CI に適用されます。
- 3. 電気通信不一致の監査:**この監査は関係テーブルで実行されます。デフォルトのフィルター条件は、TSOM のような `discovery_source` として定義されています。この条件は、機器レベルで各 CI に適用されます。

次のスクリーンショットは、認定フィルターのユーザー・インターフェースを理解するのに役立ちます

Number: CFLR0010004 Active

* Name: Filter for TSOM Logical Connections Discrepancy Audit

Description: Filter for TSOM Logical Connections Discrepancy Audit

Table: Configuration Item [cmdb_ci] Version: 2

Filter condition: No. of records matching the condition: 1 @

All of these conditions must be met

Discovery source contains TSOM AND OR

Name is pon_path_PON_LGPON AND OR

Update

Related Links
Delete inactive versions

Other Versions	Version	Search	Actions on selected rows...
Filter for TSOM Logical Connections Disc...	Filter for TSOM Logical Connections Disc...	Configuration Item [cmdb_ci]	1

手順

1. 移動先 **すべて** > **認定** > **フィルター**.
2. **[New (新規)]** を選択します。
認定フィルターインターフェイスが表示されます。
3. フィールドに入力します。
詳細については、「[認定フィルターの作成](#)」を参照してください。
4. フィルター条件を追加します。
たとえば、ディスカバリーソースに TSOM が含まれています。
5. **[Submit (送信)]** を選択します。
フィルターが作成されます。

結果

このフィルターは、論理接続の監査で使用できます。

関連トピック

[電気通信不一致監査の実行](#)

[電気通信不一致の監査と修正の例](#)

[Telecom Discrepancy Identification & Reconciliation とともにインストールされるシステムコンポーネント](#)

[CMDB 360 での属性値の不一致の構成](#)

[調整ルールを使用した CI 属性の更新の制御](#)

CMDB 360 での属性値の不一致の構成

CMDB 360 で属性比較設定を構成して、複数のディスカバリーソース間でデータの不整合を検出します。

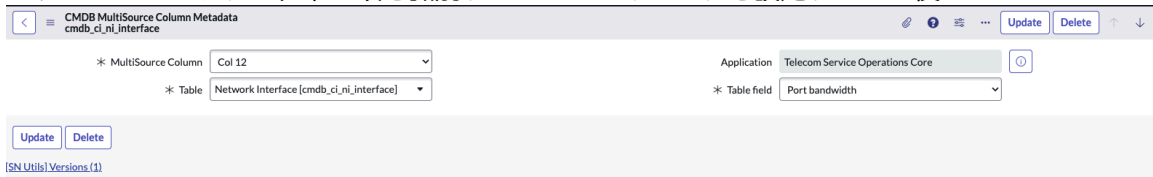
始める前に

必要なロール：admin

CMDB 360 が有効で、インスタンスで構成されていることを確認します。

このタスクについて

次のスクリーンショットは、不一致を識別するためのフィールドを設定するのに役立ちま



す。

手順

1. テーブルに移動 すべて > **cmdb_multisource_column_metadata.list**.
2. [新規] を選択して、新しいレコードを作成します。
3. [マルチソース列] フィールドで、比較する属性を選択します。
4. 属性とフィールドを比較するテーブルを選択します。
 - ❗ 注: 必要なテーブルがリストにない場合は、[テーブル] ラベルを選択して長押し (または右クリック) し、[ディクショナリの構成] を選択します。
5. オプション: 辞書エントリを設定するには、次の手順を実行します。
 - a. 辞書エントリで、[表示] を選択し、[詳細] を選択します。
 - b. [属性] フィールドで、属性を `base_start=true` および `allow_public=true` に設定します。

結果

CMDB 360 クエリで構成された属性を使用できます。

関連トピック

- [調整ルールを使用した CI 属性の更新の制御](#)
- [属性値不一致のレポートを生成](#)

調整ルールを使用した CI 属性の更新の制御

構成アイテム (CI) の特定の属性がディスカバリーや他のデータソースによって上書きされないようにするには、調整ルールを使用します。これらのルールは、複数のソースが値を提供する場合に、特定の属性を更新するために信頼できるデータソースを定義します。

始める前に

必要なロール: admin

このタスクについて

調整ルールは、識別および調整エンジン (IRE) によって処理され、CMDB 内のデータの完全性を維持するために不可欠です。

ユースケース:ディスカバリーが属性のセットを更新できないようにしながら、別のソース (SCCM や手動入力など) による更新を許可する場合は、ディスカバリーソースを除外するルールを定義します。

- ❗ 注: 競合を避けるために、特定の CI クラスと属性の組み合わせに対してアクティブにする必要がある調整ルールは 1 つだけです。

手順

1. 移動先 すべて > **CI** クラスマネージャー。
2. ターゲット CI クラス (cmdb_ci_computer など) を選択します。

3. [調整ルール] タブで、[新規] をクリックしてルールを作成します。
4. ルール名を入力し、ルールを適用するディスカバリーソースを選択します。
5. このルールが管理する属性を指定します。
6. ソースの優先順位を設定して、選択したソースのみが指定された属性を更新できるようにします。

結果

特定の CI 属性は、ディスカバリーなどの信頼できないデータソースや優先度の低いデータソースによって更新されなくなります。ルールで定義されたソース (SCCM、手動更新など) のみがこれらの属性を更新できます。

関連トピック

[CMDB 360 での属性値の不一致の構成](#)

通信事業サービスオペレーション管理の使用

通信事業サービスオペレーション管理 (TSOM) を活用して、電気通信サービスをプロアクティブに監視し、データの整合性を検証し、ネットワークインベントリとディスカバリーソース間の不一致を調整します。TSOM を使用すると、運用チームは正確な通信認識型 CMDB を維持し、リアルタイムのネットワークインサイトに基づいて行動できます。

TSOM の一般的な運用タスク

TSOM が設定されると、その機能を使用して次のことができます。

- 電気通信不一致監査の実行:検出されたデータと調整ルールに基づいて認定監査を実行することで、ネットワークの関係と構成の不整合を特定します。
- CMDB 360 で属性値の不一致を検証する:CMDB 360 の属性値不一致機能を使用して、異なるディスカバリーソース間で属性値を比較し、競合または古いレコードを検出します。
- 電気通信ディスカバリーパターンの使用:電気通信ディスカバリーパターンを適用して、電気通信固有の構成アイテム (CI) とその関係を正確に検出して入力します。
- サービスグラフコネクタからのインポートジョブの実行と検証:インポートスケジュール (Nokia Altiplano コネクタなど) を手動でトリガーして、トポロジとデバイスデータが CMDB に正常にインポートされ、正しい CI クラスにマッピングされていることを確認します。

これらのタスクにより、ServiceNow 内で電気通信ネットワークが継続的に同期され、正確で、運用上表示可能になります。

電気通信ディスカバリーパターンの使用

Telecom ディスカバリーパターンを使用して、ルーター、Cisco、Juniper などのベンダーデバイスからネットワーク機能 (xNF) を識別して分類します。パターンベースディスカバリーを活用して、電気通信リソースを CMDB にマッピングします。

テレコムルーターパターンの xNF の分類

分類される OID の完全なリストにアクセスします。

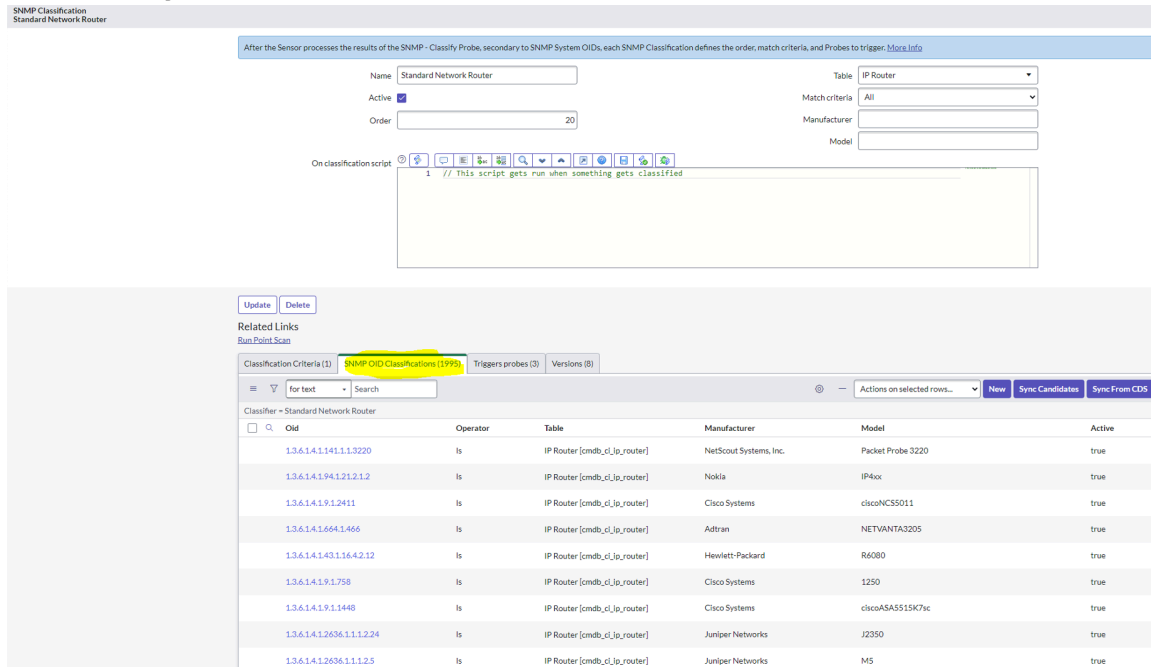
始める前に

必要なロール: admin

分類子名: 標準ネットワークルーター。

手順

1. 移動先 **すべて** > ディスカバリー定義 > **CI** の分類 > **SNMP**.
2. リストから [標準ネットワークルーター] を選択します。
3. **[SNMP OID 分類]** タブを開き、OID のリストを表示しま



す。

i 注:

分類子に OID を追加する方法の詳細については、「[ディスカバリーパターンを使用した直接ディスカバリー](#)」を参照してください。

xNF で使用される MiB テーブル:

- システム MiB
- EntityPhysicalMiB
- IfMiB
- IfXMIB
- IpMiB

テレコム **Cisco 7613** ルータ パターンの **xNF** の分類

分類される OID の完全なリストにアクセスします。

始める前に

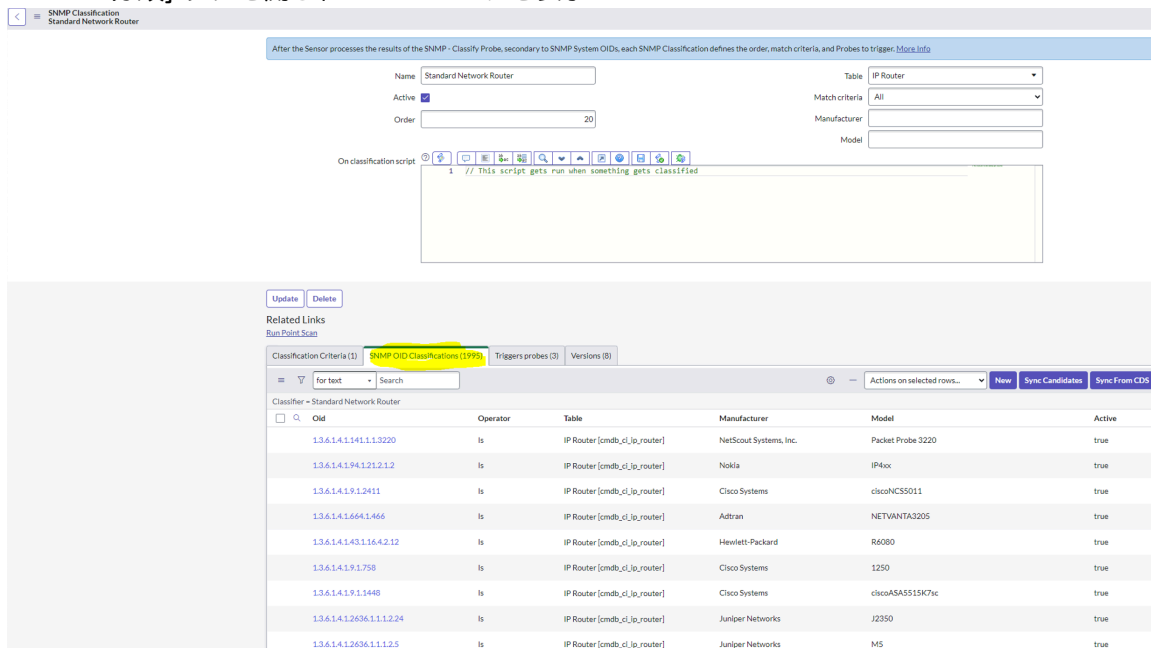
必要なロール : admin

分類子名: 標準ネットワークルーター。

手順

1. 移動先 **すべて** > ディスカバリー定義 > **CI** の分類 > **SNMP**.
2. リストから [標準ネットワークルーター] を選択します。

3. [SNMP OID 分類] タブを開き、OID のリストを表示しま



す。

注:

分類子に OID を追加する方法の詳細については、「[ディスカバリーパターンを使用した直接ディスカバリー](#)」を参照してください。

このパターンを呼び出す特定の OID のリスト:

Vendor (ベンダー)	モデル	OID	パターン
Cisco	7613	1.3.6.1.4.1.9.1.528	Telecom Cisco 7613 ルータ

xNF で使用される MiB テーブル:

- システム MIB
- EntityPhysicalMIB
- IfMIB
- IfXMIB
- IpMIB

Telecom Juniper MX SSH ルーターパターンの xNF の分類

分類される OID の完全なリストにアクセスします。

始める前に

必要なロール : admin

分類子名: 標準ネットワークルーター。

手順

1. 移動先 **すべて > ディスカバリー定義 > CI の分類 > SNMP.**
2. リストから **[標準ネットワークルーター]** を選択します。

3. [SNMP OID 分類] タブを開き、OID のリストを表示しま

After the Sensor processes the results of the SNMP - Classify Probe, secondary to SNMP System OIDs, each SNMP Classification defines the order, match criteria, and Probes to trigger. [More Info](#)

Name: Standard Network Router
 Active:
 Order: 20
 Table: IP Router
 Match criteria: All
 Manufacturer:
 Model:
 On classification script:

```
1 // This script gets run when something gets classified
```

Update Delete
 Related Links
 Run Point Scan

Classification Criteria (1) **SNMP OID Classifications (199)** Triggers probes (3) Versions (3)

Classifier - Standard Network Router

OID	Operator	Table	Manufacturer	Model	Active
1.3.6.1.4.1.141.1.1.3220	Is	IP Router [cmdb_ci_ip_router]	NetScout Systems, Inc.	Packet Probe 3220	true
1.3.6.1.4.1.94.1.2.1.2.1.2	Is	IP Router [cmdb_ci_ip_router]	Nokia	IP4xx	true
1.3.6.1.4.1.9.1.2411	Is	IP Router [cmdb_ci_ip_router]	Cisco Systems	ciscoNCS5011	true
1.3.6.1.4.1.664.1.466	Is	IP Router [cmdb_ci_ip_router]	Adtran	NETVANTA3205	true
1.3.6.1.4.1.43.1.16.4.2.12	Is	IP Router [cmdb_ci_ip_router]	Hewlett-Packard	R6080	true
1.3.6.1.4.1.9.1.758	Is	IP Router [cmdb_ci_ip_router]	Cisco Systems	1250	true
1.3.6.1.4.1.9.1.1440	Is	IP Router [cmdb_ci_ip_router]	Cisco Systems	ciscoSAS515K79c	true
1.3.6.1.4.1.2636.1.1.1.2.24	Is	IP Router [cmdb_ci_ip_router]	Juniper Networks	J2350	true
1.3.6.1.4.1.2636.1.1.1.2.25	Is	IP Router [cmdb_ci_ip_router]	Juniper Networks	M5	true

す。

注:

分類子に OID を追加する方法の詳細については、「[ディスカバリーパターンを使用した直接ディスカバリー](#)」を参照してください。

このパターンを呼び出す特定の OID のリスト:

Vendor (ベンダー)	モデル	OID	パターン
Juniper	MX80	1.3.6.1.4.1.2636.1.1.1.2.57	Telecom Juniper MX SSH ルーター
Juniper	MX104	1.3.6.1.4.1.2636.1.1.1.2.97	Telecom Juniper MX SSH ルーター
Juniper	MX240	1.3.6.1.4.1.2636.1.1.1.2.29	Telecom Juniper MX SSH ルーター
Juniper	MX480	1.3.6.1.4.1.2636.1.1.1.2.25	Telecom Juniper MX SSH ルーター

xNF:SystemMIB で使用される MiB テーブル。

使用する CLI コマンド。

- シャーシハードウェアを表示 |ノーモア |XML を表示
- インターフェイスメディアを表示 |ノーモア |XML を表示

電気通信 Cisco スイッチパターンの xNF の分類

分類される OID の完全なリストにアクセスします。

始める前に

必要なロール : admin

分類子名: 標準ネットワークスイッチ。

手順

1. 移動先 **すべて > ディスカバリー定義 > CI の分類 > SNMP.**
2. リストから [標準ネットワークスイッチ] を選択します。
3. [SNMP OID 分類] タブを開き、OID のリストを表示しま

The screenshot shows the configuration page for 'Standard Network Switch' under 'SNMP Classification'. It includes fields for Name, Active, Order, Table, Match criteria, Manufacturer, and Model. Below these is a section for 'On classification script' with a code editor containing a comment: '// This script gets run when something gets classified'. At the bottom, the 'SNMP OID Classification Criteria (427)' tab is active, displaying a table of classification criteria.

Classifier	OID	Operator	Table	Manufacturer	Model	Active
Standard Network Switch	1.3.6.1.4.1.1916.2.98	Is	IP Switch [cmdb_ci_ip_switch]	Extreme Networks	SUMMITX250-24KDC	true
Standard Network Switch	1.3.6.1.4.1.14179.2.1.1.22.57	Is	IP Switch [cmdb_ci_ip_switch]	Airespace, Inc	Cisco 1830I Unified Access Point	true
Standard Network Switch	1.3.6.1.4.1.9.1.1071	Is	IP Switch [cmdb_ci_ip_switch]	Cisco Systems	ciscoPwrC900Poe	true
Standard Network Switch	1.3.6.1.4.1.6527.6.2.1.2.1	Is	IP Switch [cmdb_ci_ip_switch]	Alcatel-Lucent Enterprise	7210 SAS-E	true
Standard Network Switch	1.3.6.1.4.1.25506.1.246	Is	IP Switch [cmdb_ci_ip_switch]	Hewlett-Packard	H3C S3100-26TP-EI	true
Standard Network Switch	1.3.6.1.4.1.25506.1.38	Is	IP Switch [cmdb_ci_ip_switch]	Hewlett-Packard	H3C S3600-52P-EI	true
Standard Network Switch	1.3.6.1.4.1.9.1.1637	Is	IP Switch [cmdb_ci_ip_switch]	Cisco Systems	ciscoE140D	true
Standard Network Switch	1.3.6.1.4.1.25506.1.654	Is	IP Switch [cmdb_ci_ip_switch]	H3C Technologies	S3100V2-16TP-SI	true
Standard Network Switch	1.3.6.1.4.1.1991.1.3.39.1	Is	IP Switch [cmdb_ci_ip_switch]	Brocade Communications Systems	Netron IMR 640 Switch	true
Standard Network Switch	1.3.6.1.4.1.2.6.157	Is	IP Switch [cmdb_ci_ip_switch]	IBM	8275-416	true

す。

注:

分類子に OID を追加する方法の詳細については、「[ディスカバリーパターンを使用した直接ディスカバリー](#)」を参照してください。

このパターンを呼び出す特定の OID のリスト:

Vendor (ベンダー)	モデル	OID	パターン
Cisco	ネクサス9000	1.3.6.1.4.1.9.12.3.1.3.1954	電気通信 Cisco スイッチ
Cisco	ネクサス 3548	1.3.6.1.4.1.9.12.3.1.3.1666	電気通信 Cisco スイッチ

xNF で使用される MiB テーブル:

- システム MIB
- EntityPhysicalMIB
- IfMIB
- IfXMIB
- IpMIB

電気通信スイッチパターンの xNF の分類

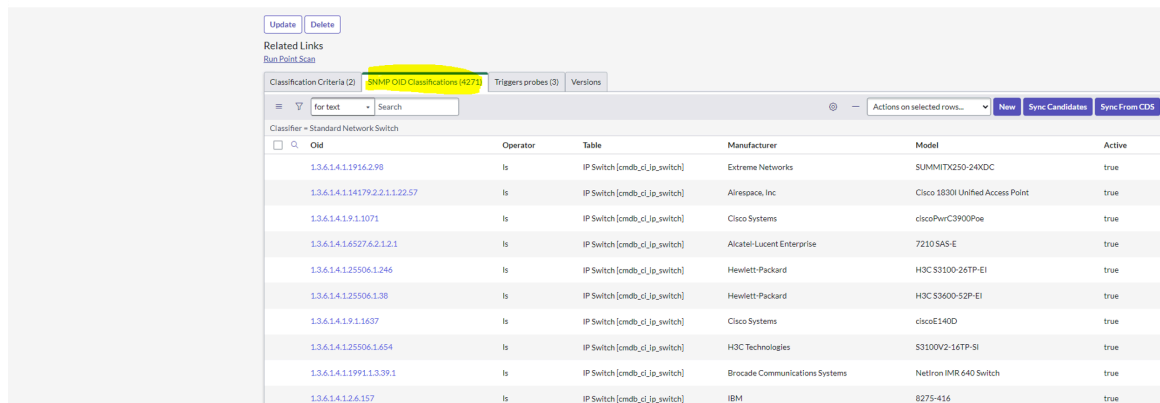
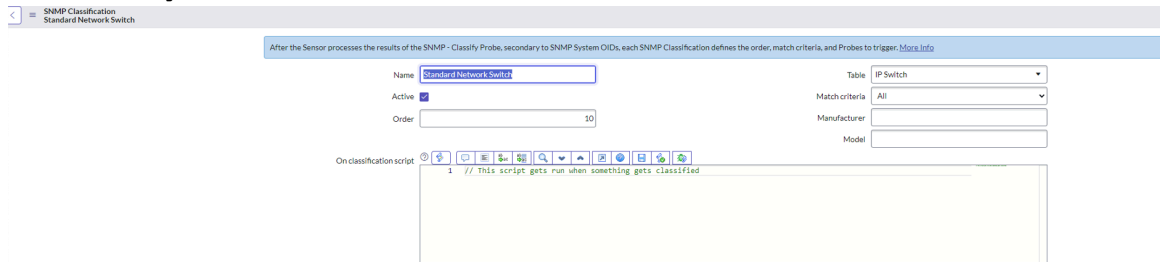
分類される OID の完全なリストにアクセスします。

始める前に
必要なロール：admin

分類子名: 標準ネットワークスイッチ。

手順

1. 移動先 すべて > ディスカバリー定義 > CI の分類 > **SNMP**.
2. リストから [標準ネットワーク スイッチ] を選択します。
3. **[SNMP OID 分類]** タブを開き、OID のリストを表示しま



す。

注:

分類子に OID を追加する方法の詳細については、「[ディスカバリーパターンを使用した直接ディスカバリー](#)」を参照してください。

xNF で使用される MIB テーブル:

- システム MIB
- EntityPhysicalMIB
- IfMIB
- IfXMIB
- IpMIB

Nokia Altiplano サービスグラフコネクタのインポートの実行と検証

Nokia Altiplano サービスグラフコネクタの構成済みインポートスケジュールを手動で実行します。インポートを実行して、コネクタのセットアップを検証したり、アドホックインポートを実行したり、新しく構成された接続エイリアスをテストしたりすることもできます。このタスクは、Altiplano からのデータが CMDB で正常にインポートまたは更新されたことを確認するのに役立ちます。

始める前に
必要なロール：admin

このタスクについて

Nokia Altiplano サービスグラフコネクタを構成し、1 つ以上のインポートスケジュールを設定した後、インポートジョブを実行できます。このタスクは、統合が期待どおりに機能していること、および CI が CMDB で正常にインポートまたは更新されていることを確認するのに役立ちます。定義された頻度に基づいてインポートスケジュールを自動的に実行することも、スケジュールを手動でトリガーしてすぐに実行して検証することもできます。

- 注:** (異なる環境またはフィルターに対して) 複数の Altiplano 接続エイリアスを構成した場合は、各エイリアスに対してインポートスケジュールが正しく設定されていることを確認します。

次のスクリーンショットは、スケジュール設定済みデータインポートプロセスを理解するのに役立ち、実行内容がインポートセットの形式で表示されます。

The screenshot displays the configuration interface for a scheduled data import task. The task name is 'Altiplano bulk'. The data source is 'SGC-Nokia Altiplano Bulk Discovery'. The application is 'Service Graph Connector for NOKIA Altiplano'. The run frequency is set to 'Daily'. The connection is 'sn_sgc_altiplano.Altiplano1'. Below the configuration, there is a table of executions:

Number	Execution Context	Import Set	Created
EXE0001043	EC00001043	Import Set: ISET0010043	2025-07-13 00:00:00
EXE0001042	EC00001042	Import Set: ISET0010042	2025-07-12 23:36:01
EXE0001040	EC00001040	Import Set: ISET0010040	2025-07-12 00:00:00

手順

1. 移動先 **すべて > Nokia Altiplano > スケジュールをインポート**.
2. 実行するインポートスケジュールを選択します。
スケジュール設定済みデータインポートのリストが表示されます。
3. スケジュール設定済みデータインポートを選択します。
4. 次のいずれかの操作を行います。
 - スケジュール がアクティブである場合、ジョブは定義された間隔で自動的に実行されます。したがって、実行されるまで待ちます。
 - インポートをすぐに実行するには、[今すぐ実行] を選択します。
5. 実行を監視して結果を検証するには:
 - a. [実行] リストまたはインポートスケジュールにリンクされているテーブルまで下にスクロールします。
 - b. 実行によって作成された最新の インポートセット レコードを開きます。
6. インポートセットログを確認して、読み取られた行数、CMDB で挿入または更新された行数、または変換の成功ステータスを確認します。
7. オプション: 移動先 **すべて > CMDB > CI** クラス またはカスタム通信テーブル (TNI ベースアイテムなど) を使用して、CI が期待どおりに作成または更新されたことを確認します。

結果

インポートスケジュールはすぐに実行され、インポートセットの実行と対応する CI の更新を CMDB で確認できます。正しく構成されている場合、コネクタは Nokia Altiplano から ServiceNow インスタンスにネットワークインベントリデータを取り込みます。

例:インポートスケジュールを実行した後:

- ログには、「12 行が読み込まれ、0 行が挿入/更新されました」と表示されます。これは、CI が存在することを示しています。
- 既存の CI を削除してインポートを再実行すると、ログに複数の挿入と更新が表示され、エンドツーエンドの機能を検証することがあります。

i 注:

- インポートスケジュールで 同時インポート オプションが有効になっている場合は、標準のインポートセットテーブルではなく、同時インポートセット テーブルにレコードが表示されます。
- 各同時実行ジョブは、独自のインポートセットとログエントリを作成します。実行レコードの構造は変わりません。

関連トピック

[Nokia Altiplano サービスグラフコネクタの構成](#)

[Nokia Altiplano 経由の電気通信ディスカバリー](#)

TSOM スキーマバリデータを使用した JSON ペイロードの検証

データをインポートする前に、TsomSchemaValidator ユーティリティクラスを使用して、TSOM スキーマに対して JSON ペイロードを検証します。これにより、エラーを早期に特定し、ETL の障害を減らし、データ品質を確認できます。

このバリデーターを使用して、インポートセットを作成する前に、JSON ペイロードが電話会社オブジェクト (デバイス、論理接続、トポロジなど) に期待されるスキーマに準拠しているかどうかを確認します。この事前検証ステップは、スキーマ不一致エラーの防止とデバッグの改善に役立ちます。

サポートされているスキーマタイプ

バリデーターは、さまざまな電話会社のデータ構造に対して複数のスキーマタイプをサポートしています。

- 論理コンポジット - コンポーネントのグループを表します:機器、PDU、ファンシエルフ
- デバイス:機器およびそれに含まれるコンポーネント
- 論理接続 - ネットワークインターフェイス間の接続
- ポート関係 - ネットワークインターフェイス間の関係 (物理、論理、遅延)
- 論理接続関係:論理接続間の関係
- トポロジ - ネットワークトポロジ

クラス構造

```
let TsomSchemaValidator = Class.create();
TsomSchemaValidator.prototype = {
  initialize: function() {
    this.schemas = new TsomGenericSchema();
  },
};
```

```

isValidJson: function(payload) {
  // Validation logic that determines if the JSON structure is valid
  // Returns boolean (true/false)
},
checkJsonValidation: function(payload) {
  // Validation logic that determines if the JSON structure is valid
  // Returns a JSON object containing errors (if exist)
},
type: 'TsomSchemaValidator'
};

```

ステップ

1. スキーマ検証ツールのインスタンス化javascriptCopyEdit

```
var TsomSchemaValidator = new sn_tsom_core.TsomSchemaValidator();
```

2. ブール検証チェックを実行

```

if (!TsomSchemaValidator.isValidJson(target_json)) {
  gs.error('Invalid JSON: ' + JSON.stringify(target_json));
  return;
}

```

3. 詳細な検証チェックを実行

```

let result = TsomSchemaValidator.checkJsonValidation(target_json);
if (!result.valid) {
  gs.error('Invalid JSON: ' + JSON.stringify(result, null, 2));
  return;
}

```

出力例

```

Example Output
{
  "schemaName": "devices",
  "errors": [
    {
      "message": "Missing required property: model_name",
      "params": { "key": "model_name" },
      "code": 302,
      "dataPath": "/devices/0/ports/0",
      "schemaPath": "/properties/devices/items/properties/ports/items/required/4"
    }
  ],
  "valid": false
}

```

関連トピック

[コネクタでの Telecom Discovery Builder フレームワーク ETL の構成](#)

[Telecom Discovery Builder フレームワーク](#)

電気通信不一致監査の実行

電気通信不一致監査は、CMDB コンプライアンスフレームワークを使用して、電気通信インベントリ全体の構成アイテム (CI) と関係の整合性を検証します。

始める前に

必要なロール：admin

このタスクについて

電気通信不一致監査は CMDB コンプライアンスフレームワークの一部であり、ServiceNow ディスカバリーまたはサービスグラフコネクタ (Nokia Altiplano など) によって検出された CI 関係および属性の不一致を検出して対処するための段階的な監査をサポートしています。

監査は手動で実行することも、定期的に行うように設定することもできます。手動実行中に、既存のフィルターから選択して監査スコープを特定の CI に制限することができます。これはテストやターゲット調整に役立ちます。

電気通信不一致監査は、次の 2 つのフェーズで機能します。

1. 最初のコンプライアンス実行: 次のデフォルト基準を使用して、CI 関係と選択された CI テーブルを検証します。
 - サポートされている CI クラス: スロット、サブスロット、カード、インターフェイス、ネットワーク機器 (すべての拡張テーブルを含む)。
 - ソース: CI がディスカバリーを介して検出されました (例: discovery_source = SG-TSOM-Altiplano)。
 - 関係タイプ: Contains::Contained By (sn_tsom_core.audit.relationship_types プロパティでカスタマイズ可能)。
2. 後続のコンプライアンス実行: 初期チェックに加えて、関係性または関連 CI の更新済みタイムスタンプが監査レコードの前の実行日より新しいかどうかを評価します。

- i** 注: 監査が失敗するたびに、手動または自動の修正のための後続タスクが作成され、電気通信 CMDB データがネットワークの状態に合わせて調整された状態に保たれていることを確認します。

次のスクリーンショットは、電気通信不一致監査を理解するのに役立ちま

The screenshot shows the configuration page for the 'Telecom Discrepancy Audit' task in ServiceNow. Key elements include:

- Name:** Telecom Discrepancy Audit
- Filter:** Filter for TSOM Discrepancy Audit - 1
- Table:** Configuration Item [cmdb_ci]
- Assignment type:** User Field
- Assign to:** Assigned to
- Assign to empty:** Create Unassigned Task
- Audit type:** Scripted
- Short description:** Telecom CMDB Discrepancy Audit
- Task description:** Remedy the discrepancies listed in the Audit Results below
- Run this script:** Turn on ECMAScript 2021 (ES12) mode

```

1 /* global gs current SNC CertificationProcessing TsomDiscrepancyAudit GlideDateTime */
2
3 var certProcessor;
4 if (gs.getCurrentScopeName().includes("global"))
5   certProcessor = new SNC.CertificationProcessing();
6 else
7   certProcessor = new CertificationProcessing();
8
9 //
10 let tsomDiscrepancyAudit = new TsomDiscrepancyAudit();
11 tsomDiscrepancyAudit.printInfoLog("Audit Started.");
12 let auditStartDateTime = new GlideDateTime();
13 auditStartDateTime.addSeconds(-1); // change the audit start time to 1 second before the current time, so that
14 // relationships added at the current time will be picked up later.
15 let lastAuditRunDateTime = current.last_run_date.toString();
16 let auditId = current.sys_id;
17 tsomDiscrepancyAudit.initAuditId(auditId, certProcessor);
  
```

す。

手順

1. 移動先 **すべて > コンプライアンス > 監査**
2. 電気通信不一致監査 (cert_auditテーブル) レコードを開きます。
3. オプション: フィルターを選択します。
 - 構成中に複数の監査フィルターが定義されている場合は、監査を実行する前に 1 つを選択できます。
 - フィルターを使用すると、ディスカバリーソース、CI クラス、特定の CI 属性などの基準に基づいて監査範囲を制限できます。
 - この手順は、レコードのサブセットのトラブルシューティングや特定のディスカバリー結果の検証を行う場合に特に役立ちます。
4. 監査を実行するには、次のいずれかを実行します。
 - [監査を実行] をクリックして、手動実行をトリガーします。
 - 定期的なジョブを設定して、スケジュールに基づいて実行する監査を構成します。
監査では、CMDB コンプライアンスを使用して、通信インベントリ全体の関係性と CI の完全性を検証します。

結果

監査結果のレビュー

- 監査では、関係の欠落や属性の不整合などのコンプライアンス違反を特定します。
- 失敗した監査レコードごとに、後続タスクが自動的に生成されます。
- タスクには、CI の更新や廃止などの推奨または自動の修復アクションが含まれます。

ユースケースの例:構成済みのフィルターは、さまざまなディスカバリーソースに使用できます。たとえば、Nokia Altiplano などです。監査を手動で実行する場合は、適切なフィルターを選択して、その特定のソースによって検出された CI のみを検証し、ターゲットを絞った効率的な監査を確認できます。

関連トピック

[監査のフィルターの構成](#)

[電気通信不一致の監査と修正の例](#)

電気通信不一致の監査と修正の例

次の例は、シナリオでの監査の仕組みを示しています。

始める前に

必要なロール: admin

このタスクについて

カード (Card04) が最初に Slot04 で発見されました。その後、Card04 は物理ネットワークで Card05 に置き換えられましたが、CMDB では Slot04 に Card04 が表示されます。新しいディスカバリーが実行されると、Card05 が検出されて同じスロットに追加され、CMDB でデータの競合が発生します。

監査動作:

- Telecom Discrepancy Audit はこの不整合を検出し、失敗した監査レコード (AUDR0001283 など) を作成します。
- 後続タスクが自動的に作成され (例:TASK0020215)、不一致の詳細な説明が表示されます。

タスクの説明の例 (リレーションシップの数が正しくない):

Card04 が最後に発見されたのは 2.5 日以上前です。

次の CI 間の関係

CI	モデル
スロット04	デモ 20532Tree
Card04	Nokia 7360 FANT-Fカードモジュール
Card05	<モデルが特定されていません>

次のスナップショットは、不一致と修正プロセスを理解するのに役立ちます。

Audit Results (45)

Created	Document	State	Column name	Desired value	Discrepancy value	Follow on task	Number	Template	Threshold	Stability
2024-11-26 04:34:33	Configuration Item: Card40	Failed	Most recent discovered date	Is not within threshold 2024-11-25 00:59:44	2024-11-22 00:59:45	TASK0020215	AUDR0001283	(empty)	Pending	Pending

Follow On Tasks (39)

Name	Table	Comments	Form action	List action	Active	Order	Condition	Updated
Follow On Task	Follow On Task [ort_follow_on_task]		true	false	true		1,000 new TaskRemediation.getRemediateConditi...	2024-11-21 08:27:48

Work notes • 2024-11-26 04:37:39

SM Service-now: Yaron Nechushtan [maint...]

TSOM CI Decommission
=====

Card: Card40

LCS set to 'End of life'
LCSS set to 'Retired'

Relationships removed:

Slot40 ==> Card40

Field Changes • 2024-11-26 04:34:33

SA System Administrator

Description Card40 was last discovered more than 2.5 days ago.
Relationships between the following CIs:
CI: Slot40 (8b2beb4247ceda10f04f83ac416d4398), Model: DEMO 20532Tree (1ba577524c1b3110f8772646dabeb9bb),
CI: Card40 (0b2beb4247ceda10f04f83ac416d4399), Model: Nokia 7360 FANT-F CARD MODULE (3af9617de5928110f877657a333391e0),
CI: Card41 (832beb4247ceda10f04f83ac416d439a), Model: ()

Field Changes • 2024-11-26 04:34:32

SA System Administrator

Active true

Audit Service Operation CMDB Compliance Audit

Configuration item Card40

Impact 3 - Low

Number TASK0020215

Opened by System Administrator

Priority 4 - Low

Short description Most recent discovery date not within configured threshold.

State Open

手順

- ナビゲート **すべて** > **コンプライアンス** > **監査** > **電気通信不一致の監査** > **監査** を実行。
後続タスクは、失敗した監査レコード (TASK0020215 など) ごとに自動的に作成されます。
- [TASK0020215]** を選択し、不一致の説明を確認します。

注: これは、「不適切な数のリレーションシップ」シナリオ用に作成された TASK0020215 説明の例です。他のシナリオや環境では、説明が異なる場合があります。

後続タスクには、不一致の詳細な説明が含まれています。説明からわかるように、Card04 CI に不一致があります。

3. 修復するには、次の手順を実行します。

- a. ナビゲート **すべて** > システム定義 > **UI** アクション。
- b. [修正] を選択します。
[修正] ボタンは、TSOM CI 廃止 サブフローをトリガーする UI アクションです。このサブフロー：
 - 古い Card04 CI を廃止します。
 - 誤った Slot04 → Card04 関係を削除します。
 - CMDB レコードをネットワークステータスと同期します。

結果

修正後:

- 後続タスク (TASK0020215) は、解決手順を要約した作業メモで更新されます。
- Card04 CI は廃止としてマークされ、誤った関係は削除されます。
- これで、CMDB はネットワークから検出された最新のステータスに調整されます。

i 注: 修正のカスタマイズ:電気通信不一致監査には、修正サブフローの例が付属しています。運用要件に合わせて、フローデザイナーを使用してカスタムサブフローを作成して添付できません。

関連トピック

[電気通信不一致の識別と調整](#)

[不一致の識別 - 不一致の種類](#)

[Telecom Discrepancy Identification & Reconciliation](#) とともにインストールされるシステムコンポーネント

[電気通信不一致の識別と調整のアクティブ化](#)

[UI アクションの定義](#)

属性値不一致のレポートを生成

CMDB 360 を使用して、異なるディスカバリーソース間、またはディスカバリーソースと CMDB ベースライン間の属性値の不一致を強調するレポートを生成します。

始める前に

必要なロール：admin

このタスクについて

属性値不一致レポートは、同じ構成アイテム (CI) を更新する複数のソースからの競合するデータを特定するのに役立ちます。これにより、CMDB 全体のデータの品質と整合性が向上します。

手順

1. 移動先 **すべて** > **CMDB** ワークスペース > **CMDB 360**。
2. [クエリの作成] を選択します。
3. クエリの種類ウィンドウで、[属性値の比較] を選択します。
4. クエリパラメーターを定義します。

- a. 分析する **CI** クラス (cmdb_ci_computer など) を選択します。
 - b. フィルターを適用してスコープを絞り込みます (たとえば、ディスカバリーソースに「TSOM」が含まれています)。
 - c. 属性を選択して、さまざまなディスカバリーソース間で値を比較します。
 - d. 比較するソース (ディスカバリー、SCCM など) を選択します。
 - e. オプション: [**CMDB と比較**] を選択して、ディスカバリーソースデータを現在の CMDB ベースラインと比較します。
 - f. [保存] をクリックし、クエリの名前を指定して、[実行] をクリックして結果を生成します。
5. オプション: [スケジュール] をクリックして、指定した間隔で実行するようにクエリを設定します。

結果

レポートには、ソース間の属性値の比較が表示されます。

- ソース 1:現在の CMDB レコード。
- ソース 1 の値:最新のディスカバリーソースによって更新された値。
- ソース 2:以前のディスカバリーソース。
- ソース 2 の値:以前のディスカバリーソースから提供された値。

CI が複数のソースによって更新された場合、それぞれの不一致は個別のレコードとしてリストされます。

関連トピック

[CMDB 360 での属性値の不一致の構成](#)

[調整ルールを使用した CI 属性の更新の制御](#)

通信事業サービスオペレーション管理の参照

いくつかのタイプのコンポーネントが、通信事業サービスオペレーション管理 アプリケーションとプラグインとともにインストールされます。

通信事業 **API** 通知とともにインストールされるシステムコンポーネント

アドミニストレーターは、ユーザーロールを割り当てて、API 通知データベーステーブルへのアクセスを許可できます。トピック [sn_api_notif_mgmt_topic] およびトピックサブスクリプション [sn_api_notif_mgmt_subscription] テーブルの次の標準ロールは、ServiceNow システムに含まれています。

通信 **API** 通知ロール

ロール	説明
sn_api_notif_mgmt.topic_subscription_viewer	トピックおよびトピックサブスクリプションテーブルへの読み取りアクセス権を有効にするロール。
sn_api_notif_mgmt.topic_creator	トピックテーブルに対する作成、読み取り、および編集アクセス権を有効にするロール。
sn_api_notif_mgmt.subscription_creator	トピックサブスクリプションテーブルの作成および読み取りアクセス権を有効にするロール。

通信 API 通知ルール (続く)

ルール	説明
sn_api_notif_mgmt.subscription_admin	<p>次の権限で有効になるルール:</p> <ul style="list-style-type: none"> トピックおよびトピックサブスクリプションテーブルに対する作成および読み取りアクセス。 登録ステータスを変更して、トピックのサブスクリプションを登録解除します。

関連トピック

[電気通信 API 通知を介した外部イベント管理](#)

[電気通信 API 通知の構成](#)

Nokia Altiplano とともにインストールされるシステムコンポーネント

システムプロパティは、ディスカバリーオプションやパフォーマンス設定など、コネクタの動作方法を制御します。

Nokia Altiplano とともにインストールされるプロパティ

プロパティ	説明
<code>sn_sgc_altiplano.enable_onu_discovery</code>	<p>ONU デバイスディスカバリー、および OLT と ONU 間の論理接続を有効または無効にします。</p> <ul style="list-style-type: none"> デフォルト値: True 場所: All>Service Graph Connectors>Nokia Altiplano>Properties または System Properties [sys_properties] テーブル 「*Altiplano」という名前でフィルタリング
<code>sn_sgc_altiplano.devices_list_batch_size</code>	<p>Altiplano REST API 呼び出しのバッチサイズを制御します。</p> <ul style="list-style-type: none"> デフォルト値: 1000 場所: All>Service Graph Connectors>Nokia Altiplano>Properties または System Properties [sys_properties] テーブル 「*Altiplano」という名前でフィルタリング

Nokia Altiplano とともにインストールされるプロパティ (続く)

プロパティ	説明
<code>sn_sgc_altiplano.parallel_number_of_data_source_jobs</code>	<p>Altiplano データを収集するための並列ジョブの数 ([並列ロードを有効にする] 設定が必要)。</p> <ul style="list-style-type: none"> デフォルト値 : 2 場所: All>Service Graph Connectors>Nokia Altiplano>Properties または System Properties [sys_properties] テーブル 「*Altiplano」という名前でフィルタリング
<code>sn_sgc_altiplano.onu_ci_class</code>	<p>ONU を ONU または ONT CI クラスとして格納するかどうかを定義します。</p> <ul style="list-style-type: none"> デフォルト値:ONU 場所: All>Service Graph Connectors>Nokia Altiplano>Properties または System Properties [sys_properties] テーブル 「*Altiplano」という名前でフィルタリング

自動翻訳

関連トピック

- [Nokia Altiplano 経由の電気通信ディスカバリー](#)
- [Nokia Altiplano サービスグラフコネクタの構成](#)

Telecom Discrepancy Identification & Reconciliation とともにインストールされるシステムコンポーネント

システムプロパティは TSOM ヴィジビリティプラグイン (sn_tsom_core) の一部であり、電気通信不一致識別および調整ログ (TSOM CMDB 監査) を制御します。TSOM ヴィジビリティプラグインは、TSOM ヴィジビリティアプリケーションを実現する役割を果たし、テレコムディスカバリーおよびテレコム不一致識別および調整ソリューション全体で共有されるロジックを含みます。

電気通信不一致識別および調整システムプロパティ (CMDB 監査に影響)

プロパティ名	推奨/デフォルト値	説明
<code>sn_tsom_core.audit.log.level</code>	Telecom Discrepancy Audits' のログ。	デバッグ、情報、デフォルト値 : in

電気通信不一致識別および調整システムプロパティ (CMDB 監査に影響) (続く)

プロパティ名	推奨/デフォルト値	説明
sn_tsom_core.audit.relationship_types	電気通信不一致監査によって処理される関係タイプの表示名。	リレーションシップ デフォルト値: Cor
sn_tsom_core.audit.discovered_date.diff.threshold.in.days	最新の検出日のしきい値 (日数)。電気通信不一致の監査に使用されます。	文字列 デフォルト値: 2.5
sn_tsom_core.audit.suppress_CI_Model_missing_discrepancy_task_creation	テーブルの不一致タスクがない CI モデルの作成を抑制します。電気通信不一致の監査に使用されます。	cmdb_ci_ni_telco
sn_tsom_core.audit.equipment_tables	電気通信不一致監査によって処理される機器テーブル。	機器テーブル名 デフォルト値: cm
sn_tsom_core.audit.interface_card_tables	電気通信不一致監査によって処理されるインターフェイスカードテーブル。	カードテーブル名 デフォルト値: cm
sn_tsom_core.audit.interface_tables	電気通信不一致監査によって処理されるネットワークインターフェイステーブル。	インターフェイス デフォルト値: cm
sn_tsom_core.audit.slot_tables	電気通信不一致監査によって処理されるスロットテーブル名。	スロットテーブル デフォルト値: cm
sn_tsom_core.audit.subslot_tables	電気通信不一致監査によって処理されるサブスロットテーブル名。	サブスロットテー デフォルト値は c

電気通信不一致の識別および調整の監査

監査名	説明
電気通信不一致の監査	<p>cmdb_rel_ciテーブルのレコードを監査します。Nokia Altiplano または NSP のドごとに、以下に基づいて親と子の CI が検証されます。 - 最新の検出日 - CI - Core がインストールされている場合)。</p> <p>i 注: すべての不一致監査では、機器の CI に適用されるフィルタリングメカニクス。フォルトでは、フィルター条件は [discovery_source CONTAINS TSOM] 。</p>
電気通信論理接続の不整合監査	<p>cmdb_ci_ni_logical_pathテーブルの監査レコード。Nokia Altiplano または NSP のドごとに、次の項目がチェックされます。 - 空のエンドポイント (ポート A または B) は、1 つの一意的論理接続にのみ接続する必要があります。</p>
電気通信ネットワークポロジの不整合監査	<p>cmdb_ci_network_topologyテーブルのレコードを監査します。Nokia Altiplano または NSP のドごとに、以下の項目が検証されます。 - ネットワークポロジ CI ごとに、以下が検証されます。 - 機器との少なくとも 1 つの Members:Member Of 関係 - 論理接続との少なくとも 1 つの Members:Member Of 関係。</p>

関連トピック

- [電気通信不一致の識別と調整](#)
- [電気通信不一致の識別と調整のアクティブ化](#)
- [電気通信不一致監査の実行](#)