テンプレート、リリース
12.2.1

目次

新規情報および変更情報 1
テンプレート （Templates） 2
新規テンプレートの作成 4
テンプレートの編集 5
テンプレートのインポート 7
Git からテンプレートのインポート 7
テンプレートのエクスポート 9
Git からテンプレートのエクスポート 9
テンプレート構造 10
テンプレートの形式 10
テンプレート変数 15
可変メタ プロパティ 20
可変注釈 27
テンプレートの内容 31
高度な機能 34
レポート テンプレート 36
テンプレートの使用方法 39
ポリシーテンプレート 39
ファブリックのテンプレート 41
プロファイル テンプレート 41
補足事項 41
使用中テンプレートのコンテンツの変更 42
著作権 43

新規情報および変更情報

次の表は、この最新リリースまでの主な変更点の概要を示したものです。ただし、今リリースまでの変更点または、新機能の一部は表に記載されていません。

リリース バージョン	特長	説明
NDFC リリース 12.2.1	Git リポジトリとの間の NDFC 非デフォルト テンプレートのインポートまたはエクスポートのサポート	この機能を使用すると、デフォルト以外の NDFC テンプレートを Git リポジトリとの間でインポートまたはエクスポートして、NDFC のデフォルト以外のテンプレートを同期できます。詳細については、「Git からテンプレートのインポート」 および「Git からテンプレートのエクスポート」を参照してください。

テンプレート （Templates）

Cisco Nexus Dashboard Fabric Controller を使用して、異なる Cisco Nexus、IOS XE、 IOS XR、および Cisco MDS プラットフォームで構成されているテンプレートを追加、編集、または削除できます。Cisco Nexus Dashboard Fabric Controller で構成されているテンプレートごとに、次のパラメータが表示されます。テンプレートは JavaScript をサポートします。テンプレートの JavaScript 関数を使用して、テンプレートの構文で算術演算と文字列操作を実行できます。

1. 管理（**Manage**）] > [テンプレート（**Templates**）] を選択します。

表 1. テンプレート テーブルのフィールドと説明

フィールド	説明
名前（**Name**）	テンプレート名を指定します。
サポートされるプラットフォーム	テンプレートがサポートするプラットフォームを指定します。
タイプ	テンプレート タイプを指定します。
サブタイプ	テンプレート サブ タイプを指定します。
変更日	テンプレート変更の日時を指定します。
タグ（**Tags**）	テンプレートがファブリックまたはデバイスにタグ付けされているかどうかを指定します。
説明	テンプレートの説明を指定します。
参照カウント	テンプレートが使用される回数を指定します。

2. テーブル ヘッダーをクリックすると、そのパラメータのアルファベット順にエントリがソートされます。

エラーのあるテンプレートは、[テンプレート（**Templates**）] ページに表示されません。エラーのあるテンプレートは
インポートできません。エラーのあるテンプレート を インポートするには、エラーを修正してからテンプレートをインポートします。

次の表では、[アクション（**Actions**）] メニューのドロップダウン リストにあるアクション アイテムについて説明します。このリストは、
[テンプレート（**Templates**）] ページに表示されます。

表 2. テンプレートのアクションと説明

Actions	説明
新しいテンプレートの作成	新しいテンプレートを作成できるようにします。詳細については、「新しいテンプレートの作成（Creating a New Template）」を参照してください。
テンプレートのプロパティの編集	テンプレートのプロパティを編集できるようにします。一度に編集できるテンプレートは 1 つだけです。詳細については、「テンプレートの 編集（Editing a Template）」を参照してください。
テンプレートの内容の編集	テンプレートの内容を編集できるようにします。一度に編集できるテンプレートは 1 つだけです。詳細については、「テンプレートの 編集（Editing a Template）」を参照してください。

Actions	説明
テンプレートの複数	選択したテンプレートを別の名前で複製できるようにします。必要に応じて、テンプレートを編集できます。一度に複製できるテンプレートは 1 つだけです。 テンプレートを複製するには、複製するテンプレートの横にあるチェックボックスをオンにし、**[アクション（Actions）] > [テンプレートの複製（Duplicate template）]** を選択します。**[テンプレートの複製（Duplicate template）]** ページが表示されます。複製されるテンプレートの名前を指定します。複製されたテンプレートの詳細については、テンプレートの編集 を参照してください。
テンプレートの削除	テンプレートを削除できるようにします。1 つのインスタンスで複数のテンプレートを削除できます。ユーザ定義テンプレートを削除できます。ただし、事前定義されたテンプレートは削除できません。 テンプレートを削除するには、削除するテンプレートの横にあるチェックボックスをオンにし、**[テンプレートの削除（Delete template）]** を選択します。警告メッセージが表示されます。テンプレートを削除する場合は、**[確認（Confirm）]** をクリックします。削除しない場合は、**[キャンセル（Cancel）]** をクリックします。テンプレートが使用中であるか、出荷テンプレートである場合は、削除できず、エラー メッセージが表示されます。 複数のテンプレートを選択して、同時に削除します。 テンプレートを完全に削除するには、ローカル ディレクトリ：Cisco Systems\dcn\ndfc\data\templates\ にあるテンプレートを削除します。
インポート	ローカル ディレクトリからテンプレートを 1 つずつインポートできます。詳細については、「テンプレートのインポート（Import a Template）」を参照してください
Zip としてインポート	.zip 形式でバンドルされた複数のテンプレートを含む .zip ファイルをインポートできます。.zip のすべての テンプレートが抽出され、個別のテンプレートとしてテーブルに一覧表気します テンプレートを使用してデバイスを設定します。詳細については、「新しいテンプレートのインポート（Import a Template）」を参照してください。
ギットからのインポート	デフォルト以外の NDFC テンプレートファイルを Git リポジトリから NDFC テンプレートにインポートできます。詳細については、「Git からテンプレートのインポート（Import a Template）」を参照してください

Actions	説明
エクスポート	ローカル ディレクトリの場所にテンプレート設定をエクスポートできます。一度に複数のテンプレートをエクスポートできます。 テンプレートをエクスポートするには、テンプレートの横にあるチェックボックスを使用して必要なテンプレートを選択し、**[**アクション（**Actions**）**]** > **[**エクスポート（**Export**）**]**を選択します。テンプレート ファイルを保存するローカル システム ディレクトリの場所を選択します。**[Save**（保存）**]** をクリックします。テンプレート ファイルは、単一のテンプレート ファイルまたは圧縮 .zip ファイルとしてローカル ディレクトリにエクスポートされます（複数のテンプレートを選択した場合）。 複数の テンプレートを 1 つの .zip ファイルとしてエクスポート。 できます。
ギットへのエクスポート	デフォルト以外の NDFCテンプレートを Git リポジトリにエクスポートできます。詳細については、「 Git へのテンプレートのエクスポート」を参照してください。

network-operator ロールを持つテンプレートのみを表示できます。このロールを持つテンプレートを作成、
編集、または保存できません。ただし、**network-stager** ロールを使用してテンプレートを作成または編集できます。

新規テンプレートの作成

Cisco Nexus Dashboard Fabric Controller UI ナビゲーション

- 管理（**Manage**）**] > [**テンプレート（**Templates**）**]** を選択します。

Cisco Nexus Dashboard Fabric Controller Web UI からユーザー定義のテンプレートを作成し、ジョブをスケジュールするには、次の手順を実行します。

1. **[**テンプレート（**Templates**）**]** ウィンドウで、**[**アクション（**Actions**）**]** ドロップダウン リストから **[**新規テンプレートの作成（**Create new template**）**]** を選択します。**[**テンプレートの作成（**Create Template**）**]** ウィンドウが表示されます。
2. ウィンドウの **[**テンプレート プロパティ（**Template Properties**）**]** ページで、テンプレート名、説明、タグを指定し、新しいテンプレートのサポート対象プラットフォームを選択します。次に、ドロップダウン リストからテンプレート タイプとサブテンプレート タイプを選択します。ドロップダウン リストからテンプレートのコンテンツ タイプを選択します。

   基本テンプレートは CLI テンプレートです。

3. **[**次へ（**Next**）**]** をクリックしてテンプレートの編集を続行するか、**[**キャンセル（**Cancel**）**]** をクリックして変更を破棄します。

編集したテンプレートのプロパティは、**[**テンプレートの編集（**Edit Template**）**]** ウィンドウの **[**テンプレート コンテンツ（**Template Content**）**]** ページに表示されます。構成テンプレートの構造については、「*テンプレートの構造*」の項を参照してください。

4. **[**検証（**Validate**）**]** をクリックして、テンプレートの構文を検証します。

   警告のみがある場合は、テンプレートの保存を続行できます。 ただし、エラーが発生した場合は、続行する前にテンプレートを編集してエラーを修正する必要があります。[開始行（Start Line）] 列の下の行番号をクリックして、テンプレートの内容でエラーを見つけます。テンプレート名がないテンプレートを検証すると、エラーが発生します。

5. **[**ヘルプ（**Help**）**]** をクリックして、右側の **[**エディタのヘルプ（**Editor Help**）**]** ペインを開きます。

   このウィンドウには、テンプレートの作成に使用された形式、変数、コンテンツ、およびデータ型に関する詳細情報が表示されます。**[**エディタのヘルプ（**Editor Help**）**]** ペインを閉じます。

6. リンクが表示されたら、エラーおよび警告をクリックします。エラーまたは警告がない場合、リンクは使用できません。エラーまたは警告が表示されている場合にリンクをクリックすると、右側に**[**エラーおよび警告（**Errors&Warnings**）**]** ペインが表示され、エラーと警告が表示されます。**[**エラーおよび警告（**Errors&Warnings**）**]** ペインを閉じます。
7. テンプレート コンテンツを作成するには、必要なテーマ、キー バインディング、およびフォント サイズをドロップダウン リストから選択します。
8. **[**完了（**Finish**）**]** をクリックしてテンプレートの編集を完了し、**[**キャンセル（**Cancel**）**]** をクリックして変更を破棄し、[前へ（Previous）] をクリックして
   [テンプレート プロパティ（**Template Properties**）**]** ページに移動します。

   テンプレートが作成されたことを示すメッセージのページが表示されます。このページには、テンプレート名、タイプ、サブタイプ、およびプラットフォームも表示されます。**[**別のテンプレートの作成（**Create another template**）**]** をクリックしてもう 1 つのテンプレートを作成するか、**[<template name>** テンプレートの編集（**Edit <template name> template**）**]** をクリックして編集したばかりのテンプレートを編集します。

9. **[**テンプレートの編集（**Edit Template**）**]** ウィンドウを閉じるか、**[**テンプレート ライブラリに戻る（**Back to template library**）**]** をクリックして **[**テンプレート（**Templates**）**]** に戻ります。
   閉じます。

テンプレートの編集

Cisco Nexus Dashboard Fabric Controller では、ユーザー定義のテンプレートを編集できます。ただし、定義済みのテンプレートおよびすでに公開されているテンプレートは編集できません。

[テンプレートの編集（**Edit Template**）**]** ウィンドウを使用して、最初にテンプレートのプロパティを編集し、次にテンプレートの内容を編集します。さらに、**[**テンプレート プロパティの編集（**Edit Template Properties**）**]** アクションを使用してテンプレート プロパティのみを編集するか、**[**テンプレート コンテンツの編集（**Edit template content**）**]** アクションを使用してテンプレート コンテンツのみを編集できます。つまり、あるインスタンスでテンプレートのプロパティを編集してから、別のインスタンスでテンプレートの内容を編集できます。このウィンドウを使用して、テンプレートのプロパティとコンテンツを表示することもできます。

テンプレートのプロパティを編集し、テンプレートの内容を編集するには、次の手順を実行します。

1. Cisco Nexus Dashboard Fabric Controller 内で、**[**管理（**Manage**）**] > [**テンプレート（**Templates**）**]**

を選択します。

2. **[**テンプレート（**Templates**）**]** ウィンドウで、テンプレートを選択します。**[**アクション（**Actions**）**]** ドロップダウン リストから、**[**テンプレート プロパティの編集（**Edit Template Properties**）**]** を選択します。

   [テンプレートの編集（**Edit Template**）**]** ウィンドウが表示されます。

3. ウィンドウの **[**テンプレート プロパティ（**Template Properties**）**]** ページに、テンプレートの名前、その説明、サポートされるプラットフォーム、タグ、およびコンテンツ タイプが表示されます。テンプレートの説明とタグを編集できます。サポートされているプラットフォームを編集するには、選択したチェックボックスをオフにして他のスイッチを選択します。次に、ドロップダウン リストからテンプレート タイプとサブテンプレート タイプを選択します。

4. **[**次へ（**Next**）**]** をクリックしてテンプレートの編集を続行するか、**[**キャンセル（**Cancel**）**]** をクリックして変更を破棄します。
   編集したテンプレートのプロパティは、**[**テンプレートの編集（**Edit Template**）**]** ウィンドウの **[**テンプレート コンテンツ（**Template Content**）**]** ページに表示されます。
   ウィンドウに入力できます。

5. **[**検証（**Validate**）**]** をクリックして、テンプレートの構文を検証します。

   警告のみがある場合は、テンプレートの保存を続行できます。ただし、エラーが発生した場合は、続行する前にテンプレートを編集してエラーを
   修正する必要があります。[開始行（Start Line）] 列の下の行番号をクリックして、
   テンプレートのコンテンツのエラーを検索します。テンプレート名がないテンプレートを検証すると、
   エラーが発生します。

6. **[**ヘルプ（**Help**）**]** をクリックして、右側の **[**エディタのヘルプ（**Editor Help**）**]** ペインを開きます。

   このウィンドウには、テンプレートの作成に使用された形式、変数、コンテンツ、およびデータ型に関する詳細情報が表示されます。**[**エディタのヘルプ（**Editor Help**）**]** ペインを閉じます。

7. リンクが表示されたら、エラーおよび警告をクリックします。エラーまたは警告がない場合、リンクは使用できません。エラーまたは警告が表示されている場合にリンクをクリックすると、右側に**[**エラーおよび警告（**Errors&Warnings**）**]** ペインが表示され、エラーと警告が表示されます。**[**エラーおよび警告（**Errors&Warnings**）**]** ペインを閉じます。

8. テンプレート コンテンツを作成するには、必要なテーマ、キー バインディング、およびフォント サイズをドロップダウン リストから選択します。

9. **[**完了（**Finish**）**]** をクリックしてテンプレートの編集を完了し、**[**キャンセル（**Cancel**）**]** をクリックして変更を破棄し、[前へ（Previous）] をクリックして
   [テンプレート プロパティ（**Template Properties**）**]** ページに移動します。

   テンプレートが保存されたことを示すメッセージが表示されたページが表示されます。このページには、テンプレート名、タイプ、サブタイプ、およびプラットフォームも表示されます。**[**別のテンプレートの作成（**Create another template**）**]** をクリックしてもう 1 つのテンプレートを作成するか、**[<template name>** テンプレートの編集（**Edit <template name> template**）**]** をクリックして編集したばかりのテンプレートを編集します。

10. **[**テンプレートの編集（**Edit Template**）**]** ウィンドウを閉じるか、**[**テンプレート ライブラリに戻る（**Back to template library**）**]** をクリックして **[**テンプレート（**Templates**）**]** に戻ります。
    閉じます。

テンプレートのインポート

1. **[管理（Manage）] > [テンプレート（Templates）]**の順に選択します。
2. **[インポート（Import）]** を **[アクション（Actions）]** ドロップダウン リストから選択します。**[テンプレートのインポート（Import Template）]** ダイアログボックスが表示されます。
3. コンピュータに保存したテンプレートを参照し、**[テンプレートのインポート（ Import Template）]** ダイアログ ボックス
にドラッグします。
4. **[OK]** をクリックしてテンプレートをインポートするか、**[キャンセル（Cancel）]** をクリックしてテンプレートを破棄します。zip 形式のテンプレートをインポートする場合も、同じ手順に従います。

   テンプレートの「\n」は、テンプレートをインポートおよび編集すると改行文字と見なされますが、ZIP ファイルとしてインポートされると正常に機能します。

   ZIP テンプレート ファイルをインポートした後、成功またはエラー メッセージのどちらかが受信します。[OK] をクリックします。
5. 必要に応じて、テンプレート パラメータとコンテンツを編集できます。詳細については、「新しいテンプレートの編集（Editing a Template）」を参照してください。

   圧縮されたテンプレート ファイルをインポートすると、**[テンプレートの編集（Edit Template）]** ページが表示されないことがあります。必要な場合、[テンプレートの編集（Edit template）] プロパティまたは [テンプレート コンテンツの編集（Edit template content）] オプションのどちらかを使用して、テンプレート パラメータおよびコンテンツを編集できます。
6. テンプレートのプロパティまたはコンテンツを編集しない場合は、**[次へ（Next）]**、**[完了（Finish）]**、**[テンプレート ライブラリに戻る（Back to template library）]** の順にクリックして、**[テンプレート（Templates）]** ページに戻ります。

Git からテンプレートのインポート

デフォルト以外の NDFC テンプレート ファイルを Git リポジトリから NDFC テンプレートにインポートできます。

NDFC テンプレート ファイルのみが Git からインポートされます。他のファイルはサポートされていません。

はじめる前に

1. Git リポジトリを作成します。
2. Git リポジトリから NDFC テンプレートに NDFC テンプレートをインポートするには、github.com で Gitトークンを生成します。

Cisco Nexus Dashboard を使用した **Git IP** ルートの構成

1. Cisco Nexus Dashboard GUI で、 **[管理（Admin）] > [**システム設定（**System Settings** ）**] > [**全般（**General**）**] > [**ルート（管理ネットワークルート）（**Routes (Management Network Routes)**）**]** の順に選択します。
2. **[**ルート（**Routes**）**]** タイルの **[**編集（**Edit**）**]** アイコンをクリックします。
3. **[**管理ネットワーク ルートの追加（**Add Management Network Routes**）**]** をクリックし、github.com IPルートを指定します。
4. **[**保存（**Save**）**]** をクリックします。
5. NDFC でテンプレート パラメータと格納ファイルを

   編集します。詳細については、「新しいテンプレー

   トの編集（Editing a Template）」を参照してくださ

   い。

Cisco Nexus Dashboard を使用したプロキシ サーバの構成

グローバル パブリック Git リポジトリにアクセスする必要がある場合は、Cisco Nexus Dashboard を使用してプロキシ サーバを構成します。

1. Cisco Nexus Dashboard GUI で、[管理（Admin）] > [システム設定（System Settings）] > [全般（General）] > [プロキシ構成 （Proxy Configuration）] の順に選択します。
2. **[**プロキシ構成（**Proxy Configuration**）**]**タイルの **[**編集（**Edit**）**]** アイコンをクリックします。
3. **[**サーバの追加（**Add Server**）**]** をクリックし、必要な情報を指定します。
4. **[**保存（**Save**）**]** をクリックします。

Git リポジトリから **NDFC** へのテンプレートのインポート

1. **[**管理（**Manage**）**] > [**テンプレート（**Templates**）**]**の順に選択します。
2. **[Git** からインポート（**Import From Git**）**]** を **[**アクション

   （**Actions**）**]** ドロップダウン リストから選択します。**[Git**

   からインポート（**Import From Git**）**]** ダイアログ ボックス

   が表示されます。

3. 必要な情報を入力します。

フィールド	説明
ギットリポジトリ	Git リポジトリのディレクトリの場所を指定します。
支店	Git ブランチの名前を指定します。
ギットトークン	Git トークンを指定します。

4. **[**取得（**Fetch**）**]** をクリックします。

インポート プロセスの前に、NDFC にインポートされるすべてのファイルがリストされた Git から **[**情報（**Information**）**]** ダイアログ ボックスが表示されます。

インポート プロセスの後、Git から NDFC にインポートされた NDFC テンプレートが表示されます。

テンプレートのエクスポート

1. **[**管理（**Manage**）**] > [**テンプレート（**Templates**）**]**の順に選択します。
2. **[**エクスポート（**Export**）**]** を **[**アクション（**Actions**）**]**

   ドロップダウン リストから選択します。NDFC は、テン

   プレート ファイルをローカル システムにエクスポート

   します。

Git からテンプレートのエクスポート

デフォルト以外の NDFC テンプレートを GIT にエクスポートできます。

はじめる前に

1. Git リポジトリを作成します。
2. デフォルト以外の NDFC テンプレートを GIT リポジトリにエクスポートするには、github.com で Git トークンを生成します。

Cisco Nexus Dashboardを使用した **Git IP** ルートの構成

1. Cisco Nexus Dashboard GUI で、 **[**管理（**Admin**）**] > [**システム設定（**System Settings** ）**] > [**全般（**General**）**] > [**ルート（管理ネットワークルート）（**Routes (Management Network Routes)**）**]** の順に選択します。
2. **[**ルート（**Routes**）**]** タイルの **[**編集（**Edit**）**]** アイコンをクリックします。
3. **[**管理ネットワーク ルートの追加（**Add Management Network Routes**）**]** をクリックし、github.com IPルートを指定します。
4. **[**保存（**Save**）**]** をクリックします。
5. NDFC でテンプレート パラメータと格納ファイルを

   編集します。詳細については、「新しいテンプレー

   トの編集（Editing a Template）」を参照してくださ

   い。

Cisco Nexus Dashboard を使用したプロキシ サーバの構成

グローバル パブリック Gitリポジトリにアクセスする必要がある場合は、Cisco Nexus Dashboardを使用してプロキシ サーバを構成します。

1. Cisco Nexusダッシュボード GUI で 、[管理（**Admin**）] > [システム設定（**System** 設定 ）] > [全般（**General**）] > [プロキシ設定 （**Proxy Configuration**）] の順に選択します。
2. **[**プロキシ構成（**Proxy Configuration**）]タイルの **[**編集（**Edit**）] アイコンをクリックします。
3. **[**サーバの追加（**Add Server**）] をクリックし、必要な情報を指定します。
4. [保存

（Save）] をク

リックします。

Git へのエクス

ポート

1. **[**管理（**Manage**）] > [テンプレート（**Templates**）]の順に選択します。
2. **[Git** にエクスポート（**Export To Git**）] を **[Actions]** ドロ

   ップダウン リストから選択します。**[Git** へのエクスポー

   ト（**Export To Git**）] ダイアログ ボックスが表示されま

   す。

3. 必要な情報を入力します。

フィールド	説明
ギットリポジトリ	Git リポジトリのディレクトリの場所を指定します。
支店	Git ブランチの名前を指定します。
ギットトークン	Git トークンを指定します。

4. **[**取得（**Fetch**）] をクリックして、Git リポジトリからすべてのディ

   レクトリを取得します。Git リポジトリに NDFC テンプレートが表

   示されます。

テンプレート構造

構成テンプレートの内容は、主に 4 つの部分で構成されます。 テンプレートのコンテンツの編集については、
[テンプレート コンテンツ（Template Content）] の横にある **[**ヘルプ（**Help**）] アイコンをクリックします。

テンプレートの形式

ここでは、テンプレートの基本情報について説明します。次の表に、使用可能なフィールドの詳細を示します。

プロパティ名	説明	有効な値	任意かどうか
名前（name）	テンプレートの名前	テキスト	いいえ
説明	テンプレート に関する簡単な説明	テキスト（Text）	はい
userDefined	ユーザがテンプレートを作成したかどうかを示します。ユーザが作成した場合、値は「true」です。	「true」または「false」	はい
プロパティ名	説明	有効な値	任意かどうか
supportedPlatforms	この設定テンプレートをサポートするデバイス プラットフォームのリスト。すべてのプラットフォームをサポートするには、「All」を指定します。	N1K 、 N3K 、 N3500 、 N4K 、 N5K、 N5500、 N5600、 N6K、 N7K、N9K、MDS、VDC、N9K-9000v、IOS-XE、IOS-XE、IOS-XR、その他、すべての Nexus スイッチのリストがカンマで区切られています。できます。	いいえ
templateType	使用するテンプレートのタイプを指定します。	・CLI ・POAP POAP オプションは、Cisco Nexus Dashboard Fabric ControllerLAN ファブリックの展開には適用されません。 ・ポリシー ・SHOW ・プロファイル ・ファブリック ・[抽象（ABSTRACT）] ・レポート	はい

プロパティ名	説明	有効な値	任意かどうか
templateSubType	テンプレートに関連付けられたサブ タイプを指定します。	• CLI ◦ なし • POAP ◦ なし ◦ VXLAN ◦ FABRICPATH ◦ VLAN ◦ PMN POAP オプションは、Cisco Nexus Dashboard Fabric ControllerLAN ファブリックの展開には適用されません。 • ポリシー ◦ VLAN ◦ interface-vlan ◦ INTERFACE_VPC ◦ INTERFACE_ETHERNET ◦ INTERFACE_BD ◦ INTERFACE_PORT_CHANNEL ◦ INTERFACE_FC ◦ INTERFACE_MGMT ◦ INTERFACE_LOOPBACK ◦ INTERFACE_NVE ◦ INTERFACE_VFC ◦ INTERFACE_SAN_PORT_CHANNEL ◦ DEVICE ◦ FEX ◦ INTRA_FABRIC_LINK ◦ INTER_FABRIC_LINK	

		◦ INTERFACE	
プロパティ名	説明	有効な値	任意かどうか
templateSubType（続き）	テンプレートに関連付けられたサブ タイプを指定します。	・SHOW ◦ VLAN ◦ interface-vlan ◦ INTERFACE_VPC ◦ INTERFACE_ETHERNET ◦ INTERFACE_BD ◦ INTERFACE_PORT_CHANNEL ◦ INTERFACE_FC ◦ INTERFACE_MGMT ◦ INTERFACE_LOOPBACK ◦ INTERFACE_NVE ◦ INTERFACE_VFC ◦ INTERFACE_SAN_PORT_CHANNEL ◦ DEVICE ◦ FEX ◦ INTRA_FABRIC_LINK ◦ INTER_FABRIC_LINK ◦ INTERFACE ・プロファイル ◦ VXLAN ・ファブリック ◦ 該当なし	

プロパティ名	説明	有効な値	任意かどうか
templateSubType（続き）	テンプレートに関連付けられたサブ タイプを指定します。	・[抽象（ABSTRACT）] ◦ VLAN ◦ interface-vlan ◦ INTERFACE_VPC ◦ INTERFACE_ETHERNET ◦ INTERFACE_BD ◦ INTERFACE_PORT_CHANNEL ◦ INTERFACE_FC ◦ INTERFACE_MGMT ◦ INTERFACE_LOOPBACK ◦ INTERFACE_NVE ◦ INTERFACE_VFC ◦ INTERFACE_SAN_PORT_CHANNEL ◦ DEVICE ◦ FEX ◦ INTRA_FABRIC_LINK ◦ INTER_FABRIC_LINK ◦ INTERFACE ・レポート ◦ アップグレード ◦ GENERIC	

プロパティ名	説明	有効な値	任意かどうか
contentType		・CLI ◦ TEMPLATE_CLI ・POAP ◦ TEMPLATE_CLI POAP オプションは、Cisco Nexus Dashboard Fabric ControllerLAN ファブリックの展開には適用されません。 ・ポリシー ◦ TEMPLATE_CLI ◦ PYTHON ・SHOW ◦ TEMPLATE_CLI ・プロファイル ◦ TEMPLATE_CLI ◦ PYTHON ・ファブリック ◦ PYTHON ・[抽象（ABSTRACT）] ◦ TEMPLATE_CLI ◦ PYTHON ・レポート ◦ PYTHON	はい
実装（Implement）	抽象テンプレートを実装するために使用されます。	テキスト（Text）	はい
依存関係	スイッチの特定の機能を選択するために使用されます。	テキスト（Text）	はい
公開	テンプレートを読み取り専用としてマークし、変更を回避するために使用されます。	「true」または「false」	はい

テンプレート変数

このセクションには、テンプレートに使用されるパラメータの宣言された変数、データ型、デフォルト値、および有効な値の条件が含まれます。これらの宣言された変数は、動的コマンド生成プロセス中にテンプレート コンテンツ セクションの値の置換に使用されます。また、これらの変数は、意思決定およびテンプレート コンテンツ セクションの反復ブロックで使用されます。変数には事前定義されたデータ型があ

ります。変数に関する説明を追加することもできます。次の表に、使用可能なデータ型の構文と使用方法を示します。

変数の型	有効値	反復可能?
boolean	true|false	いいえ
enum	例：copy running-config、startup-config	いいえ
浮動	浮動小数点形式	いいえ
floatRange	例：10.1、50.01	はい
整数型（Integer）	任意の数値	いいえ
integerRange	「-」で区切られた連続する番号「,」で区切られた個別の番号 例：1-10、15、18、2	はい
インターフェイス	形式：<if type><slot>[/<sub slot>]/<port> 例：eth1/1、fa10/1/2 など。	いいえ
interfaceRange	例：eth10/1/20-25、eth11/1-5	はい
IPアドレス	IPv4 または IPv6 アドレス	いいえ

変数の型	有効値	反復可能?
ipAddressList	IPv4 、 IPv6 、 または 両方の タイプ の アドレスの組み合わせのリストを作成できます。 例 1：172.22.31.97、 172.22.31.99、 172.22.31.105、 172.22.31.109 例 2： 2001:0db8:85a3:0000:0000:8a2e:0370:7334、 2001:0db8:85a3:0000:0000:8a2e:0370:7335、 2001:0db8:85a3:1230:0000:8a2f:0370:7334 例 3：172.22.31.97, 172.22.31.99, 2001:0db8:85a3:0000:0000:8a2e:0370:7334、 172.22.31.254	はい
ipAddressWithoutPrefix	例：192.168.1.1 または 例：1:2:3:4:5:6:7:8	いいえ
ipV4Address	IPv4 アドレス	いいえ
ipV4AddressWithSubne t	例：192.168.1.1/24	いいえ
ipV6Address	[IPv6 アドレス（IPv6 address）]	いいえ
ipV6AddressWithPrefix	例：1:2:3:4:5:6:7:8 22	いいえ
ipV6AddressWithSubne t	IPv6アドレスとサブネット	いいえ

変数の型	有効値	反復可能?
ISISNetAddress	例：49.0001.00a0.c96b.c490.00	いいえ
long	例: 100	いいえ
MAC アドレス	14 または 17 文字長の MAC アドレス形式	いいえ

string	変数の説明などに使用される自由テキスト
	
	例:
	string scheduledTime
	{
	regularExpr=^([01]\d|2[0-3]):([0-5]\d)$;
	}
		いいえ

string[]	例：{a,b,c,str1,str2}	はい

構造体	単一の変数にバンドルされているパラメータのセット。
	
	struct <structure name declaration > {
	<parameter type> <parameter 1>;
	<parameter type> <parameter 2>;
	.....
	} [<structure_inst1>] [, <structure_inst2>] [,
	<structure_array_inst3 []>];
	
	struct interface_detail
	{ string inf_name;
	string inf_description;
	ipAddress inf_host;
	enum duplex {
	validValues = auto、full、half;
	};
	}myInterface, myInterfaceArray[];
		いいえ
		
		構造体変数が配列として宣言されている場合、変数は反復型です。

wwn（Cisco Nexus Dashboard Fabric Controller Web クライアントでのみ使用可能）	例：20:01:00:08:02:11:05:03	いいえ

可変メタ プロパティ

テンプレート変数セクションで定義されている各変数には、一連のメタ プロパティがあります。メタ プロパティは、主に変数に定義されている検証ルールです。

次の表に、使用可能な変数タイプに適用されるさまざまなメタ プロパティを示します。変数メタ プロパティ テーブル、第 1 部

変数の型	説明	可変メタ プロパティ				
		デフォルト値	有効な値	10 進数の長さ	最低	最大
boolean	ブール値。 例：true	はい				
enum			はい			
浮動	符号付き実数。 例：75.56、-8.5	はい	はい	はい	はい	はい
floatRange	符号付き実数の範囲。 例：50.5 - 54.75	はい	はい	はい	はい	はい
integer	符号付き実数。 例：50、-75	はい	はい		はい	はい
integerRange	符号付き実数の範囲。 例：50-65	はい	はい		はい	はい
インターフェイス	インターフェイス/ポートを指定します。 例：イーサネット 5/10	はい	はい			
interfaceRange		はい	はい			
IPアドレス	IPv4 または IPv6 フォーマット	はい				

ipAddressList	IPv4、IPv6、または両方のタイプのアドレスの組み合わせのリストを作成できます。 例 1： 172.22.31.97、 172.22.31.99、 172.22.31.105、 172.22.31.109 例 2： 2001:0db8:85a3:0000:0000:8a2e:0370:7334, 2001:0db8:85a3:0000:0000:8a2e:0370:7335, 2001:0db8:85a3:1230:0000:8a2f:0370:7334 例 3： 172.22.31.97、 172.22.31.99、 2001:0db8:85a3:0000:0000:8a2e:0370:7334, 172.22.31.254 リスト内のアドレスは、ハイフンではなくカンマで区切ります。	はい				
ipAddressWithoutPrefix	IPv4 または IPv6 アドレス（プレフィックス/サブネットは不要）。					
ipV4Address	IPv4 アドレス	はい				
ipV4AddressWithSubnet	IPv4アドレスとサブネット	はい				
ipV6Address	[IPv6 アドレス（IPv6 address）]	はい				
ipV6AddressWithPrefix	プレフィックス付きIPv6アドレス	はい				
ipV6AddressWithSubnet	IPv6アドレスとサブネット	はい				
ISISNetAddress	例： 49.0001.00a0.c96b.c490.00					

long	例: 100	はい			はい	はい
MAC アドレス	MAC アドレス					

string	リテラル文字列
	string 正規表現の例：
	string scheduleTime
	{
	regularExpr=^([01]\d|2[
	0-3]):([0-5]\d)$;
	}
		はい				

string[]	カンマ　（、）で区切られた文字列リテラル 例：{string1, string2}	はい				
構造体	単一の変数にバンドルされているパラメータのセット。 struct <structure name declaration > { <parameter type> <parameter 1>; <parameter type> <parameter 2>; ..... } [<structure_inst1>] [, <structure_inst2>] [, <structure_array_inst3 []>];					
wwn	WWN アドレス					

変数メタ プロパティ テーブル、第 2 部

変数の型	説明	可変メタ プロパティ						
		最小ス	最大スロット	最小ポ	最大ポート	最小長	最大長	正規表現

		ロット		ート				
boolean	ブール値。 例：true							
enum								
浮動	符号付き実数。 例：75.56、-8.5							
floatRange	符号付き実数の範囲。 例：50.5 - 54.75							
integer	符号付き実数 例：50、-75							
integerRange	符号付き実数の範囲。 例：50-65							
インターフェイス	インターフェイス/ポートを指定します。 例：イーサネット 5/10	はい	はい	はい	はい			
interfaceRange		はい	はい	はい	はい			
IPアドレス	IPv4 または IPv6 形式の IP アドレス。							

ipAddressList	IPv4、IPv6、または両方のタイプのアドレスの組み合わせのリストを作成できます。 例 1： 172.22.31.97、 172.22.31.99、 172.22.31.105、 172.22.31.109 例 2： 2001:0db8:85a3:0000:0000:8a2e:0370:7334, 2001:0db8:85a3:0000:0000:8a2e:0370:7335, 2001:0db8:85a3:1230:0000:8a2f:0370:7334 例 3： 172.22.31.97、 172.22.31.99、 2001:0db8:85a3:0000:0000:8a2e:0370:7334, 172.22.31.254 リスト内のアドレスは、ハイフンではなくカンマで区切ります。							
プレフィックスなしのipAddress	IPv4 または IPv6 アドレス（プレフィックス/サブネットは不要）。							
ipV4Address	IPv4 アドレス。							
ipV4AddressWithSubnet	サブネット付きのIPv4 アドレス 。							
ipV6Address	IPv6 アドレス。							
ipV6AddressWithPrefix	プレフィックス付きのIPv6 アドレス 。							
ipV6AddressWithSubnet	サブネット付きのIPv6 アドレス 。							

ISISNetAddress	例： 49.0001.00a0.c96b.c490.00							
long	例: 100							
MAC アドレス	MAC アドレス							

string	リテラル文字列
	
	string 正規表現の例：
	string scheduledTime
	{
	
	regularExpr=^([01]\d|2[0-3]):([0-5]\d)$;
	}
						はい	はい	はい

string[]	カンマ（、）で区切られた文字列リテラル 例：{string1, string2}							

構造体	単一の変数にバンドルされているパラメータのセット。
	
	struct <structure name declaration > {
	<parameter type> <parameter 1>;
	<parameter type> <parameter 2>;
	.....
	}
	[<structure_inst1>] [, <structure_inst2>] [, <structure_array_inst3 []>];							

wwn	WWN アドレス							

例：メタ プロパティの使用

```
##template variables

integer VLAN_ID
{ min = 100;
max= 200;
};

string USER_NAME
{ defaultValue = admin123;
minLength = 5;
};

struct
    interface_a{ string
    inf_name; string
    inf_description;
    ipAddress inf_host;
    enum duplex {
        validValues = auto、full、half;
    };
}myInterface;
```

```
##
```

可変注釈

注釈を使用して変数をマーキングする変数プロパティを設定できます。

可変注釈は、POAP でのみ使用できます。ただし、注釈は
テンプレート タイプ「CLI」への影響は、ありません。 テンプレート変数
セクションでは、次の注釈を使用できます。

注釈キー	有効な値	説明
AutoPopulate	テキスト（Text）	あるフィールドから別のフィールドに値をコピーします。
DataDepend	テキスト	
説明	[テキスト（Text）]	ウィンドウに表示されるフィールドの説明
DisplayName	テキスト（Text） スペースがある場合は、テキストを引用符で囲みます。	ウィンドウに表示されるフィールドの表示名
列挙体	Text1、Text2、Text3 など	選択するテキストまたは数値をリストします
IsAlphaNumeric	「true」または「false」	文字列には、英数字を使用します。
IsAsn	「true」または「false」	
IsDestinationDevice	「true」または「false」	
IsDestinationFabric	「true」または「false」	
IsDestinationInterface	「true」または「false」	
IsDestinationSwitchName	「true」または「false」	
IsDeviceID	「true」または「false」	
IsDot1qId	「true」または「false」	
IsFEXID	「true」または「false」	
IsGateway	「true」または「false」	IP アドレスがゲートウェイかどうかを検証します。
IsInternal	「true」または「false」	フィールドを内部にし、ウィンドウに表示しません。 この注釈は、ipAddress 変数にのみ使用します。

注釈キー	有効な値	説明
IsManagementIP	「true」または「false」 この注釈は、変数「ipAddress」に対してのみマークする必要があります。	
is_mandatory	「true」または「false」	値をフィールドに強制的に渡す必要があるかどうかを検証します
IsMTU	「true」または「false」	
IsMultiCastGroupAddress	「true」または「false」	
IsMultiLineString	「true」または「false」	文字列　　フィールドを複数行の文字列テキスト エリアに変換します
IsMultiplicity	「true」または「false」	
IsPassword	「true」または「false」	
IsPositive	「true」または「false」	値が正であるかどうかを確認します。
IsReplicationMode	「true」または「false」	
IsShow	「true」または「false」	ウィンドウのフィールドを表示または非表示にします
IsSiteId	「true」または「false」	
IsSourceDevice	「true」または「false」	
IsSourceFabric	「true」または「false」	
IsSourceInterface	「true」または「false」	
IsSourceSwitchName	「true」または「false」	
IsSwitchName	「true」または「false」	
IsRMID	「true」または「false」	
IsVPCDomainID	「true」または「false」	
IsVPCID	「true」または「false」	
IsVPCPeerLinkPort	「true」または「false」	
IsVPCPeerLinkPortChannel	「true」または「false」	
IsVPCPortChannel	「true」または「false」	
[パスワード（Password）]	テキスト（Text）	パスワードフィールドを検証します
PeerOneFEXID	「true」または「false」	
PeerTwoFEXID	「true」または「false」	
PeerOnePCID	「true」または「false」	
PeerTwoPCID	「true」または「false」	
PrimaryAssociation		

ReadOnly	「true」または「false」	フィールドを読み取り専用にします
ReadOnlyOnEdit	「true」または「false」	
SecondaryAssociation	テキスト（Text）	
注釈キー	有効な値	説明
セクション		
UsePool	「true」または「false」	
UseDNSReverseLookup		
ユーザ名	テキスト（Text）	ウィンドウにユーザ名フィールドを表示します。
警告	テキスト（Text）	Description 注釈をオーバーライドするテキストを提供します。

例：**AutoPopulate** 注釈

```
##template variables
string BGP_AS;
 @(AutoPopulate=" BGP_AS" )
  string SITE_ID;
##
```

例：**DisplayName**注釈

```
##template variables
@(DisplayName=" Host Name" , Description = " Description of the host" ) String
hostname;
@(DisplayName=" Host Address" , Description = "  test description"  IsManagementIP=true)
ipAddress hostAddress;
##
```

例：**IsMandatory**注釈

```
##template variables
@(IsMandatory=" ipv6!=null" )
ipV4Address ipv4;
@(IsMandatory=" ipv4!=null" )
ipV6Address ipv6;
##
```

例：**IsMultiLineString**注釈

```
##template variables
@(IsMultiLineString=true)
string EXTRA_CONF_SPINE;
##
```

IsShow注釈

例 1##template variables
boolean isVlan;
@(IsShow=" isVlan==true")
integer vlanNo;
##

例 2##template variables boolean
enableScheduledBackup;
@(IsShow=" enableScheduledBackup==true" ,Description=" Server time")
string scheduledTime;
##
条件「enableScheduledBackup==true」は、true/false に評価されます。

例 3##template variables
@(Enum=" Manual,Back2BackOnly,ToExternalOnly,Both")
string VRF_LITE_AUTOCONFIG;
@(IsShow=" VRF_LITE_AUTOCONFIG!=Manual" , Description=" Target Mask") integer
DCI_SUBNET_TARGET_MASK
##
条件「VRF_LITE_AUTOCONFIG!=Manual」は文字列比較に一致し、true または false に評価されます。

例：警告の注釈

##template variables
@(Warning=" This is a warning msg")
string SITE_ID;
##

テンプレートの内容

この項には、テンプレートで使用する構成コマンドと、すべてのパラメータが含まれています。これらのコマンドには、テンプレート変数セクションで宣言された変数を含めることができます。コマンド生成プロセス中に、変数の値がテンプレートの内容に適切に置き換えられます。

含めるコマンドは、グローバル構成コマンド モードを他のデバイスに入力する時
のように指定する必要があります。コマンドを指定するときは、コマンドモードを考慮する必要があります。

テンプレートの内容は、変数の使用によって決まります。

- スカラ変数：反復に使用できない値の範囲または配列を取得しません（

変数タイプテーブルでは、 iterate-able が「No」としてマークされています）。スカラ変数はテンプレートの内容内で定義する必要があります。

構文：$$<variable
name>$$ Example:
$$USER_NAME$$

- 反復変数：ブロックの反復に使用されます。これらのループ変数は、次に示すように、繰り返しブロック内でアクセスする必要があります。

構文：@<loop variable>
例：
foreach val in
$$INTEGER_RANGE_VALUE$$ { @val
}

- スカラー構造体変数：構造体メンバー変数は、テンプレートの内容からアクセスできます。

構文：$$<structure instance name>.<member variable
name>$$ Example: $$myInterface.inf_name$$

- 配列構造変数：構造体のメンバー変数は、テンプレートの内容からアクセスできます。

構文：$$<structure instance name>.<member variable
name>$$ Example: $$myInterface.inf_name$$

テンプレート変数に加えて、次のステートメントを使用して、条件付きコマンドと反復コマンドの生成を使用できます。

- if-else if-else ステートメント：その中の変数に割り当てられた値に基づいて、設定コマンドのセットの包含/除外を論理的に決定します。

構文：if(<operand 1> <logical operator> <operand 2>){ command1 ..
command2..
..
}
else if (<operand 3> <logical operator> <operand 4>)
{
Command3 ..
Command4..
..

```
}
else
{
Command5 ..
Command6..
..
}
例：if-else if-else ステートメント
if($$USER_NAME$$ == 'admin'){
Interface2/10
no shut
}
else
{ Interface2/1
0 shut
}
```

- foreach ステートメント：コマンドのブロックを反復するために使用されます。反復は、割り当てられたループ変数値に基づいて実行されます。

```
構文：
foreach <loop index variable> in $$<loop
variable>$$ { @<loop index variable> ..
}
例：foreach Statement
foreach ports in $$MY_INF_RANGE$${ interface
@ports
no shut
}
```

- オプション パラメータ：デフォルトでは、すべてのパラメータが必須です。パラメータをオプションにするには、パラメータに注釈を付ける必要があります。

変数セクションには、次のコマンドを含めることができます。

- @(IsMandatory=false)
- Integerfrequency;

  テンプレートの内容の項では、「if」条件チェックを使用せずに、パラメータに値を割り当てることで、コマンドを除外または含めることができます。オプションのコマンドは、次のように構成できます。

- probeicmp[frequencyfrequency-value][timeoutseconds][retry-countretry-count-value]

高度な機能

次に、テンプレートの構成に使用できる高度な機能を示します。

- 割り当て操作

構成テンプレートは、テンプレート コンテンツ セクション内の変数値の割り当てをサポートします。変数の宣言されたデータ型の値が検証されます。不一致がある場合、値は割り当てられません。

割り当て操作は、次のガイドラインに従って使用できます。

- 左側の演算子は、テンプレート パラメータまたは for ループ パラメータのいずれかである必要があります。
- 正しい値の演算子は、テンプレート パラメータ、ループ パラメータ、引用符で囲まれたリテラル文字列値、または単純な文字列値のいずれかの値です。

ステートメントがこれらのガイドラインに従っていない場合、またはこの形式に適合しない場合は、割り当て操作とは見なされません。これは、他の通常の行と同様に、コマンド生成時に置き換えられます。

例：割り当て操作 ##template プロパティを持つテンプレート

```
name =vlan creation;
userDefined= true;
supportedPlatforms = All;
templateType = CLI;
published = false;
##
##template variables
integerRange vlan_range;
@(internal=true)
integer vlanName;
##
##template content
foreach vlanID in $$vlan_range$${ vlan
@vlanID
$$vlanName$$=@vlanID
name myvlan$$vlanName$$
}
##
```

- Evaluate メソッド

設定テンプレートは、Java ランタイムが提供する Java スクリプト環境を使用して、算術演算（ADD、SUBTRACT など）、文字列操作などを実行します。

テンプレート リポジトリ パスで JavaScript ファイルを見つけます。このファイルには、算術文字列関数の主要なセットが含まれています。カスタム JavaScript メソッドを追加することもできます。

これらのメソッドは、次の形式の設定テンプレート コンテンツ セクションから呼び出すことができます。

```
例 1：
$$somevar$$ = evalscript(add, " 100" , $$anothervar$$)
```

また、次のような if 条件の内部で *evalscript* を呼び出すことができます。

```
if($$range$$ > evalscript(sum, $$vlan_id$$, -10)){ do
something...
}
```

Java スクリプト ファイルのバックエンドにあるメソッドを呼び出すことができます。

- 動的な決定

構成テンプレートは、特殊な内部変数「LAST_CMD_RESPONSE」を提供します。この変数には、コマンド実行中のデバイスからの最後のコマンド応答が格納されます。これは、デバイスの状態に基づいてコマンドを提供するための動的な決定を行うために、構成テンプレートのコンテンツで使用できます。

if ブロックの後には、空の場合もある新しい行で else ブロックを続ける必要があります。

VLAN がデバイス上に存在しない場合の VLAN の作成例。

```
例：Create VLAN
##template content show
vlan id $$vlan_id$$
if ($$LAST_CMD_RESPONSE$$ contains " not found" ) { vlan
$$vlan_id$$
} else {
}
##
```

この特別な暗黙的変数は、「IF」ブロックでのみ使用できます。

- テンプレート参照

すべての変数を定義した基本テンプレートを作成できます。この基本テンプレートは、複数のテンプレートにインポートできます。基本テンプレートの内容は、拡張テンプレートの適切な場所に置き換えられます。インポートしたテンプレート パラメータと内容は、拡張テンプレート内でアクセスできます。

```
例：テンプレート参照ベーステンプ
レート：
##template properties
```

```
    name =a vlan base;
    userDefined= true;
    supportedPlatforms = All;
    templateType = CLI;
    published = false;
    timestamp = 2015-07-14 16:07:52; imports
    = ;
##
##template
    variables integer
    vlan_id;
##
##template content
    vlan $$vlan_id$$
##

派生テンプレート：
##template プロパティ
    name =a vlan extended;
    userDefined= true;
    supportedPlatforms = All;
    templateType = CLI;
    published = false;
    timestamp = 2015-07-14 16:07:52; imports
    = a vlan base,template2;
##
##template variables
    interface vlanInterface;
##
##template content
    <substitute a vlan base>
    interface $$vlanInterface$$
    <substitute a vlan base>
....
```

拡張テンプレートを起動すると、基本テンプレートのパラメータ入力も取得されます。また、置換された内容は、完全な CLI コマンドの生成に使用されます。

レポート テンプレート

REPORT テンプレートのテンプレート タイプは python で、2 つのサブタイプ（UPGRADE と GENERIC）があります。

アップグレード

UPGRADE テンプレートは、ISSU 前後のシナリオに使用されます。これらのテンプレートは、ISSU ウィザードに表示されます。

ISSU 前後の処理の詳細については、Nexus Dashboard Fabric Controller にパッケージ化されている

デフォルトのアップグレード テンプレートを参照してください。デフォルトのアップグレード テンプレートは issu_vpc_check です。

ISSU 操作を実行するには、新しい NDFC ユーザーが最初に [ログイン情報管理（Credential Management）] ページで必要なデバイスクレデンシャルを設定する必要があります。最初に適切なデバイス ログイン情報を設定しないと、ISSU 操作を実行できません。

GENERIC

GENERIC テンプレートは、リソース、スイッチ インベントリ、SFP、NVE VNI カウンタに関する情報の収集など、一般的なレポート シナリオに使用されます。このテンプレートを使用して、トラブルシューティング レポートを生成することもできます。

リソース レポート

このレポートには、特定のファブリックのリソース使用状況に関する情報が表示されます。

[サマリ（Summary）]セクションには、すべての情報技術プールと現在の使用率が表示されます。より多くの列を表示するには、ウィンドウの下部にある水平スクロール バーを使用します。

POOL NAME：プールの名前を指定します。

POOL RANGE：プールの IP アドレス範囲を指定します。

SUBNET MASK：サブネット マスクを指定します。

MAX ENTRIES：プールから割り当て可能な最大エントリ数を示します。**USAGE INSIDE RANGE**：プー

ル範囲内に割り当てられている現在のエントリ数を指定します。**USAGE OUTSIDE RANGE**：プール範

囲外に設定されている現在のエントリ数を指定します。

USAGE PERCENTAGE：これは、(範囲内での使用数/最大エントリ数) *100 という式を使用して計算されます。
*100。

[詳細の表示（View Details）]をクリックして、各技術情報プールに割り当てられた、または設定されたリソースのビューを表示します。たとえば、SUBNET の詳細セクションには、サブネット内で割り当てられたリソースに関する情報が含まれます。

スイッチ インベントリ レポート

このレポートは、スイッチ インベントリに関する概要を提供します。

[詳細の表示（View Details）] をクリックして、モジュールとライセンスに関する詳細情報を表示します。

SFPレポート

このレポートは、ファブリックおよびデバイス レベルでの SFP の使用率に関する情報を提供します。

スイッチ インベントリおよび SFP レポートは、Cisco Nexus デバイスでのみサポートされます。

トラブルシューティング レポート

これらのレポートは、トラブルシューティングのシナリオに役立つように生成されます。現在、定義済みのトラブルシューティング レポートは
NVE VNI カウンタ レポートのみです。**NVE VNI** カウンタ レポートの生成では、ネットワーク トラフィックに基づいて上位ヒットの VNI を特定するための定期的なチェックが実行されます。大規模なセットアップでは、レポートの生成頻度を 60 分以上に制限することをお勧めします。

NVE VNI カウンタ レポート

このレポートは、ファブリック内の各 VNI の nve vni counters command コマンド出力を収集します。

最も古いレポートと最新のレポートを比較すると、**[**サマリ（**Summary**）**]** セクションには上位 10 件のヒット VNI が表示されます。上位ヒット VNI は、次のカテゴリに表示されます。

- ユニキャスト トラフィック用の L2 または L3 VNI
- マルチキャスト トラフィック用の L2 または L3 VNI
- ユニキャスト トラフィック用の L2 のみの VNI
- マルチキャスト トラフィック用の L2 のみの VNI
- ユニキャスト トラフィック用の L3 のみの VNI
- マルチキャスト トラフィック用の L3 のみの VNI

最も古いレポートは、現在のレポート タスクで保存された最初のレポートを参照します。現在のレポートと比較する必要がある最初のレポートとして特定のレポートを選択する場合は、選択したレポートが最初で最も古いレポートになるように、選択したレポートよりも古いすべてのレポートを削除します。

たとえば、昨日の午前 8 時、午後 4 時、および午後 11 時に 3 つのレポートが実行されたとします。今日のレポートの最初の最も古いレポートとして午後 11 時にレポートを使用する場合は、昨日の午前 8 時と午後 4 時に実行されました。

定期レポートの場合、最も古いレポートは、期間の開始時刻に実行される最初のレポートです。日次および週次レポートの場合、現在のレポートが以前に生成されたレポートと比較されます。

[サマリ（**Summary**）**]** セクションには、送信された合計バイト数と VNI に関する情報を含むカラムごとのレポートが表示されます。より多くの列を表示するには、ウィンドウの下部にある水平スクロール バーを使用します。

NVE VNI カウンタ レポートの **[**サマリ（**Summary**）**]** セクションでは、スイッチのリロード後またはスイッチのカウンタのクリア後にレポートが生成された場合、[合計送信バイト数（TOTAL TX BYTES）] 列に負の数が表示されます。番号は、後続のレポートで
正しく表示されます。 回避策として、古いレポートをすべて削除するか
スイッチをリロードする前、またはカウンタをクリアする前に新しいジョブを作成することをお勧めします。**[**詳細の表示（**View Details**）**]** をクリックして
詳細情報を表示します。このセクションでは、スイッチごとに NVE VNI とカウンタを示します。

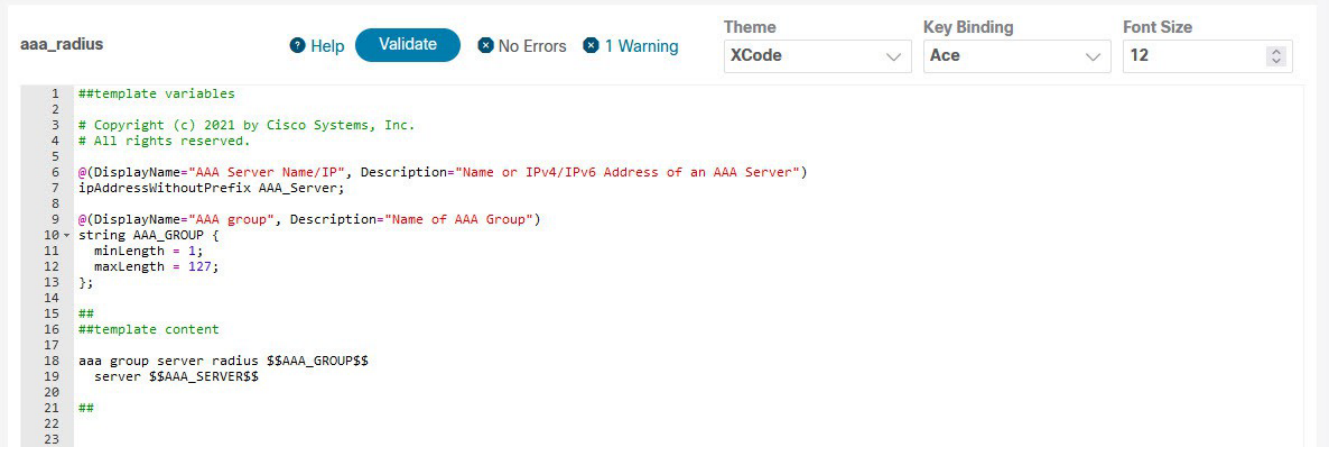
レポートの表示方法の詳細については、「*プログラム可能なレポート*」の章を参照してください。

テンプレートの使用方法

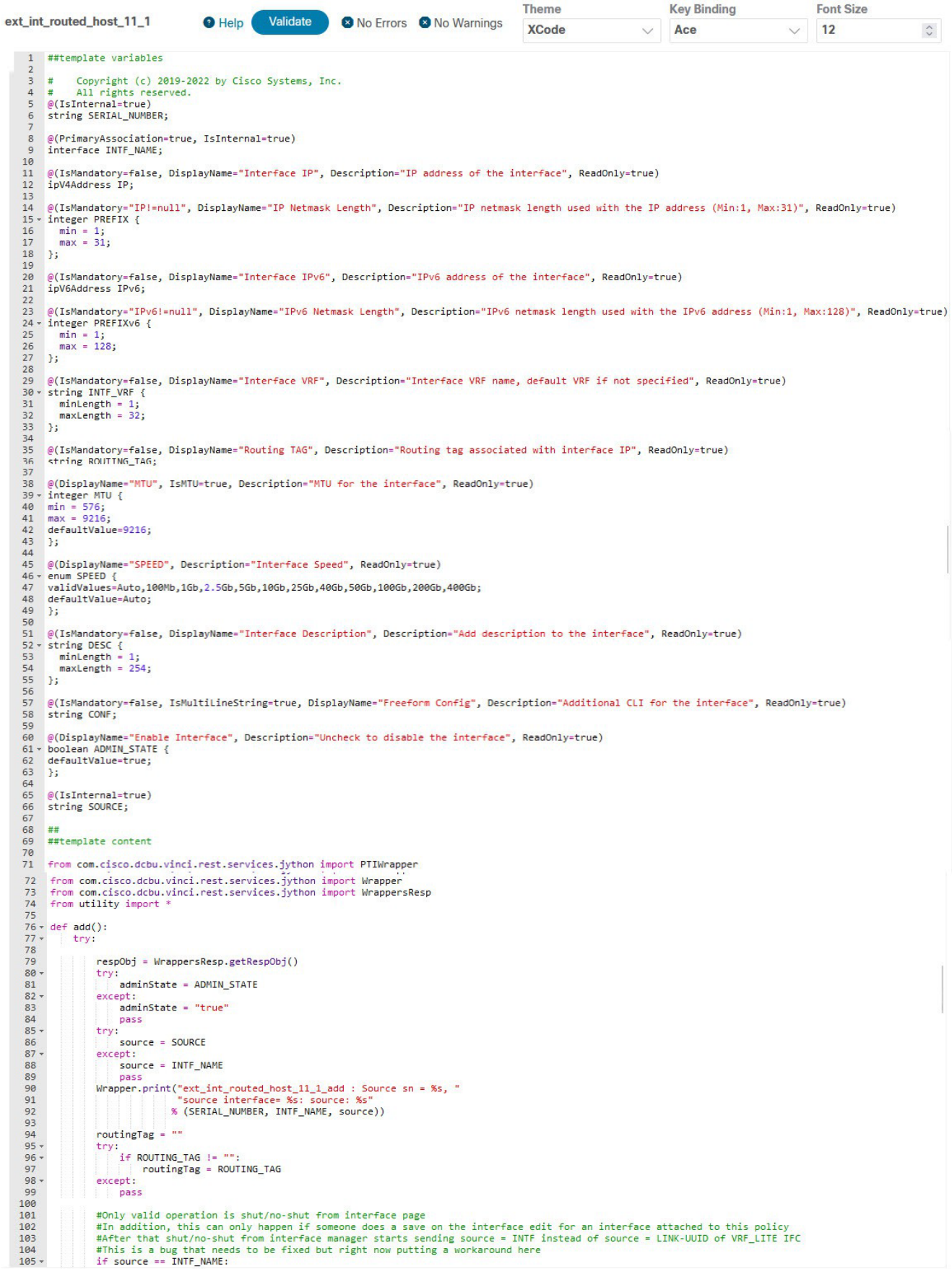
テンプレート タイプ	使用するテンプレートのタイプを指定します。	・ポリシー ・SHOW ・プロファイル ・ファブリック ・レポート ・内部 ・実行
テンプレートのコンテンツ タイプ	テンプレートのコンテンツ タイプを指定します。	・CLI ・PYTHON ・PYTHON3 ・PYTHON3_CLI ・PYTHON_CLI ・TEXT

ポリシーテンプレート

ポリシー テンプレートには、CLI と PYTHON の 2 つのテンプレート コンテンツ タイプがあります。CLI コンテンツ タイプでは、ポリシー テンプレートはパラメータ化された CLI テンプレートです。それらは多くの変数と CLI を持つことができます。通常、CLI ポリシー テンプレートは小さく、if-else-for などのような構造はありません。AAA サーバー構成の CLI ポリシー テンプレートの例を以下に示します。



ただし、テンプレート コンテンツ タイプ PYTHON のポリシー テンプレートを使用することもできます。基本的に、これにより、複数の CLI ポリシー テンプレートを共通の「送信元」と組み合わせて、一度にすべての適用/適用解除を行うことができます。たとえば、vPC ホスト ポートを作成する場合、vPC ペアの一部である両方のピアで対称的に作成する必要があります。さらに、ポートチャネル、メンバー インターフェイス、チャネル グループなどを作成する必要があります。これが、Python vPC ホスト ポリシー テンプレートが追加された理由です。ルーテッド インターフェイスを設定するためのインターフェイス PYTHON テンプレートの例を以下に示します。



各ポリシー テンプレートには、DEVICE、INTERFACE などのテンプレート サブタイプがあります。これにより、適切なポリシー テンプレートが適切な選択ポイントに表示されます。たとえば、[インターフェイス（Interface）] ウィンドウには、インターフェイス ポリシー テンプレートのみが表示されます。

これらのテンプレートのいずれかをコピーして、必要に応じてカスタマイズできます。これは、

カスタマイズの典型的な使用例です。 既存のポリシーを変更せずにコピーを作成し、要件に従ってカスタマイズしてください。そうしないと、DCNM のアップグレード後に変更が失われる可能性があります。

一般に、すでに使用されているテンプレート、つまりファブリック内のスイッチにすでに適用されているテンプレートは編集できません。

ヒ

[タイプなし（No Type）]： CLI テンプレートはファブリック コントローラ ペルソナでのみ使用されます。それらはすべて、スーパー セットであるより強力なポリシー テンプレートに置き換えられます。

ファブリックのテンプレート

ファブリック テンプレートは基本的に python テンプレート、具体的には jython、つまり java + python です。ファブリック テンプレートは非常に包括的であり、ファブリック全体内のすべてのスイッチの目的の構成を生成するために必要なすべてのロジックを含む、ファブリックの展開に必要なルールが組み込まれています。構成は、公開されているシスコのベストプラクティス ガイドラインに基づいて生成されます。組み込みルールに加えて、ファブリック テンプレートは、リソース マネージャ、トポロジ データベース、デバイス ロール、構成コンプライアンスなどの他のエンティティとも統合し、ファブリック内のすべてのデバイスに応じて構成を生成します。これは、NDFC ファブリック固有の部分です。

独自のファブリック テンプレートを作成する必要はありません。NDFC は、Data CenterVXLAN EVPN、外部接続ネットワーク、VXLAN EVPN マルチサイト、BGP ファブリックなど、すぐに使用できるいくつかのファブリック テンプレートを提供します。

プロファイル テンプレート

プロファイル テンプレートは、オーバーレイ（ネットワークまたは VRF）のプロビジョニングに使用されます。オーバーレイ構成を適用する場合、複数の構成要素を組み合わせる必要があるという考え方です。たとえば、VXLAN EVPN ファブリックの有効なレイヤ 3 ネットワーク構成には、VLAN、SVI、int nve 構成、EVPN ルート ターゲットなどが必要です。これらの要素はすべて、いわゆる構成プロファイル（NX-OS コンストラクト）にまとめられます。そして、一度に効果的に適用されます。スイッチ上で、構成プロファイル全体が適用されるか、何も適用されません。このようにして、スイッチにぶら下がったり迷い込んだりする構成が残されることはありません。リーフまたはボーダーのいずれの種類のオーバーレイ構成でも、NDFC はプロファイル テンプレートを使用します。

以下に示すように、タグで区別される 4 種類のプロファイル テンプレートがあります。

- ネットワーク プロファイル（ロール リーフを持つすべてのデバイスに適用）
- ネットワーク拡張プロファイル（ロール「border*」を持つすべてのデバイスに適用）
- VRF プロファイル（ロール リーフを持つすべてのデバイスに適用）
- VRF 拡張プロファイル（ロール「border*」を持つすべてのデバイスに適用）

NDFC のネットワークと VRF ワークフローを介してオーバーレイ構成を適用する方法の詳細については、「*ネットワークと VRF の作成と展開*」セクションを参照してください。

補足事項

ポリシーまたはプロファイル テンプレートが適用されると、テンプレートのアプリケーションごとにインスタンスが作成されます。これに使用される一般的な用語は、ポリシー テンプレート インスタンスまたは PTI です。PTI は、実質的にポリシーまたはプロファイル テンプレート + 置換後の特定のインスタンスを与える名前と値の

ペアです。デバイス用に作成された PTI は、ファブリック ビルダのそのデバイスの [ポリシーの表示/編集（View/Edit policies）] オプションで表示できます。表形式のビューでは、[ポリシーの表示/編集（View/Edit policies）] ボタンを使用して、ファブリック全体のデバイスのサブセット全体でポリシーの選択と一括作成/削除を行うことができます。詳細については、「*ポリシーの表示と編集*」セクションを参照してください。

使用中テンプレートのコンテンツの変更

一般に、テンプレートは、ポリシー、ファブリック、またはプロファイル テンプレートのいずれであっても、インスタンス化されると変更できません。ただし、テンプレートのバグを修正したり、すでに展開されている構成を変更したりするなど、テンプレートのコンテンツを編集したい場合もあります。これは、**[管理（Admin）] > [**システム設定（**System Settings** ）**] > [**サーバ設定（**Server Settings**）**] > [LAN -** ファブリック（**LAN-Fabric**）**]** タブで **[**使用中の テンプレートのオーバーライド（**Template In-Use Override**）**]** オプションを切り替えることで実現できます。

1. **[**サーバ設定（**Server Settings**）**]** の **[LAN** ファブリック（**LAN-Fabric**）**]** タブの下の **[**使用中のテンプレートの上書き（**Template In-Use Override**）**]** チェックボックスをオンにします。
2. **[**保存（**Save**）**]** をクリックします。
3. 目的のテンプレートを編集します。
4. [ファブリックの概要（Fabrics Overview）] に移動し、**[**再計算して展開（**Recalculate and Deploy**）**]** をクリックします。

   これにより PTI が再生成され、更新されたコンテンツが取得され、望む構成（またはインテント）に使用されます。

5. コンテンツが再生成され展開されると、**[**使用中のテンプレートのオーバーライド（**Template In-Use Override**）**]** チェックボックスのチェックを外し、パフォーマンスの問題を回避します。

著作権

このマニュアルに記載されている仕様および製品に関する情報は、予告なしに変更されることがあります。このマニュアルに記載されている表現、情報、および推奨事項は、すべて正確であると考えていますが、明示的であれ黙示的であれ、一切の保証の責任を負わないものとします。このマニュアルに記載されている製品の使用は、すべてユーザー側の責任となります。

対象製品のソフトウェア ライセンスと限定保証は、製品に添付された『Information Packet』に記載されており、この参照により本マニュアルに組み込まれるものとします。添付されていない場合には、代理店にご連絡ください。

Cisco が採用している TCP ヘッダー圧縮機能は、UNIX オペレーティング システムの UCB（University of California, Berkeley）のパブリック ドメイン バージョンとして、UCB が開発したプログラムを採用したものです。All rights reserved. Copyright © 1981, Regents of the University of California.

ここに記載されている他のいかなる保証にもよらず、各社のすべてのマニュアルおよびソフトウェアは、障害も含めて「現状のまま」として提供されます。シスコおよび上記代理店は、商品性、特定目的適合、および非侵害の保証、もしくは取り引き、使用、または商慣行から発生する保証を含み、これらに限定することなく、明示または暗黙のすべての保証を放棄します。

いかなる場合においても、シスコおよびその供給者は、このマニュアルの使用または使用できないことによって発生する利益の損失やデータの損傷をはじめとする、間接的、派生的、偶発的、あるいは特殊な損害について、あらゆる可能性がシスコまたはその供給者に知らされていても、それらに対する責任を一切負わないものとします。

このマニュアルで使用している IP アドレスおよび電話番号は、実際のアドレスおよび電話番号を示すものではありません。マニュアルの中の例、コマンド出力、ネットワーク トポロジ図、およびその他の図は、説明のみを目的として使用されています。説明の中に実際の IP アドレスおよび電話番号が使用されていたとしても、それは意図的なものではなく、偶然の一致によるものです。

この製品のマニュアルセットは、偏向のない言語を使用するように配慮されています。このドキュメントセットでの偏向のない言語とは、年齢、障害、性別、人種的アイデンティティ、民族的アイデンティティ、性的指向、社会経済的地位、およびインターセクショナリティに基づく差別を意味しない言語として定義されています。製品ソフトウェアのユーザインターフェイスにハードコードされている言語、RFP のドキュメントに基づいて使用されている言語、または参照されているサードパーティ製品で使用されている言語によりドキュメントに例外が存在する場合があります。

Cisco およびCisco のロゴは、Cisco またはその関連会社の米国およびその他の国における商標または登録商標です。
商標または登録商標です。シスコの商標の一覧は、http://www.cisco.com/ go/trademarks でご確認いただけます。記載されているサードパーティの商標は、それぞれの所有者に帰属します。「パートナー」という言葉が使用されていても、シスコと他社の間にパートナー関係が存在することを意味するものではありません。（1110R）。