



# テンプレート

- [テンプレート, on page 1](#)

## テンプレート

### UI ナビゲーション

- [オペレーション (Operations)] > [テンプレート (Templates)] を選択します。

Cisco Nexus ダッシュボード ファブリック コントローラ Web クライアントを使用して、異なる Cisco Nexus、IOS-XE、IOS-XR、および Cisco MDS プラットフォームで設定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 ダッシュボード ファブリック コントローラ Web クライアントで設定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。

**Table 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	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。

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。



**Note** エラーのあるテンプレートは、[テンプレート (Templates)] ウィンドウに表示されません。エラーがあるテンプレートはインポートできません。このようなテンプレートをインポートするには、エラーを修正してインポートします。

次の表では、[テンプレート (Templates)] ウィンドウに表示される [アクション (Actions)] ドロップダウン リストのアクション項目について説明します。

**Table 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、 <a href="#">新規テンプレートの作成, on page 4</a> 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、 <a href="#">テンプレートの編集, on page 5</a> 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、 <a href="#">テンプレートの編集, on page 5</a> を参照してください。
テンプレートの複数	<p>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。</p> <p>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、[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)] ウィンドウが表示されます。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、<a href="#">テンプレートの編集, on page 5</a>を参照してください。</p>

Actions	説明
テンプレートの削除	<p>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。</p> <p>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。</p> <p>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、<b>[テンプレートの削除 (Delete template)]</b> 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、<b>[確認 (Confirm)]</b> をクリックします。削除しない場合は、<b>[キャンセル (Cancel)]</b> 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。</p> <p><b>Note</b> 複数のテンプレートを選択して、同じインスタンスで削除します。</p> <p>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。</p>
インポート	<p>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、<a href="#">テンプレートのインポート, on page 7</a>を参照してください。</p>
Zip としてインポート	<p>.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</p> <p>ZIPファイル内のすべてのテンプレートが抽出され、個々のテンプレートとしてテーブルにリストされます。</p> <p>詳細については、「<a href="#">テンプレートのインポート, on page 7</a>」を参照してください。</p>

Actions	説明
エクスポート	<p>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。一度にエクスポートできるテンプレートは1つだけです。</p> <p>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選択し、[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がローカルディレクトリにエクスポートされます。</p>

**network-operator** ロールを持つテンプレートのみを表示できます。このロールでテンプレートを作成、編集、または保存することはできません。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。

この項の内容は、次のとおりです。

## 新規テンプレートの作成

### Nexusダッシュボード ファブリック コントローラUI ナビゲーション

- [オペレーション (Operations)] > [テンプレート (Templates)] を選択します。

Cisco Nexusダッシュボード ファブリック コントローラ Web UI からユーザ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。

#### Procedure

**ステップ 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。

**ステップ 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。

**Note** 基本テンプレートは CLI テンプレートです。

**ステップ 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。

**ステップ 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。

**Note** 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)] 列の下に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。

**ステップ 5** [ヘルプ (Help)] をクリックして、右側の [エディタ (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。

**ステップ 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。

**ステップ 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。

**ステップ 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、[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。

**ステップ 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)] ウィンドウに戻ります。

## テンプレートの編集

Nexusダッシュボード ファブリック コントローラUI ナビゲーション

• [オペレーション (Operations)] > [テンプレート (Templates)] を選択します。

ユーザ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。

### Procedure

- ステップ 1** [**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。
- ステップ 2** 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。
- ステップ 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。
- ステップ 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。  
**Note** 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。
- ステップ 5** [**ヘルプ (Help)**] をクリックして、右側の[**エディタ (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。
- ステップ 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。

**ステップ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。

**ステップ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、[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。

**ステップ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**)] ウィンドウに戻ります。

## テンプレートのインポート

### Nexusダッシュボード ファブリック コントローラUI ナビゲーション

- [オペレーション (**Operations**)] > [テンプレート (**Templates**)] を選択します。

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。



#### Note

テンプレート内の「\n」は、インポートおよび編集され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。

Cisco Nexusダッシュボード ファブリック コントローラ Web UI からテンプレートをインポートするには、次の手順を実行します。

#### Procedure

**ステップ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 [テンプレートのインポート (**Import template**)] を選択します。

[テンプレートのインポート (**Import Template**)] ウィンドウが表示されます。

**ステップ2** コンピュータに保存されているテンプレートを参照して選択します。

**ステップ3** [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。

**Note** 圧縮されたテンプレート ファイルをインポートすると、成功またはエラー メッセージが表示されます。[OK] をクリックします。

**ステップ 4** 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、[テンプレートの編集](#) , on page 5を参照してください。

**Note** 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)]** ウィンドウが表示されないことがあります。ただし、必要に応じて**[テンプレートの編集 (Edit Template)]** アクションを使用して、テンプレートパラメータとコンテンツを編集できます。

**ステップ 5** 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)]** ウィンドウに戻ります。

## 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。

この項の内容は、次のとおりです。

### 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	×
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-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。	いいえ

プロパティ名	説明	有効な値	任意かどうか
templateType	使用するテンプレートのタイプを指定します。	<ul style="list-style-type: none"> <li>• CLI</li> <li>• POAP</li> </ul> <p><b>Note</b> POAP オプションは、Cisco Nexus ダッシュボード ファブリック コントローラ LAN ファブリックの展開には適用されません。</p> <ul style="list-style-type: none"> <li>• ポリシー</li> <li>• SHOW</li> <li>• プロファイル</li> <li>• ファブリック</li> <li>• [抽象 (ABSTRACT) ]</li> <li>• レポート</li> </ul>	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。		

プロパティ名	説明	有効な値	任意かどうか
		<ul style="list-style-type: none"> <li>• CLI                             <ul style="list-style-type: none"> <li>• なし</li> </ul> </li> <li>• POAP                             <ul style="list-style-type: none"> <li>• なし</li> <li>• VXLAN</li> <li>• FABRICPATH</li> <li>• VLAN</li> <li>• PMN</li> </ul> </li> <li><b>Note</b> POAP オプションは、Cisco Nexus ダッシュボード ファブリックコントローラ LAN ファブリックの展開には適用されません。</li> <li>• ポリシー                             <ul style="list-style-type: none"> <li>• VLAN</li> <li>• interface-vlan</li> <li>• INTERFACE_VPC</li> <li>• <del>INTERFACE_HRNET</del></li> <li>• INTERFACE_BD</li> <li>• <del>INTERFACE_CHANNEL</del></li> </ul> </li> </ul>	

プロパティ名	説明	有効な値	任意かどうか
		<ul style="list-style-type: none"> <li>• INTERFACE_FC</li> <li>• NIERFACE_MGMT</li> <li>• NIERFACE_LOOBACK</li> <li>• NIERFACE_NVE</li> <li>• NIERFACE_VFC</li> <li>• <del>NIERFACE_CHANNEL</del></li> <li>• DEVICE</li> <li>• FEX</li> <li>• NIRA_FABRIC_LINK</li> <li>• NIER_FABRIC_LINK</li> <li>• INTERFACE</li> <li>• SHOW                             <ul style="list-style-type: none"> <li>• VLAN</li> <li>• NIERFACE_VLAN</li> <li>• NIERFACE_VPC</li> <li>• NIERFACE_ETHNET</li> <li>• INTERFACE_BD</li> <li>• <del>NIERFACE_CHANNEL</del></li> <li>• INTERFACE_FC</li> <li>• NIERFACE_MGMT</li> <li>• NIERFACE_LOOBACK</li> <li>• NIERFACE_NVE</li> <li>• NIERFACE_VFC</li> <li>• <del>NIERFACE_CHANNEL</del></li> <li>• DEVICE</li> <li>• FEX</li> <li>• NIRA_FABRIC_LINK</li> <li>• NIER_FABRIC_LINK</li> <li>• INTERFACE</li> </ul> </li> </ul>	

プロパティ名	説明	有効な値	任意かどうか
		<ul style="list-style-type: none"> <li>• プロファイル                             <ul style="list-style-type: none"> <li>• VXLAN</li> </ul> </li> <li>• ファブリック                             <ul style="list-style-type: none"> <li>• 該当なし</li> </ul> </li> <li>• [抽象 (ABSTRACT) ]                             <ul style="list-style-type: none"> <li>• VLAN</li> <li>• NIERFACE_VLAN</li> <li>• INIERFACE_VPC</li> <li>• NIERFACE_EIHRNET</li> <li>• INIERFACE_BD</li> <li>• <del>NIERFACE_CHANL</del></li> <li>• INIERFACE_FC</li> <li>• NIERFACE_MGMT</li> <li>• NIERFACE_COBACK</li> <li>• INIERFACE_NVE</li> <li>• INIERFACE_VFC</li> <li>• <del>NIERFACE_CHANL</del></li> <li>• DEVICE</li> <li>• FEX</li> <li>• NIERFACE_FRCINK</li> <li>• NIERFACE_FRCINK</li> <li>• INTERFACE</li> </ul> </li> <li>• レポート                             <ul style="list-style-type: none"> <li>• アップグレード</li> <li>• GENERIC</li> </ul> </li> </ul>	

プロパティ名	説明	有効な値	任意かどうか
contentType			はい

プロパティ名	説明	有効な値	任意かどうか
		<ul style="list-style-type: none"> <li>• CLI                             <ul style="list-style-type: none"> <li>• TEMPLATE_CLI</li> </ul> </li>   <li>• POAP                             <ul style="list-style-type: none"> <li>• TEMPLATE_CLI</li> </ul> </li>   <li><b>Note</b> POAP                              オプション                              は、Cisco                              Nexus                              ダッ                              シュ                              ボード                              ファブ                              リック                              コント                              ローラ                              LAN                              ファブ                              リック                              の展開                              には適                              用され                              ませ                              ん。                         </li>   <li>• ポリシー                             <ul style="list-style-type: none"> <li>• TEMPLATE_CLI</li> <li>• PYTHON</li> </ul> </li>   <li>• SHOW                             <ul style="list-style-type: none"> <li>• TEMPLATE_CLI</li> </ul> </li>   <li>• プロファイル                             <ul style="list-style-type: none"> <li>• TEMPLATE_CLI</li> <li>• PYTHON</li> </ul> </li>   <li>• ファブリック                             <ul style="list-style-type: none"> <li>• PYTHON</li> </ul> </li> </ul>	



プロパティ名	説明	有効な値	任意かどうか
		<ul style="list-style-type: none"> <li>• [抽象 (ABSTRACT) ]</li> <li>• TEMPLATE_CLI</li> <li>• PYTHON</li> <li>• レポート</li> <li>• PYTHON</li> </ul>	
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	Example: running-config, startup-config	いいえ
浮動	浮動小数点形式	いいえ
floatRange	Example: 10.1, 50.01	はい
整数型 (Integer)	任意の数値	いいえ

変数の型	有効値	反復可能?
integerRange	「-」で区切られた連続する番号 「,」で区切られた個別の番号  Example: 1-10,15,18,20	はい
インターフェイス	形式: <if type><slot>[/<sub slot>]/<port>  Example: eth1/1, fa10/1/2 etc.	いいえ
interfaceRange	Example: 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
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。  Example 1: 172.22.31.97, 172.22.31.99, 172.22.31.105, 172.22.31.109  Example 2: 2001:0cb8:85a3:0000:0000:8a2e:0370:7334,  2001:0cb8:85a3:0000:0000:8a2e:0370:7335,  2001:0cb8:85a3:1230:0000:8a2f:0370:7334 Example 3: 172.22.31.97, 172.22.31.99,  2001:0cb8:85a3:0000:0000:8a2e:0370:7334,  172.22.31.254	はい
ipAddressWithoutPrefix	Example: 192.168.1.1  または Example: 1:2:3:4:5:6:7:8	いいえ
ipV4Address	IPv4 アドレス	いいえ
ipV4AddressWithSubnet	Example: 192.168.1.1/24	いいえ
ipV6Address	[IPv6 アドレス (IPv6 address) ]	いいえ

変数の型	有効値	反復可能?
ipV6AddressWithPrefix	Example: 1:2:3:4:5:6:7:8 22	いいえ
ipV6AddressWithSubnet	IPv6アドレスとサブネット	いいえ
ISISNetAddress	Example: 49.0001.00a0.c96b.c490.00	いいえ
long	Example: 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  Example: string scheduledTime { regularExpr="^([01]\d 2[0-3]):([0-5]\d)\$"; }	いいえ
string[]	Example: {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。  <pre>struct &lt;structure name declaration &gt; { &lt;parameter type&gt; &lt;parameter 1&gt;; &lt;parameter type&gt; &lt;parameter 2&gt;; .... } [&lt;structure_inst1&gt;] [, &lt;structure_inst2&gt;] [, &lt;structure_array_inst3 []&gt;;</pre> <pre>struct interface_detail { string inf_name; string inf_description; ipAddress inf_host; enum duplex { validValues = auto, full, half; }; }myInterface, myInterfaceArray[];</pre>	いいえ  <b>Note</b> 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。

変数の型	有効値	反復可能?
wwn (Cisco Nexusダッシュボード ファブリック コントローラ Web クライアントでのみ使用 可能)	Example: 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	最小スロット	最大スロット	最小ポート	最大ポート	最小長	最大長	正規表現
boolean	ブール値。 Example: true	はい											
enum			はい										
浮動	符号付き実数。 Example: 75.56, -8.5	はい	はい	はい	はい	はい							
floatRange	符号付き実数の範囲 Example: 50.5 - 54.75	はい	はい	はい	はい	はい							

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	最小スロット	最大スロット	最小ポート	最大ポート	最小長	最大長	正規表現
integer	符号付き実数 Example: 50, -75	はい	はい		はい	はい							
integer	符号付き実数の範囲 Example: 50-65	はい	はい		はい	はい							
interface	インターフェイス/ポートを指定します Example: Ethernet 5/10	はい	はい				はい	はい	はい	はい			
interface		はい	はい				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	はい											

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	最小スロット	最大スロット	最小ポート	最大ポート	最小長	最大長	正規表現
<code>boolean</code>		はい											

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	最小スロット	最大スロット	最小ポート	最大ポート	最小長	最大長	正規表現
	<p>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。</p> <p>Example 1:  <code>IP223.9, IP223.9, IP223.15, IP223.10</code></p> <p>Example 2:  <code>IP10.20, IP10.20, IP10.20, IP10.20</code></p> <p>Example 3:  <code>IP223.9, IP223.9, IP10.20, IP223.23</code></p> <p><b>Note</b></p>	リス											

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	最小スロット	最大スロット	最小ポート	最大ポート	最小長	最大長	正規表現
		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。											



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	最小スロット	最大スロット	最小ポート	最大ポート	最小長	最大長	正規表現
<del>IPv4</del>	IPv4 または IPv6 アドレス (プレフィックス/サブネットは不要)。												
<del>IPv4</del>	IPv4 アドレス	はい											
<del>IPv4</del>	IPv4 アドレスとサブネット	はい											
<del>IPv6</del>	[IPv6 アドレス (IPv6 <del>ads</del> ) ]	はい											

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	最小スロット	最大スロット	最小ポート	最大ポート	最小長	最大長	正規表現
IPv6	プレフィックス付き IPv6 アドレス	はい											
IPv6	IPv6 アドレスとサブネット	はい											
IPv6	Example: <code>4013:3:1:1::1</code>												
long	Example: 100	はい			はい	はい							
MAC	MAC アドレス												
string	リテラル文字列  Example for string  Regular expression string  string { <code>^0123\$</code> }	はい									はい	はい	はい

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	最小スロット	最大スロット	最小ポート	最大ポート	最小長	最大長	正規表現
string[]	カンマ (,) で区切られた文字列リテラル  Example: {string1, string2}	はい											

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	最小スロット	最大スロット	最小ポート	最大ポート	最小長	最大長	正規表現
構造体	<p>単一の変数にバンドルされているパラメータのセット。</p> <pre> struct &lt;structure name definition &gt; { &lt;parameter type&gt; &lt;parameter 1&gt;; &lt;parameter type&gt; &lt;parameter 2&gt;; ... } [&lt;struct1&gt; [, &lt;struct2&gt; [, &lt;struct3&gt; [1]&gt;; </pre>												
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。



**Note** 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」には影響しません。

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 (Text) ]	
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) <b>Note</b> 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

注釈キー	有効な値	説明
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。  <b>Note</b> この注釈は、ipAddress変数にのみ使用します。
IsManagementIP	「true」または「false」  <b>Note</b> 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。	

注釈キー	有効な値	説明
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領域に変換します
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」	

注釈キー	有効な値	説明
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)	
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

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

```
##template variables
boolean enableScheduledBackup;
@(IsShow="enableScheduledBackup==true",Description="Server time")
string scheduledTime;
##
The condition "enableScheduledBackup==true" evaluates to true/false
```

```
##template variables
@(Enum="Manual,Back2BackOnly,ToExternalOnly,Both")
string VRF_LITE_AUTOCONFIG;
@(IsShow="VRF_LITE_AUTOCONFIG!=Manual", Description="Target Mask")
integer DCI_SUBNET_TARGET_MASK
##
The condition "VRF_LITE_AUTOCONFIG!=Manual" matches string comparison to evaluate to true or false
```

#### 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。



**Note** 使用するコマンドは、任意のデバイスのグローバル構成コマンドモードで入力するのと同じ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。

```
Syntax: $$<variable name>$$
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。

```
Syntax:@<loop variable>
Example:
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。

```
Syntax: $$<structure instance name>.<member variable name>$$
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。

```
Syntax: $$<structure instance name>.<member variable name>$$
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。

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

```

Command5 ..
Command6..
..
}
Example: if-else if-else statement
if($$USER_NAME$$ == 'admin'){
Interface2/10
no shut
}
else {
Interface2/10
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。

```

Syntax:
foreach <loop index variable> in $$<loop variable>$$ {
@<loop index variable> ..
}
Example: 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)**

- **Integer frequency;**

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。

- **probe icmp [frequency frequency-value] [timeout seconds] [retry-count retry-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。

```
Example: Template with assignment operation
##template properties
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。

```
Example1:
$$somevar$$ = 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。



**Note** 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。

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

この特別な暗黙の変数は、「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。

```
Example: Template Referencing
Base template:
##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 $$vlan_id$$
##

Derived Template:
##template properties
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 ダッシュボード ファブリック コントローラ 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。

### 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、すべてのリソース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。

[**詳細の表示 (View Details)**] をクリックして、各リソース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。



(注) 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 **show nve vni counters** 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。



- 
- (注) 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、スイッチをリロードするか、カウンタをクリアする前に、古いレポートをすべて削除するか、新しいジョブを作成することを推奨します。
- 

詳細については、**[詳細の表示 (View Details)]** をクリックしてください。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。