Appendix A: Managing Configuration Templates

The section contains the following topics:

Managing Configuration Templates

Configuration templates can be used when there are multiple devices that have very similar configuration requirements, contain a small number of parameters that need to be different for each device. For example, a network may use identical configuration for all switches, except that each switch has a unique hostname and management IP address. Configuration templates allow you to have a single configuration file containing all the common configuration, with placeholders for the elements of the configuration that need to be unique.

There are two parts to a configuration template – the configuration itself, and the metadata that controls how the placeholders are presented in the user interface when a device record is being created. The following sections describe each of these pieces in detail.

Configuration Syntax

The configuration part of a configuration template is a text document that is very similar to a regular device configuration. When creating a configuration template, the recommended approach is to start from a configuration backup taken from a sample device that is already configured with the features and settings that the template should enable. A configuration template differs from a device configuration in that device-specific parameters – such as a hostname – are replaced with placeholders.

When you create a new device record, you are presented with a form allowing you to supply the correct values for each of the placeholders in the configuration template. These values are merged with the configuration template to generate the actual configuration that will be sent to the device.

Note The placeholder values are merged with the configuration template when the configuration is being sent to the device. This means the final device configuration may be different from that shown in the preview if any system variables change before the device connects to the Manager.

Configurations are created as Mustache templates *https://mustache.github.io/*. Mustache allows a variety of placeholders – referred to as tags in the Mustache documentation – to be used, including:

- Simple variables, where the placeholder is replaced with the value specified in the device record. A simple variable has the form **{{name}}**.
- Sections, where the placeholder encloses a block of configuration – optionally including other placeholders. The content of the section may be excluded from the final configuration, included once or repeated several times.

  The behavior of this type of placeholder is defined by the metadata in the template and the values that the user provides when creating a device record.

  A section has the form **{{#name}}…{{/name}}** where the first tag marks the beginning of the block and the second tag marks the end.
- Comments can be used to document the configuration template. A comment has the form {{**! This is a comment**}}.

Following is an example of a simple template:

```
!
hostname {{hostname}}
!
{{! Insert a list of VLANs}}
{{#vlans}}
interface vlan {{vlan-id}}
 name {{vlan-name}}
!
{{/vlans}}
```

In this example, there are several different placeholders:

- **{{hostname}}** is a simple variable. It will be replaced by the value set for the hostname in the device record.
- There is a comment placed just after the hostname configuration. The comment will not be included in the configuration sent to the device.
- **{{#vlans}}…{{/vlans}}** is a section that is used in this example to hold a list of individual VLANs. For each VLAN defined in the device record, a copy of the contents of this container will be created in the device configuration.
- **{{vlan-id}}** and **{{vlan-name}}** are both simple variables, but they are contained within the {{#vlans}} list. When the device record is created, you may specify multiple values for **{{vlan-id}}** and **{{vlan-name}}** and they will be used to generate the configuration required to create each of those VLANs.

For more details on the Mustache syntax, consult the Mustache man page at *https://mustache.github.io/mustache.5.html*.

Template Metadata

Each configuration template contains metadata that describes how each placeholder should be presented to the user when device records are being created. This metadata is generated when creating templates using the template editor.

When you create or edit a configuration template, the template editor is displayed with the configuration itself displayed on the left and a form on the right that allows you to set the metadata for each placeholder.

Each placeholder in the configuration is shown on the right, along with the following controls:

- A **Required** checkbox. This control determines whether the user must provide a value for this placeholder or not.
- A **Type** drop-down list. This allows you to select the type of placeholder, which controls how that placeholder is displayed to the user.
- A **Title**. This may be used to provide a more user-friendly name for the parameter on the GUI. If a title is not specified for a placeholder, then the placeholder itself with be displayed.
- An **Edit** icon. Certain types have more settings available to control presentation. For example, a string placeholder may be further refined to be an IP address or URL, and the input form will display an error if the text entered does have the correct format. Certain types can also be set based on system information rather than user input. See System and Dynamic Variables below for more details.
- **Move up/down** controls. These arrows allow you to change the order in which placeholders are displayed to the user. Placeholders may be grouped based on what makes the most sense to the user, rather than the order in which they appear in the configuration.

The template editor also provides a preview function which may be used to provide an example of how the placeholders form will appear to the user when creating and editing device records.

Placeholder Types

The following placeholder types are available:

- **String** – Placeholders of this type will be shown in the GUI as a simple text input box.
- **Integer** – Integers are displayed as a text input box with controls to increase or decrease the value of the number displayed. Only numbers may be entered into this field.
- **Boolean** – A Boolean placeholder is displayed in the GUI as a checkbox. If the checkbox is checked, the placeholder be set to the string value ‘true’. If the checkbox is unchecked, the value is ‘false’. A section may also be designated as a boolean, in which case the configuration contained within the section will only be included when the checkbox for the section is checked.
- **Container** – The Container type may be used to group other placeholders in the form.
- **List** – A list is a container or section of configuration that may be repeated multiple times in a generated configuration file. When form elements are generated for the placeholders inside a list, additional controls are added to add or remove elements in the list.

In addition to the simple types listed above, string variables can be further refined by clicking the **edit** icon. Options available include:

- Specifying a default value for the placeholder.
- Setting the minimum and/or maximum length for string placeholders.
- Specifying a pre-defined list of choices that may be selected (using the Enum option).
- Constraining the format of a string to be one of a hostname, URI, IPv4 address or IPv6 address. A string may also be designated as a text area if there is likely to be a significant amount of content to enter.

System and Dynamic Variables

Placeholders can not only take their values from user input, but can also take their values from parameters defined within the system. System variables are parameters that have been defined for the Manager itself – such as the Manager IP address.

By setting a placeholder to take its value from a system variable, the Manager will insert that value into the configuration without any user intervention. Some more complex deployments may require user input for the System Variables to work correctly. See Managing Platform Settings for more details.

Dynamic variables are similar to System variables, but the values generated dynamically are based on information such as the logged in user, or the device group the device belongs to. System and Dynamic variables are used to allow templates to be more portable between devices and systems.

Creating Configuration Templates

The recommended approach to creating config templates is to start by configuring a network device of the appropriate type with the desired settings, then take a backup of the device configuration and upload it to the Manager to use as a starting point.

Alternatively, you can create a copy of an existing template using the 'Save As' function. Either way, starting from an existing configuration can help reduce the time taken to create a template and also to reduce the number of revisions required to achieve the desired result.

When creating a new template, you will need to specify an organization that the template will belong to and the Product IDs (PIDs) that the template may be used with. The Product IDs may contain a ***'s** and **?'s** as wildcard characters.

Once you have your starting configuration created, you may update it using the following process:

1. Navigate to **Network Plug and Play** > **Configurations**,
2. Open your starting configuration in the template editor by selecting the configuration and clicking the **edit** icon.

   The template editor is displayed with the initial configuration file displayed on the left in a text editor window. The text editor supports many common editing functions including search, replace, and several cursor manipulation key sequences. See the table below for a list of commands.
3. Modify the configuration by inserting placeholders as described in Configuration Syntax, on page 1. Each time a new placeholder is inserted, a corresponding entry is added to the form on the right.
4. Modify the metadata associated with each placeholder using the form on the right to ensure that the placeholder is presented to the user in the most appropriate way. See Managing Configuration Templates, on page 1 above for more details on specifying metadata. You can use the Preview function to see how the form will be presented to the user when a device record is being created.
5. Repeat steps 3 and 4 until you have created placeholders for all of the configuration parameters that should vary between devices.
6. Once the template has been completed to your satisfaction, click **Save**.

Note Each time a template is saved, a new version of the template is created. Older versions of templates are retained in the Manager unless you explicitly delete them. When a template is assigned to a device, a specific version of the template is assigned – the latest version by default. As new versions are created, existing devices will continue to use the version that was assigned when they were created. A template version that is currently assigned to a device may not be deleted.

Table 1: Common Editor Commands

Function	Description	Key Bindings	
		PC	Mac
Select All	Select the whole content of the editor.	Ctrl-A	Cmd-A
Kill Line	Deletes the part of the line after the cursor. If that consists only of whitespace, the newline at the end of the line is also deleted.		Ctrl-K
Delete Line	Deletes the whole line under the cursor, including newline at the end.	Ctrl-D	Cmd-D
Undo	Undo the last change.	Ctrl-Z	Cmd-Z
Redo	Redo the last undone change.	Ctrl-Y	Shift-Cmd-Z Cmd-Y
Go Doc Start	Move the cursor to the start of the document.	Ctrl-Home	Cmd-Up Cmd-Home
Go Doc End	Move the cursor to the end of the document.	Ctrl-End	Cmd-End Cmd-Down
Go Line Start	Move the cursor to the start of the line.	Alt-Left	Ctrl-A
Go Line End	Move the cursor to the end of the line.	Alt-Right	Ctrl-E
Indent More	Indent the current line or selection.	Ctrl-]	Cmd-]
Indent Less	Outdent the current line or selection.	Ctrl-[	Cmd-[
Find		Ctrl-F	Cmd-F
Find Next		Ctrl-G	Cmd-G
Find Prev		Shift-Ctrl-G	Shift-Cmd-G
Replace		Shift-Ctrl-F	Cmd-Alt-F
Replace All		Shift-Ctrl-R	Shift-Cmd-Alt-F