



# Cisco VMS Service Extensions

This chapter provides an overview of VMS service extensions and contains the following topics:

- [Understanding How Cisco VMS Service Extensions Work, page 37](#)
- [Creating a VMS Service Extension Template XML File, page 38](#)

## Understanding How Cisco VMS Service Extensions Work

VMS service extensions simplifies how configuration snippets can be applied to a service or a device. VMS leverages the underlying capability of Cisco Network Services Orchestrator (NSO) custom templates, which get pushed along with the derived configuration templates. Service extensions can be used, in most cases, to map services to device configurations, without the need for any additional programming.

VMS service extension templates use variables to map service attributes to the corresponding device configurations and are applied. The service extensions allow a declarative way to describe such manipulations. The VMS operator can apply a service extension template to an existing service chain in VMS or to a device, without having to manually go into the NSO CLI. This service extension template is used by NSO to add, modify, or delete service configuration snippets before NSO pushes the configuration to the devices.

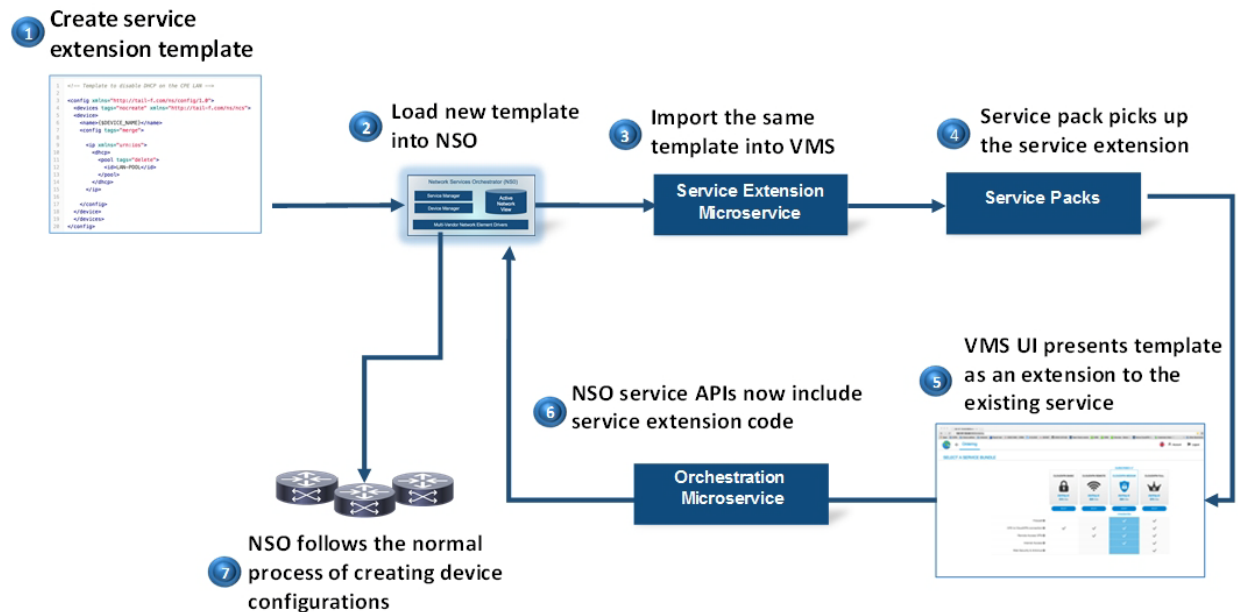
You can apply VMS service extension templates to a service ordering workflow or a single device. When you import a service extension template into VMS, you can specify if the template is to be applied to a service workflow or a device.

When a service extension template is applied to service ordering workflow, VMS service workflow gathers the parameter values the tenant users enter during service ordering process. These values are passed to NSO, which further uses these values in the device configurations.

When a service extension template is applied to a device after the service is ordered, the service extension template is applied to the device outside the workflow, and is available for use for individual devices.

The illustration below depicts the end-to-end workflow that needs to be followed to work with service extension templates in VMS.

Figure 24 VMS Service Extensions Workflow



## Creating a VMS Service Extension Template XML File

The VMS service extension template is an XML file. The structure of that file is defined by the YANG model.

The basic principles of defining a template are as following:

1. A template is an XML file (for example *mytemplate.xml*) that corresponds to a node in the device tree.
2. Each value in a template is stored as a string. This string value is converted to the actual value type of the YANG model when the template is applied.
3. The value part of the XML tag that needs to be configured must be represented with a variable name prefixed with '\$' literal.
4. The templates allow for defining different behavior while applying the template. This is accomplished by setting tags such as `merge`, `replace`, `delete`, `create` or `ncreate` on the relevant nodes in the template.

For example, to create a template to set the NTP server on a device, the sample template XML file should be:

```
<config
  xmlns="http://tail-f.com/ns/config/1.0">
  <device
    xmlns="http://tail-f.com/ns/ncs">
    <name>ntp</name>
    <config tags="merge">
      <system xmlns="http://pica8.org/yang">
        <ntp-server-ip>{$NTP}</ntp-server-ip>
      </system>
    </config>
  </device>
</config>
```

## Creating a VMS Service Extension Template XML File

```
<ntp xmlns="urn:ios">
  <server>
    <server-list>
      <ip-address>{$NTP}</ip-address>
    </server-list>
  </server>
</ntp>
</config>
</devices>
</config>
```

After you create a service extension template, you need to do the following:

- Import the template XML file into NSO. For details see, the service pack guides.
- Import the template XML file into VMS. For details see, the service pack guides.

