CHAPTER 18

Enabling Support for Additional Device Types and Software Versions

The chapter contains the following sections:

Enabling Support for Additional Device Types by Creating U-VNEs

Using the VCB, you can enable Prime Network to discover and manage devices that have no system-defined VNE driver and are therefore not currently supported. To enable support for additional device types, you create user-defined VNE drivers, called U-VNEs. After you have created a U-VNE, you can add it to an AVM using Prime Network Administration, and it will be modeled and added to the Prime Network device inventory. The level of modeling depends on the amount of detail provided to the system when you create the U-VNE.

This section contains the following subsections:

Approaches to Creating a U-VNE

Several approaches are available for creating U-VNEs. Depending on the approach you choose, the modeling capabilities will be more specific or less specific. The available approaches are listed below from most specific to least specific:

1. Create a U-VNE by cloning a developed VNE driver. This approach will produce the most specific results in terms of device modelling. Use this approach when the unsupported device has the same or a very similar management interface, and/or device OS version, as the developed VNE driver.

   For example, a new device that belongs to a supported device series, or runs a new maintenance device OS release would typically share the same management interface as a supported device. A U-VNE created by cloning inherits the behavior of the source VNE driver, and can be further extended using the VCB to add more device module discovery and event recognition. (For supported device types and OS versions, see the *Cisco Prime Network 3.8 Reference Guide*.)

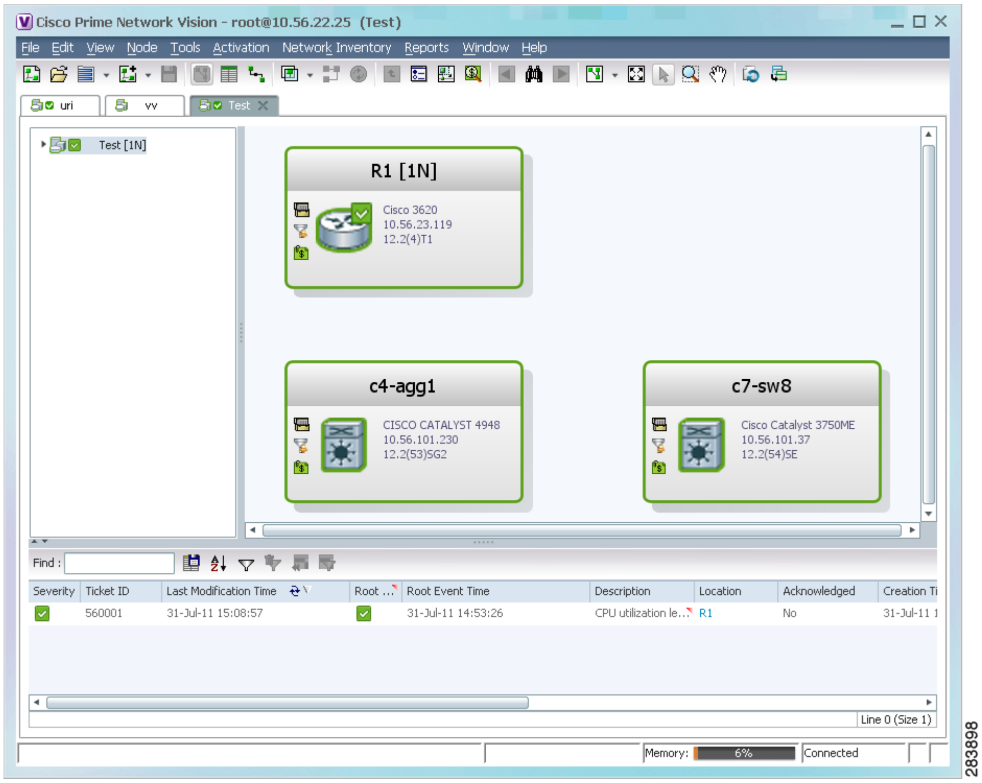
2. Create a U-VNE based on a supported device family. Use this approach when there is no specific developed VNE driver to clone but the unsupported device belongs to a supported device family.
3. Create a U-VNE based on a supported software version. Use this approach if there is no specific device to clone but you know that the unsupported device's software version is supported by a developed VNE driver.
4. Create a U-VNE by referencing a U-VNE template. This approach will produce the least specific results in terms of device modelling. Use this approach when the instrumentation or management interface for the unsupported device does not match that of any supported VNE. Typically, non-Cisco devices and Cisco device families for which there are no developed VNE drivers would be in this category. See U-VNE Templates, page 22-1.

U-VNEs are created as separate VNE definitions in the Prime Network registry, and can be further extended using Prime Network soft-properties and command builder features.

How are U-VNEs Represented in Prime Network?

In the Prime Network GUI, all managed devices are represented by a unique name and an icon, as shown in the Prime Network Vision map below.

Figure 18-1 Device Type Representation in Prime Network Vision Map



Severity	Ticket ID	Last Modification Time	Root ...	Root Event Time	Description	Location	Acknowledged	Creation Ti
✓	560001	31-Jul-11 15:08:57	✓	31-Jul-11 14:53:26	CPU utilization le...	R1	No	31-Jul-11 1

The icon represents the device category (Ethernet Switch, Router, and so on) or the device family (Cisco Catalyst 3750 Metro Series Switches, Cisco 12000 Series Routers, and so on) to which the device belongs.

When you create a U-VNE, you define its “device type”. This determines which name and icon will be used to represent the new device in the Prime Network maps and tables. The device type definition includes providing a unique name for the device, and specifying the device category and the device family to which the device belongs. See Defining Device Types to Determine the GUI Representation of VNE Drivers, page 18-13.

Advantages and Limitations of U-VNEs

U-VNEs enable you to manage additional device types and to manage additional traps and syslogs. However, U-VNEs are not as powerful as developed VNE drivers for the following reasons:

- Dependence on U-VNE templates—If the U-VNE references a template, it will support only the NE instrumentation for device discovery that is defined in the selected U-VNE template.
- Standard MIB NE instrumentation support—NE instrumentation is based on the most commonly used subset of standard MIB-II MIBs which some device manufacturers might alter; information from the proprietary implementation of third-party device vendors is not available through the use of standard MIBs.
- Limited fault management—Supports traps for U-VNEs that were created using the Generic U-VNE template; by design, syslogs are not supported by the Generic U-VNE template. Otherwise, developed VNEs and U-VNEs are presented with almost the same limitations and capabilities:

  - For events that were added via the VCB, correlation is done based on the DC key (correlation using network flow is not supported). The event is correlated to service events with the same source, but users cannot customize correlation parameters such as weight, correlation delay, and so on. In addition, events added via the VCB cannot be root cause events.
  - For factory-defined events, capabilities are the same for developed VNEs and U-VNEs, provided that the U-VNE supports the technology.
- Limited path tracing—U-VNEs that were created using the Generic U-VNE template provide limited support for path tracing. If the U-VNE does not support routing, network paths that traverse the U-VNE will stop there. Therefore, the result of the trace will show only the path to or from the U-VNE. The exact behavior depends on the technologies used by the device, the level of adherence to standard MIB support, and the location of the U-VNE on the path. We recommend that you keep this limitation in mind when considering whether to use a U-VNE to manage a particular device, especially a device in the network core.

  Note A U-VNE created via cloning will have the same capability as the device from which it was cloned.

- Limited topology support—U-VNEs support the Ethernet topology, including its underlying physical topology, on Ethernet ports. However, dynamic topology discovery is limited to U-VNEs on the network edge that are connected to a developed VNE. In addition, U-VNEs support IP topologies on high-level data link control (HDLC) and serial ports under the following conditions:
  - Point-to-point links
  - No duplicate IP addresses
  - Same IP subnet
- Limited trap support—U-VNE trap support is limited to those listed in the specific U-VNE template.

Note For a side-by-side comparison of generic VNEs, developed VNEs, and U-VNEs, see Generic VNE, U-VNE, and Developed VNE Comparison, page 18-5.

Generic VNE, U-VNE, and Developed VNE Comparison

Table 18-1 compares the features available in generic VNEs, U-VNEs, and developed VNEs. The information in Table 18-1 is for comparative purposes only. The support provided by each U-VNE template is on a best-effort basis and might vary from template to template.

Table 18-1 Comparison of VNE Types

Feature	Generic VNEs	U-VNEs	Developed VNEs
Customization (using the VCB)	Partial (events)	Yes	Yes (modules and events)
Fault Analysis	Support for standard traps that are supported by regular Generic VNEs, and can be extended to support proprietary traps and syslogs. Refer to the *Cisco Prime Network 3.8 Reference Guide* for a full list of supported traps.	Standard and proprietary traps and syslogs that are: • Defined in the U-VNE template (when a U-VNE is added using a template) • Supported by the device from which the U-VNE was cloned (when a U-VNE is added by cloning) **Note** By design, the GenericUVNE template supports only standard traps and no syslogs	Standard and proprietary traps and syslogs
MIB support	Standard MIB-II	Standard MIB-II and private MIBs for: • Physical inventory discovery, depending on the selected U-VNE template • Importing traps to add event recognition	Standard MIB-II and private MIBs

Table 18-1 Comparison of VNE Types (continued)

Feature	Generic VNEs	U-VNEs	Developed VNEs
CLI or XML over Telnet is used to query NE	No	• For a clone-based U-VNE, the protocol used to query is inherited from cloned-from VNE driver • For a template-based U-VNE, the protocol used to query depends on what is defined in the U-VNE template	Yes
Multivendor	Yes	Yes	Yes
NBI	Yes	Yes	Yes
Discovery	Yes	Yes (CDP is supported for Cisco devices)	Yes
Physical Inventory, Containment	No	Yes, depending on the selected template and the NE instrumentation	Yes, depending on the NE instrumentation
Logical Inventory and Technologies	Routing table, ARP table, default bridge, IP interfaces	Routing table, ARP table, default bridge, IP interfaces[1]	Depends on Prime Network supported technologies and NE instrumentation; see the *Cisco Prime Network 3.8 Reference Guide*

Table 18-1 Comparison of VNE Types (continued)

Feature	Generic VNEs	U-VNEs	Developed VNEs
Alarm Correlation (root cause analysis)	No	Yes for factory-defined events only, provided that the related technologies are supported by the U-VNE[2] Event associated to managed element by default **Note** For events that were added via the VCB, correlation is done based on the DC key (correlation using network flow is not supported). The event is correlated to service events with the same source, but users cannot customize correlation parameters such as weight, correlation delay, and so on. In addition, events added via the VCB cannot be root cause events.	Yes (for factory-defined events) **Note** For events that were added via the VCB, correlation is done based on the DC key (correlation using network flow is not supported). The event is correlated to service events with the same source, but users cannot customize correlation parameters such as weight, correlation delay, and so on. In addition, events added via the VCB cannot be root cause events.
NE Image Management	No	No	Yes
Configuration Archival	No	No	Yes
Path Tracing	No	Only physical and Ethernet are supported	Yes
Topology Discovery	Physical and Ethernet	• For template-based U-VNEs, only physical and Ethernet are supported—Supported technologies are documented in the templates • For cloned U-VNEs, depends on the VNE from which it was cloned[1]	Full

[1] For a U-VNE driver cloned from a developed VNE driver, the logical inventory, technology and topology discovery instrumentation usage are inherited from the developed VNE driver.

2. For example, link up/down correlates to the card-out service alarm, while MPLS TE Tunnel down is associated to managed element and is not correlated to other alarms.

Creating a U-VNE

The following sections describe the steps involved in creating a U-VNE:

- Creating a U-VNE—A Step-by-Step Example, page 18-8
- Planning for Creating a U-VNE, page 18-9
- Creating a U-VNE - Procedure, page 18-10
- Testing and Certifying U-VNEs, page 18-36

Creating a U-VNE—A Step-by-Step Example

The following example shows how to enable Prime Network to manage an unsupported device by creating a U-VNE. For the purposes of this example, we will create a U-VNE to enable support for a Linux server. The U-VNE will be based on the generic U-VNE template.

Step 1 Obtain the sysOID for the NE that you want to manage by querying the device using, for example, the snmpget command:

```
snmpget -c SNMPCommunityString IPAddress .1.3.6.1.2.1.1.2.0
```

Step 2 Confirm that Prime Network does not already support the sysOID. Do one of the following:

- In the VCB GUI, filter the VNE Drivers table by the sysOID. If no driver is found, it is not supported. See Filtering the VCB Tables, page 17-3 for information about the filter.
- Search for the sysOID in the output of the following command on the Prime Network gateway:

```
vcb uvne view -sysoid all | grep sysOID
```

If the value that you entered for *sysOID* is not listed, it is not supported.

Step 3 Decide on the approach you want to take to add the U-VNE. See Approaches to Creating a U-VNE, page 18-2. For this example, the new U-VNE will be based on the generic U-VNE template.

Step 4 Open Prime Network Administration.

Step 5 Click **Tools** > **VNE Drivers.**

The VNE Drivers tab lists all the VNE drivers.

Step 6 Click **Add Row**. A row is added to the table, enabling you to enter values in the relevant fields or select from the drop-down lists.

Step 7 Enter the sysOID of the new device in the SysOID field.

Step 8 Specify the name that will represent the device in the Prime Network GUI by creating a new device type, as follows:

- In the Device Type field, click the down arrow to display the Device Type Selector.
- Click the Tools icon in the upper right area of the Device Type Selector and select **Add**.
- In the Device Type field, enter the device name that will represent the device in the Prime Network maps and element tables. We recommend that you prefix the device type name with the network element vendor name for easy identification.

- In the Category field, select the category to which the device belongs. In this case, select **Server**.
- Click **OK**. The new Device Type name will appear in the Device Type field.

Step 9 In the Type field, select **User Defined VNE - by Template**.

Step 10 In the Cloning Reference field, select **GenericUVNE**.

Step 11 Click **Save**.

Step 12 Prepare to test; see Setting Up the Test Environment, page 18-37.

Step 13 Complete tests and certify the customization; see Procedures forTesting and Certifying U-VNEs, page 18-39.

Planning for Creating a U-VNE

Before you create a U-VNE, you need to have a good understanding of the device type for which you want to enable support in When creating a U-VNE to enable support for an additional device type, you have several options, as described in Approaches to Creating a U-VNE, page 18-2. Before you can decide which approach to take, the initial steps of Creating a U-VNE—A Step-by-Step Example, page 18-8 describe obtaining the sysObjectID for a device and confirming that it is not supported in Prime Network before you start customizing. While you are in the planning phase, here are some additional steps to assist you in determining the approach to take when creating a U-VNE:

1. Perform research, investigating the NE that you want to add. Gather information from sources such as the following:
   - Vendor- or device- specific documentation or website for details.
   - Element management systems (EMS) or other management systems for property or detail views.
   - Device or network experts.
   - Advanced services or CA personnel who deal with Prime Network.
   - VCB commands that enable you to view details for similar VNEs, modules, and events—See Viewing Existing Device Types, page 18-14,
   - *Cisco Prime Network 3.8 Reference Guide* to compare the supported device scope with the device scope that you need to manage. If gaps exist—in the areas of recognizing device type, hard cards, device maintenance software, SNMP traps, or syslogs—use the VCB to extend the VNE drivers.
2. Evaluate whether you can clone. To create a U-VNE by cloning from an existing VNE driver, you must identify a suitable VNE driver.

   For the technologies that are modeled for a particular scheme under a particular device series, see the *Cisco Prime Network 3.8 Reference Guide*. After you identify the device series, use the **vcb uvne view** command to obtain the sysoid of a particular device type as follows:

   ```
   vcb uvne view -sysoid all -user root -password admin
   ```

   Note In the output of the command, ignore any entries for which the Device Family column contains base or da.

3. If the cloning approach is not a good fit for the device that you want to discover, add the U-VNE by referencing a template. Perform analysis, looking for gaps between the information that you gathered and the information in the templates; see U-VNE Templates, page 22-1. Evaluate the U-VNE templates to determine which template most closely matches the data that you gathered.
4. Obtain the sysOID for the NE that you want to manage by doing any of the following:

- Query the device using, for example, the snmpget command

```
snmpget -c SNMPCommunityString IPAddress .1.3.6.1.2.1.1.2.0
```

- Use any MIB browser and query for the OID 1.3.6.1.2.1.1.2.0
- Look into the CISCO-PRODUCTS-MIB if you are investigating a Cisco product. The following URL opens the Cisco SNMP Object Navigator to the CISCO-PRODUCT-MIB:

  http://tools.cisco.com/Support/SNMP/do/BrowseMIB.do?local=en&mibName=CISCO-PRODUCTS-MIB

5. Confirm that Prime Network does not already support the sysOID by doing one of the following:
   - Search for the sysOID in the output of the following command:

```
vcb uvne view -sysoid all | grep sysOID
```

If the value that you entered for *sysOID* is not listed, it is not supported.

Creating a U-VNE - Procedure

To add a new VNE driver using the Prime Network VCB GUI:

Step 1 Click **Tools** > **VNE Drivers.**

The VNE Drivers tab lists all the VNE drivers.

Step 2 Click **Add Row**. A row is added to the table, enabling you to enter values in the relevant fields or select from the drop-down lists.

Define the following parameters for the new VNE driver:

Parameter	Description
SysOID	The unique object identifier of the VNE driver. For more details on how to obtain the SysOID for the U-VNE that you want to manage, see Planning for Creating a U-VNE, page 18-9
Device Type	The name that will represent the device in the Prime Network GUI. Click the down arrow in the Device Type field to display the Device Type Selector which contains a list of existing device types for selection. In the Device Type Selector, you can: • Select a device type. • Create a new device type by clicking on the Tools icon and then selecting **Add**. We recommend that you prefix the device type name with the network element vendor name for easy identification. The new device type will be added to the list and you will be able to select it. • Toggle between the tree-structure view and the standard view. • Click **Back** to go to the previous view.
Type	The type of the VNE driver: • Prime Network Default • User-Defined VNE- by Template • User-Defined VNE- by SysOID • User-Defined VNE- by Software Version • User-Defined VNE- by Device Family You can use these options to define if you want to create a VNE driver based on device family or software version. See Approaches to Creating a U-VNE, page 18-2 for more information.
Cloning Reference	The item from which you want to clone the new VNE.
Overriding System Default	This parameter is populated automatically by the system: Yes—If the U-VNE was created by editing a system default VNE. No—If the U-VNE was created from scratch and not by editing a system default VNE.

Step 3 Click **Save.**

Step 4 Test and certify the U-VNE. See Testing and Certifying U-VNEs, page 18-36.

Managing Existing VNE Drivers

The Prime Network VCB GUI enables you to manage VNE drivers in your network. You can perform the following actions to manage the VNE Driver:

- Editing a VNE Driver, page 18-12
- Deleting a VNE Driver, page 18-12

- Viewing Modules in a Selected VNE Driver, page 18-13
- Defining Device Types to Determine the GUI Representation of VNE Drivers, page 18-13
- Adding Support for a New Software Version, page 18-18
- Enabling Support for Additional Events, page 20-2

Viewing Existing VNE Drivers

To view VNE drivers:

Step 1 In the VCB tool, click **Tools** > **VNE Drivers**.

For more details of the selected VNE driver, click the arrow icon in each row.

Editing a VNE Driver

Note To edit an existing row in the VNE Driver table, double-click or click on the row you want to edit.

To edit an existing VNE driver using the Prime Network VCB GUI:

Step 1 Log into the Prime Network VCB GUI using the login credentials.

On successful login, Prime Network VCB GUI displays the home page.

Step 2 Click **Tools** > **VNE Drivers**.

The VNE Drivers tab lists all the VNE drivers.

Step 3 Select a VNE driver from the list.

Step 4 Click **Edit**.

Note You will receive a warning message while trying to edit the Prime Network Default VNE driver. Click **Ok** to continue and **Cancel** to view the VNE Drivers window.

Step 5 Click **Save** to save the changes.

Step 6 Click **Cancel** to view the VNE Drivers window.

Step 7 Restart the AVM, that manages the devices.

Step 8 Test the VNEs.

Step 9 Restart the Prime Network server.

Deleting a VNE Driver

To delete an existing VNE driver using the Prime Network VCB GUI:

Step 1 Log into the Prime Network VCB GUI using the login credentials.

On successful login, Prime Network VCB GUI displays the home page.

Step 2 Click **Tools** > **VNE Drivers**.

The VNE Drivers tab lists all the VNE drivers.

Step 3 Select a VNE driver from the VNE Drivers list.

Step 4 Click **Delete**.

Note You cannot delete the default VNE Drivers.

Step 5 A window appears prompting you to confirm deletion of selected VNE driver. Click **OK** to continue deleting the device type.

Step 6 Click **Cancel** to view the VNE Drivers window.

Step 7 Restart the AVM, that manages the devices.

Step 8 Test the VNEs.

Step 9 Restart the Prime Network server.

Viewing Modules in a Selected VNE Driver

To view modules in an existing device type using the Prime Network VCB GUI:

Step 1 Log into the Prime Network VCB GUI using the login credentials.

On successful login, Prime Network VCB GUI displays the home page.

Step 2 Click **Tools** > **VNE Drivers**.

The VNE Drivers tab lists all the VNE drivers.

Step 3 Click the arrow to the left of the required VNE driver to expand its display.

Step 4 Click the Module Group link. The displayed dialog box lists all the modules in the module group.

Note For details on how to add new module and manage existing modules, see Chapter 19, “Enabling Support for Additional Modules”.

Defining Device Types to Determine the GUI Representation of VNE Drivers

You can perform the following actions to manage device types:

- Viewing Existing Device Types, page 18-14
- Adding a New Device Type, page 18-14
- Editing a Device Type, page 18-15

- Deleting a Device Type, page 18-16

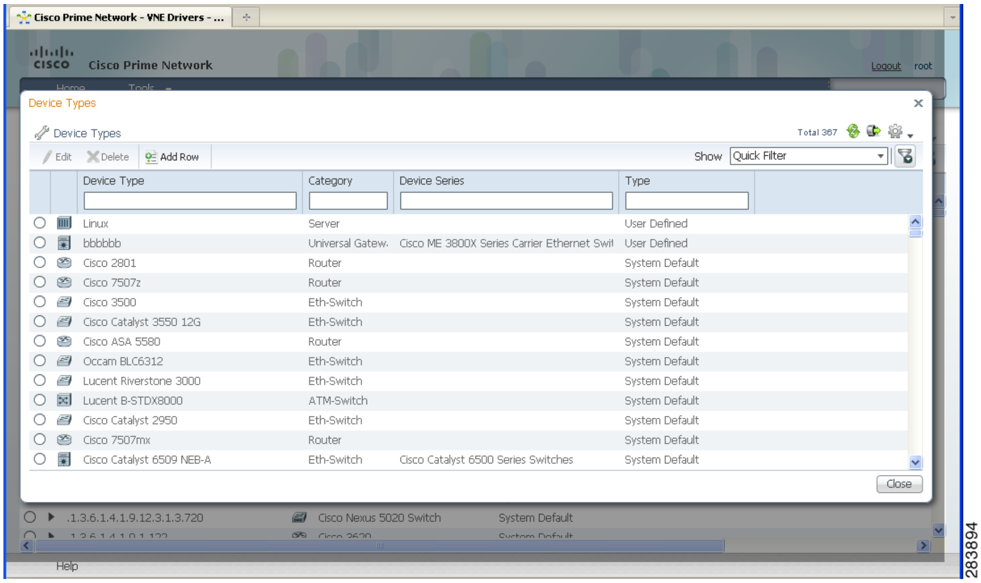
Viewing Existing Device Types

To view a list of device types:

Step 1 In the VCB window, click **Tools** > **VNE Drivers**.

Step 2 Click **Configure Device Types**. The Device Types window appears.

Figure 18-2 Device Types



		Device Type	Category	Device Series	Type
○		Linux	Server		User Defined
○		bbbbbb	Universal Gatew.	Cisco ME 3800X Series Carrier Ethernet Swit	User Defined
○		Cisco 2801	Router		System Default
○		Cisco 7507z	Router		System Default
○		Cisco 3500	Eth-Switch		System Default
○		Cisco Catalyst 3550 12G	Eth-Switch		System Default
○		Cisco ASA 5580	Router		System Default
○		Occam BLC6312	Eth-Switch		System Default
○		Lucent Riverstone 3000	Eth-Switch		System Default
○		Lucent B-STDX8000	ATM-Switch		System Default
○		Cisco Catalyst 2950	Eth-Switch		System Default
○		Cisco 7507mx	Router		System Default
○		Cisco Catalyst 6509 NEB-A	Eth-Switch	Cisco Catalyst 6500 Series Switches	System Default

○ ▶ .1.3.6.1.4.1.9.12.3.1.3.720 Cisco Nexus 5020 Switch System Default

The Device Types window provides a table listing all the available device types. The table columns show the following information for each device type:

- **Device Type**—The name that represents the VNE driver in the Prime Network GUI.
- **Category**—The type of device the VNE driver represents. For example, Router, Eth-Switch, and so on.
- **Device Series**—The device family. For example, Cisco12000 Series Routers.
- **Type**—Whether the device type is provided with the system (system default) or user-defined.

Adding a New Device Type

If none of the existing device types is a suitable representation of your U-VNE, you can create a new device type. You can add a new device type in one of the following ways:

- From the VNE drivers window, during the VNE driver creation/editing process. See Creating a U-VNE - Procedure, page 18-10.

- From the Device Types window, as described below.

To add a new device type:

Step 1 In the VCB window, click **Tools** > **VNE Drivers**.

Step 2 Click **Configure Device Types**. The Device Types window appears.

Step 3 Click **Add Row**. A row is added to the device types table, enabling you to define the parameters of the new device type.

Step 4 Define the following parameters:

Table 18-2 Add New Device Type

Parameter	Description
Device Type	The name that will represent the device in the Prime Network GUI, in maps and tables. We recommend that you prefix the device type name with the network element vendor name for easy identification.
Category	The type of device, for example, router, switch, and so on.
Device Series	The device family to which the device type belongs, for example, Cisco 10000 Routers. This will determine which icon will be used to represent the device.
Type	Since you are adding a device type that is not provided as a system default, this field is automatically populated as "User-Defined".

Step 5 Click **Save**.

Editing a Device Type

You can edit user-defined device types. System default device types cannot be edited.

To edit an existing device type:

Step 1 In the VCB window, click **Tools** > **VNE Drivers**.

Step 2 Click **Configure Device Types**. The Device Types window appears.

Step 3 Double-click on a device type or select a device type and Click **Edit**.

Step 4 Change the device type parameters as required. See Adding a New Device Type, page 18-14 for information about the parameters.

Step 5 Click **Save**.

Deleting a Device Type

You can delete user-defined device types. System default device types cannot be deleted.

To delete a device type:

Step 1 In the VCB window, click **Tools** > **VNE Drivers**.

Step 2 Click **Configure Device Types**. The Device Types window appears.

Step 3 Select a device type from the device types list.

Step 4 Click **Delete**. You are prompted to confirm that you want to delete the device type. Click **OK** to continue deleting the device type.

Enabling Support for Additional Software Versions

The VCB enables you to enable support for additional software versions and to manage currently supported software versions. You can perform the following actions to manage the software version:

- Adding Support for a New Software Version, page 18-18
- Editing a Software Version, page 18-18
- Deleting a Software Version, page 18-19

Viewing the Software Versions

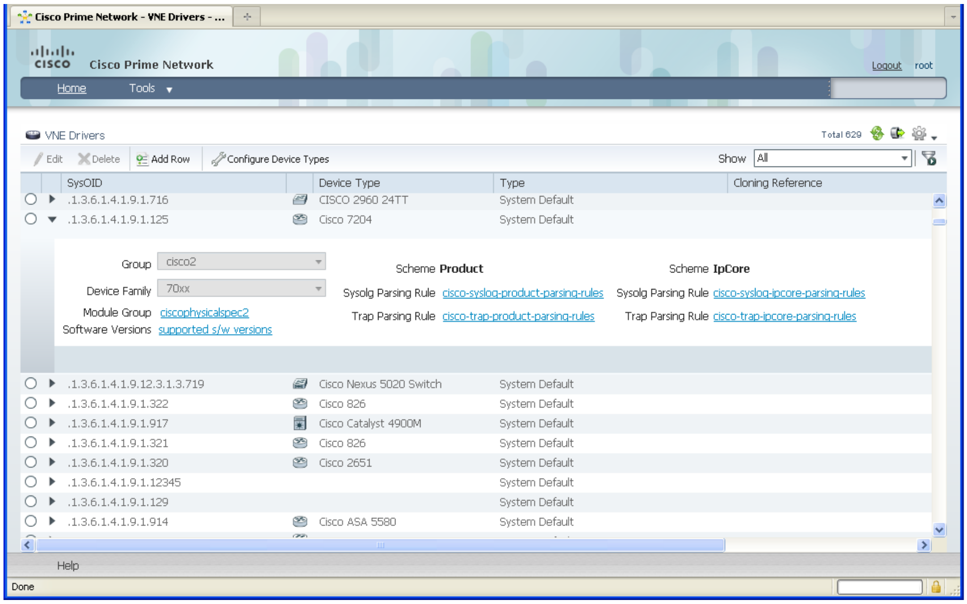
To view the software versions of a selected VNE driver in Prime Network VCB GUI:

Step 1 Log into the Prime Network VCB GUI application.

Step 2 Click **Tools** > **VNE Drivers**.

Step 3 Click the arrow next to the required VNE driver to expand its display.

Figure 18-3 Accessing the Link to Supported Software Versions



	SysOID	Device Type	Type	Cloning Reference
○	.1.3.6.1.4.1.9.1.716	CISCO 2960 24TT	System Default	
○	.1.3.6.1.4.1.9.1.125	Cisco 7204	System Default	

Group: cisco2
Device Family: 70xx
Module Group: ciscophysicalspec2
Software Versions: supported s/w versions

Scheme **Product**
Sysolg Parsing Rule: cisco-syslog-product-parsing-rules
Trap Parsing Rule: cisco-trap-product-parsing-rules

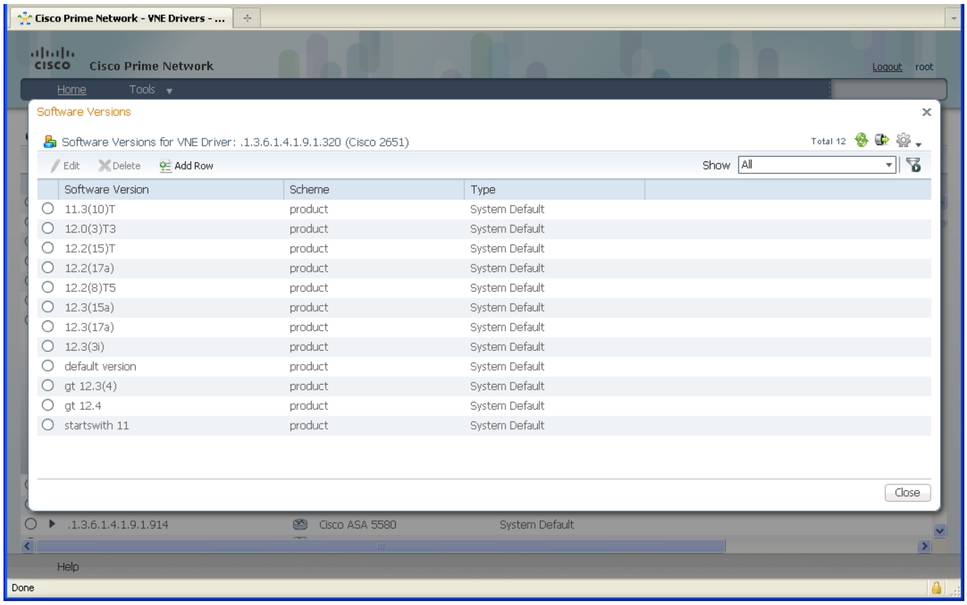
Scheme **IpCore**
Sysolg Parsing Rule: cisco-syslog-ipcore-parsing-rules
Trap Parsing Rule: cisco-trap-ipcore-parsing-rules

	SysOID	Device Type	Type	Cloning Reference
○	.1.3.6.1.4.1.9.12.3.1.3.719	Cisco Nexus 5020 Switch	System Default	
○	.1.3.6.1.4.1.9.1.322	Cisco 826	System Default	
○	.1.3.6.1.4.1.9.1.917	Cisco Catalyst 4900M	System Default	
○	.1.3.6.1.4.1.9.1.321	Cisco 826	System Default	
○	.1.3.6.1.4.1.9.1.320	Cisco 2651	System Default	
○	.1.3.6.1.4.1.9.1.12345		System Default	
○	.1.3.6.1.4.1.9.1.129		System Default	
○	.1.3.6.1.4.1.9.1.914	Cisco ASA 5580	System Default	

Step 4 Click **supported s/w versions** in the Software Versions field.

The list of currently supported software for the selected VNE driver is displayed.

Figure 18-4 Software Versions



Software Versions for VNE Driver: .1.3.6.1.4.1.9.1.320 (Cisco 2651)

	Software Version	Scheme	Type
○	11.3(10)T	product	System Default
○	12.0(3)T3	product	System Default
○	12.2(15)T	product	System Default
○	12.2(17a)	product	System Default
○	12.2(8)T5	product	System Default
○	12.3(15a)	product	System Default
○	12.3(17a)	product	System Default
○	12.3(3i)	product	System Default
○	default version	product	System Default
○	gt 12.3(4)	product	System Default
○	gt 12.4	product	System Default
○	startswith 11	product	System Default

	SysOID	Device Type	Type
○	.1.3.6.1.4.1.9.1.914	Cisco ASA 5580	System Default

You can perform the following actions from the Software Versions window:

- Add support for a new software version—Click **Add Row**. See Adding Support for a New Software Version, page 18-18.
- Edit a software version—Select a software version in the list and click **Edit**.
- Delete a software version—Select a software version in the list and click **Delete**.

Adding Support for a New Software Version

To add support for a new software version:

Step 1 In the VCB window, click **Tools** > **VNE Drivers**.

Step 2 Click the arrow next to the VNE driver for which you want to add the new software version.

Step 3 Click **supported s/w versions** in the Software Versions field.

The list of currently supported software for the selected VNE driver is displayed.

Step 4 Click **Add Row**.

Step 5 Enter the values for the following parameters:

Table 18-3 Add New Software Version

Parameter	Description
New Software Version	The name of the software version to be supported.
Scheme	The scheme of the new VNE, either Product or IpCore.
Supported Software Version	The already supported software version to be cloned in order to add support for the new software version.

Step 6 Click **OK**.

Editing a Software Version

Note You cannot edit system default software versions.

To edit a software version:

Step 1 In the VCB window, click **Tools** > **VNE Drivers**.

Step 2 Click the arrow next to the VNE driver for which you want to add the new software version.

Step 3 Click **supported s/w versions** in the Software Versions field.

The list of currently supported software for the selected VNE driver is displayed.

Step 4 Select the software version you want to edit.

Step 5 Click **Edit**.

Step 6 Edit the software version as required.

Step 7 Click **OK**.

Deleting a Software Version

Note You cannot delete system default software versions.

To delete a software version:

Step 1 In the VCB window, click **Tools** > **VNE Drivers**.

Step 2 Click the arrow next to the VNE driver for which you want to add the new software version.

Step 3 Click **supported s/w versions** in the Software Versions field.

The list of currently supported software for the selected VNE driver is displayed.

Step 4 Select the software version you want to delete.

Step 5 Click **Delete**.

Step 6 Click **OK.** A confirmation dialog box is displayed.

Step 7 Click **OK**.

Using the CLI to Create and Manage U-VNEs

You have the option to use the CLI to perform VCB customizations. Use the CLI if you are an advanced VCB user or if the current VCB GUI does not support the customization you need to perform.

This section describes the CLI commands that can be used to create and manage U-VNEs and device types, as follows:

- Introductory Command Reference and Global Options, page 18-20
- Commands for Adding and Managing U-VNEs, page 18-23
- Commands for Adding and Managing Device Types, page 18-31

Introductory Command Reference and Global Options

This command reference introduces the **vcb** command and gives details for the **vcb sitechanges** command which you must use to deploy, maintain, or remove any VCB customizations. This command reference includes the following sections:

- vcb, page 18-20
- vcb sitechanges, page 18-22

vcb

The **vcb** command is the wrapper for all VCB-related commands.

Synopsis

The syntax of the **vcb** command is as follows:

```
{uvne|module|pluggablemodule|event|eventpattern|eventparsingrule|devicetype}
{add|view|modify|delete} args… [-help ] [-debuglevel {error|warn|info|debug}] [-logfile
logfile] -user username -password password
```

```
vcb {eventarg} {view} [-help ] [-debuglevel {error|warn|info|debug}] [-logfile logfile]]
-user username -password password
```

```
vcb {sitechanges} {view|export|delete} [-help ] [-debuglevel {error|warn|info|debug}]
[-logfile logfile]] -user username -password password
```

For information—descriptions, options, and usage—about specific commands, see specific command references:

- Commands for Adding and Managing U-VNEs, page 18-23
- Commands for Adding and Managing Device Types, page 18-31
- Commands for Managing Modules, page 19-10
- Commands for Managing Pluggable Modules, page 19-18
- Using the CLI to Customize Events, page 20-13

Global Command Options

Table 18-4 describes the global options and arguments that are common to all **vcb** commands and subcommands.

Table 18-4 Global Options and Arguments—vcb

Option ***Argument***	**Description**
help	Displays online help about the command.Use this option for each subcommand.
debuglevel	Determines which messages are logged based on the message severity. Valid values are: • error • warn (default) • info • debug For example, if debuglevel is set to warn, all warning and error messages are saved to the log file.
logfile *logfile*	Logs the CLI output to the file specified in the *logfile* argument.
userdefined	Displays the registrations that have been added to the site.xml file using the VCB or any other tool.
user *username*	BQL username.
password *password*	BQL password.

Description

Use the **vcb** command to create and update VNE registry configuration files. Use the **vcb** command to add, delete, view, and modify information in these files, based on the subcommands that are used with it.

The **vcb** command performs the following high-level operations:

- Executes Prime Network administrator-level operations. It also provides a service to handle authorization errors. VCB users must have the Prime Network admin user role and associated privileges in order to perform registry configuration commands.
- Verifies that the expected version of Prime Network is running.
- Provides help information for each command.
- Logs all internal commands.
- Centralizes error handling for exceptions.

General Error Codes

Table 18-5 General Error Codes—vcb

Code	Description
0	Signifies OK.
10	Operation not permitted. An attempt was made to perform an operation limited to processes with appropriate privileges or to the owner of a file or other resources.
11	User does not have Prime Network admin privileges.
20	No such file or directory. A component of a specified pathname did not exist, or the pathname was an empty string. This might occur when the argument is a DSP or to a registry hive.
30	Failed to connect to Prime Network server.
40	I/O error. Some physical input or output error occurred.
50	Invalid argument. Some invalid argument was supplied.
60	Incompatible Prime Network version of VCB; expected $vcb_expected_version$ and found Prime Network $ana_running_version$.
70	Bad procedure for program. A Remote Procedure Call (RPC) call was attempted for a procedure which does not exist in the VCB program.
80	Function not implemented. Attempted a system call that is not available on this system.

vcb sitechanges

The **vcb sitechanges** command affects all extensions in the local site.xml file—displaying, exporting, or deleting them—whether the extensions were created by the VCB or by other Prime Network utilities.

Synopsis

```
vcb sitechanges {view|export|delete} [-help ] [-debuglevel {error|warn|info|debug}]
[-logfile logfile]] -user username -password password
```

Description

Use the **vcb sitechanges** command to view all customizations that were made to the site.xml registry file whether by the VCB or by other Prime Network utilities. Use the **vcb sitechanges** command also to write all customizations to script files that enable you to import changes to another system or to delete all changes, returning the system to factory default.

Because the VCB does not differentiate between changes made by the VCB and changes made by other Prime Network utilities, the **vcb sitechanges export** and **vcb sitechanges delete** commands create script files to enable you to inspect commands before you execute them. Edit the script files before running them to ensure that only the customizations that you are interested in are acted upon.

Commands for Adding and Managing U-VNEs

Use the following **vcb uvne** commands to create, view, modify, and delete U-VNEs:

- vcb uvne add, page 18-23
- vcb uvne view, page 18-25
- vcb uvne modify, page 18-28
- vcb uvne delete, page 18-30

vcb uvne add

The **vcb uvne add** command creates a U-VNE by associating a U-VNE template with the device type associated with the given sysOID, or by cloning from an existing device or device family.

Synopsis

```
vcb uvne add -sysoid sysoid -template template name -group template filename [-devicetype device type name] -user username -password password

vcb uvne add -sysoid sysoid -clonesysoid clone_sysoid -user username -password password

vcb uvne add -sysoid sysoid -clonedevicefamily devicefamily [-devicetype device type] -user username -password password

vcb uvne add -sysoid sysoid -softwareversion software-version -clonesoftwareversion clone-softwareversion -clonedevicefamily devicefamily -scheme <product|ipcore > [-devicetype device type name] -user username -password password
```

Description

The **vcb uvne add** command creates a U-VNE-driver for the device type associated with the given sysOID using the specified U-VNE template. Creating a U-VNE-driver enables the Auto Detect feature in Prime Network to associate a device with this sysOID with the VNE-driver implementation defined by the template.

This command creates a separate registry configuration for the U-VNE. Any configuration setting given in the command parameters affect this copy.

Note This command does not create the device category for the device. For more information, see vcb devicetype add, page 18-31.

Usage Examples

Example 1

```
vcb uvne add -sysoid .1.2.3.4 -template GenericUVNE -group uvne -user root -password admin
```

Enables discovery of the device with sysOID 1.2.3.4 using the GenericUVNE template, which is located in group uvne-product.

The command does not create any device attributes for the newly added device. To assign the device a user-friendly name and the correct device category, use the **-devicetype** option (see the "vcb devicetype add" section on page 18-31).

Example 2

```
vcb uvne add -sysoid .1.2.3.4 -clonesysoid .1.3.6.1.4.1.9.1.108 -user root -password
admin
```

Enables discovery of a new device (with sysOID .1.2.3.4) that points to registration—scheme and instrumentation commands—of the already supported sysOID .1.3.6.1.4.1.9.1.108.

(To create a list of already supported VNEs, see vcb uvne view, page 18-25.)

Example 3

```
vcb uvne add -sysoid .1.2.3.4 -softwareversion 12.6(2) -clonesoftwareversion 12.0(23)S2
-clonedevicefamily 100xx -scheme product -user root -password admin
```

Adds support for a new device (with sysOID .1.2.3.4), that points to registrations—scheme and instrumentation commands—of the already supported software version 12.0(23)S2 for the device family 100xx. (For a list of already supported software versions, see vcb uvne view, page 18-25.)

Example 4

```
vcb uvne add -sysoid .1.2.3.4 -clonedevicefamily 100xx -user root -password admin
```

Enables discovery of a new device (with sysOID .1.2.3.4), that points to registrations—scheme and instrumentation commands—for the device family 100xx.

Options

Table 18-6 Options and Arguments—vcb uvne add

Option *Argument*	Description
sysoid *sysoid*	The sysObject ID of the device for which to create a U-VNE-driver using the implementation defined in the U-VNE template. **Note** Each sysOID value in the system must be unique.
template *template name*	The name of the U-VNE template from which to create the U-VNE-driver. See U-VNE Templates, page 22-1. **Note** This option is mutually exclusive with the **-clonesysoid** and **-clonedevicefamily** options.
group *template filename*	The name of the file in which the U-VNE template is located. U-VNE templates are located in the uvne-product file. **Note** Use this option with the **-template** option only.
devicetype *device type name*	(Optional) The U-VNE devicetype name. If not specified, the device type is: • Defined as Unknown when the option is not specified. • Inherited from the device or device family when you use the **-clonesysoid** or **-clonedevicefamily** options. The device type name appears in the UI and is associated with a category; its category determines which icon is displayed, and other presentation aspects are derived from this reference. To use a new device type name, first add it using the **vcb devicetype add** command. For more information, see vcb devicetype add, page 18-31.
clonesysoid *clone-sysoid*	The sysObject ID of an already supported VNE. To view sysOIDs for supported VNEs, use the **vcb uvne view -sysoid** all command.

Table 18-6 Options and Arguments—vcb uvne add (continued)

Option *Argument*	Description
softwareversion *software-version*	The software version that you want to add for a U-VNE. To obtain the software version string, use the **show version** command.
clonesoftwareversion *clone-softwareversion*	An already supported software version that you want to clone. The software version must already be supported for a particular device family. For a list of supported device families, use the command **vcb uvne view –scheme** <*ipcore*|*product*> **–user** *user* **-password** *password*. For more information, see vcb uvne view, page 18-25.
clonedevicefamily *devicefamily*	A supported device family. For a list of supported device families, use the command **vcb uvne view –scheme** <*ipcore*|*product*> **–user** *user* **-password** *password*. For more information, see vcb uvne view, page 18-25. **Note** This option is mutually exclusive with the **-template** option.

Note For the list of global options, see Global Command Options, page 18-21.

Error Codes

Table 18-7 Error Codes—vcb uvne add

Code	Description
101	The sysOID already exists and is already modeled as a VNE.
102	U-VNE template file not found.
103	No such template name in the templates file.
104	No such device type configuration exists.

Note For the list of general VCB error codes, see General Error Codes, page 18-22.

vcb uvne view

The **vcb uvne view** command returns an existing U-VNE configuration.

Synopsis

```
vcb uvne view -sysoid {sysoid | all} -user username -password password
[-userdefined][-detail]
```

```
vcb uvne view -template {template name| all} -group <template group name> -user username
-password password
```

Description

Use the **vcb uvne view** command to:

- Display information based on the specified sysOID.
- Display available templates found in the specified group.

Usage Examples

Example 1

```
vcb uvne view -sysoid .1.2.3.4 -user root -password admin -userdefined
```

Returns the configuration of the VNE with the sysOID 1.2.3.4, including the U-VNE template (or the device family for a developed VNE). If device-type associations are defined (using the **vcb devicetype add** command), these associations are also displayed.

Example 2

```
vcb uvne view -sysoid all -user root -password admin
```

Returns all the sysoid supported in Prime Network.

Example 3

```
vcb uvne view -sysoid all -user root -password admin -detail
```

Returns all known configured sysOIDs (regular VNEs) and also this command would show following additional informations

- Module Spec Name associated with the sysoid.
- Syslog and Trap Parsing rule name associated with the sysoid

Here is an example of the output:

```
SysOid......:.9.7.6.5.4.3
DeviceFamily:76xx
CloneSysOid.:.1.3.6.1.4.1.9.1.863
Scheme........:product
Module Spec.........:ciscophysicalspec2
Trap Parsing Rule...:cisco-trap-product-parsing-rules
Syslog Parsing Rule.:cisco-syslog-product-parsing-rules
Scheme........:ipcore
Module Spec.........:ciscophysicalspec2
Trap Parsing Rule...:cisco-trap-ipcore-parsing-rules
Syslog Parsing Rule.:cisco-syslog-ipcore-parsing-rules

SysOid.....:.3.4.5.6.7
Template...:GenericUVNE
Group......:uvne
Device Type:UNKNOWN
Scheme........:product
Module Spec.........:N/A
Trap Parsing Rule...:genericuvne-trap-parsing-rules
Syslog Parsing Rule.:genericuvne-syslog-parsing-rules
```

Options

Table 18-8 Options and Arguments—vcb uvne view

Option ***Argument***	**Description**
sysoid *sysoid*	Returns the configuration of the specified VNE, including the device type, the user-friendly name, and the template (for U-VNEs). **Tip** Enter **all** as the *sysoid* to view the configuration of all sysOIDs (U-VNEs and developed VNEs) configured in the system.
group *template filename*	Returns the configuration of all sysOIDs and templates contained in the specified group. **Tip** Enter **all** as the *template filename* to view the template associations for each group in the system.
userdefined	Lists the U-Vne created through vcb command only.
details	Shows following additional information: • Module Spec Name associated with the sysoid. • Syslog and Trap Parsing rule name associated with the sysoid

Note For the list of global options, see Global Command Options, page 18-21.

Error Codes

Table 18-9 Error Codes—vcb uvne view

Code	**Description**
102	U-VNE template file not found.
103	No such template name in the templates file.
104	No such device type configuration exists.
111	The SysOID specified does not exist.

Note For the list of general VCB error codes, see General Error Codes, page 18-22.

vcb uvne modify

The **vcb uvne modify** command modifies the configuration of an existing U-VNE.

Synopsis

```
vcb uvne modify -sysoid sysoid -template template name -group template filename -user
username -password password
```

```
vcb uvne modify -sysoid sysoid -devicetype device type -user username -password password
```

```
vcb uvne modify -sysoid sysoid -clonesysoid sysoid -user username -password password
```

```
vcb uvne modify -sysoid sysoid -softwareversion softwareversion
-clonesoftwareversion clonesoftwareversion -clonedevicefamily devicefamily -scheme
vnescheme -user username -password password
```

```
vcb uvne modify -sysoid sysoid -clonedevicefamily devicefamily -user username -password
password
```

Description

Use the **vcb uvne modify** command to:

- Associate the U-VNE with another U-VNE template.
- Change the device type associated with the U-VNE.

Usage Examples

Example 1

```
vcb uvne modify -sysoid .1.2.3.4 -group uvne-product
-template cisco-chassis-mib-template -user root -password admin
```

Modifies the template of the U-VNE to the newly specified template defined in the given group.

Example 2

```
vcb uvne modify -sysoid .1.2.3.4 -devicetype CISCO_1760 -user root -password admin
```

Modifies the device type for the U-VNE with sysOID 1.2.3.4.

Example 3

```
vcb uvne modify -sysoid .1.2.3.4 -clonesysoid .1.3.6.1.4.1.9.1.108 -user root -password
admin
```

Modifies a device, with sysOID .1.2.3.4, to point to a new device family based on the clone sysoid .1.3.6.1.4.1.9.1.108.

Example 4

```
vcb uvne modify -sysoid .1.2.3.4 -softwareversion 12.6(2) -clonesoftwareversion
12.0(23)S3 -clonedevicefamily 100xx -scheme product -user root -password admin
```

Modifies support for a device with sysOID .1.2.3.4, to point to a new software version, 12.0(23)S3, for the device family 100xx.

Example 5

```
vcb uvne modify -sysoid .1.2.3.4 -clonedevicefamily 12xxx -user root -password admin
```

Modifies support for a new device, with sysOID .1.2.3.4, to point to registrations—scheme and instrumentation commands—of the device family 12xxx.

Options

Table 18-10 Options and Arguments—vcb uvne modify

Option *Argument*	Description
sysoid *sysoid*	The sysObject ID of the U-VNE-driver configuration that you want to modify.
template *template name*	The name of the U-VNE template to which the U-VNE should be associated. Use this option to modify the U-VNE template from which the U-VNE-driver derives its configuration. **Note** Using this option does not overwrite other configuration changes made with the VCB, such as user-experience attributes that are defined with the vcb devicetype add command.
group *template filename*	The name of the group that includes the U-VNE template, such as uvne-product.
devicetype *device type*	The device type associated with the U-VNE.
clonesysoid *clone-sysoid*	The sysObject ID of an already supported VNE. To view sysOIDs for supported VNEs, use the **vcb uvne view -sysoid** all command.
softwareversion *software-version*	The software version that you want to support for a U-VNE. To obtain the software version string, use the **show version** command.
clonesoftwareversion *clone-softwareversion*	An already supported software version that you want to clone. The software version must already be supported for a particular device family.
clonedevicefamily *deviceFamily*	A supported device family. For a list of supported device families, use the command **vcb uvne view –sysoid all –user** *user* **-password** *password*. For more information, see vcb uvne view, page 18-25.

Note For the list of global options, see Global Command Options, page 18-21.

Error Codes

Table 18-11 Error Codes—vcb uvne modify

Code	Description
102	U-VNE template file not found.
103	No such template name in the templates file.
104	No such device type configuration exists.
112	The sysOID does not exist or already exists and is already modeled as a VNE.

Note For the list of general VCB error codes, see General Error Codes, page 18-22.

vcb uvne delete

The **vcb uvne delete** command deletes a U-VNE.

Synopsis

```
vcb uvne delete -sysoid sysoid -user username -password password
```

```
vcb uvne delete -sysoid sysoid -devicefamily DeviceFamilyName -scheme schemeName
-softwareversion "softwareVersionNumber" -user username -password password
```

Description

The **vcb uvne delete** command is useful when migrating from a U-VNE to a developed VNE. If, in an upgrade, Prime Network provides a DSP that contains a developed VNE to support the device type, the need for the U-VNE is eliminated. You must delete the U-VNE before Prime Network can use the developed VNE to model and manage the device.

Deleting a template-based U-VNE has no effect on the U-VNE template from which it derives its implementation.

Usage Example

Example 1

```
vcb uvne delete -sysoid .1.2.3.4 -user root -password admin
```

Deletes the U-VNE-driver configured for the device with sysOID 1.2.3.4.

Example 2

```
vcb uvne delete -sysoid .1.2.3.4 -devicefamily 70xx -scheme product
-softwareversion "12.0(23)S3"
```

Deletes the U-VNE driver configured for the device with sysOID .1.2.3.4 and running software version 12.0(23)S3 from device family 70xx in the product scheme.

Note that the vcb uvne delete syntax should match the vcb uvne add syntax to avoid items being left in the site.xml after the delete action. For example, if the vcb uvne add syntax is as follows:

```
vcb uvne add -sysoid .1.3.6.1.4.1.9.1.917 -softwareversion "15.0(2)SG1"
-clonesoftwareversion "gt 12.2(52)SG" -clonedevicefamily cisco-catalyst-4900-series
-scheme product -devicetype CISCO_CATALYST_4900M -override -user root -password admin
```

then, the vcb uvne delete syntax should be as follows:

```
vcb uvne delete -sysoid .1.3.6.1.4.1.9.1.917 -devicefamily cisco-catalyst-4900-series
-scheme product -softwareversion "15.0(2)SG1" -user root -password admin
```

Options

Table 18-12 Options and Arguments—vcb uvne delete

Option *Argument*	**Description**
sysoid *sysoid*	The sysObject ID of the U-VNE configuration that you want to delete. **Note** Deleting the U-VNE does not delete or otherwise affect the U-VNE template from which the U-VNE was created.

Note For the list of global options, see Global Command Options, page 18-21.

Error Codes

Table 18-13 Error Codes—vcb uvne delete

Code	**Description**
112	The sysOID does not exist or already exists and is already modeled as a VNE.

Note For the list of general VCB error codes, see General Error Codes, page 18-22.

Commands for Adding and Managing Device Types

Use the following **vcb devicetype** commands to create, view, modify, and delete user-experience attributes for the specified device type:

- vcb devicetype add, page 18-31
- vcb devicetype view, page 18-33
- vcb devicetype modify, page 18-34
- vcb devicetype delete, page 18-35

vcb devicetype add

The **vcb devicetype add** command creates new user-experience attributes for the specified device type. Each device type is associated with a user-friendly name, icon, and device grouping.

Synopsis

```
vcb devicetype add -devicetype device type -category prime network device category -name
device name [-key device type key]-user root -password admin
```

Description

The **vcb devicetype add** command creates user-experience attributes that affect how a device type is managed and displayed in Prime Network.

Note This command is typically used before adding a new U-VNE using the **vcb uvne add** command. For more information, see vcb uvne add, page 18-23.

Usage Example

```
vcb devicetype add -devicetype CISCO_1760 -category ROUTER -name "Cisco 1760 Router"
-user root -password admin
```

Adds a device-type definition for the Cisco 1760 router, including the device category and user-friendly name that will appear in the Prime Network UI. Reference this definition when adding U-VNE definitions for a Cisco 1760 device using the **vcb uvne add** command.

Options

Table 18-14 Options and Arguments—vcb devicetype add

Option *Argument*	Description
devicetype *device type*	The name to assign to the new device type. Each name must be unique. To see existing device types, use the **vcb devicetype view** command.
category *prime network device category*	The category to assign to the new device type, entered as a string (router, switch, unknown, and so on). By default, the device category defined in the U-VNE template is used.
name *device name*	The name to display for this device type in Prime Network. By default, the name is empty. **Note** The name need not be unique. The VCB does not enforce a naming convention for this value.
key *device type key*	(Optional) The unique ID of the new device type. This value is not displayed in Prime Network. **Note** We recommend that you *not* use this option, and let the VCB define the key instead.

Note For the list of global options, see Global Command Options, page 18-21.

Error Codes

Table 18-15 Error Codes—vcb devicetype add

Code	Description
501	Device type name already exists in the deviceTypes
502	Category value does not match any of the possible values
503	Key for this template is not unique
504	Device type name contains illegal characters
505	SysOID not found or not bound to device type in devicetypes.xml

Note For the list of general VCB error codes, see General Error Codes, page 18-22.

vcb devicetype view

The **vcb devicetype view** command returns an existing device-type association.

Synopsis

```
vcb devicetype view -devicetype {device type | all} -user root -password admin
```

Description

Use the **vcb devicetype view** command to:

- Display user-experience attributes based on the specified device type.
- Display the list of device types defined in the system.
- Display all available device categories.

Usage Examples

Example 1

```
vcb devicetype view -devicetype all -user root -password admin
```

Returns a list of all device types defined in Prime Network.

Example 2

```
vcb devicetype view -devicetype CISCO_1760 -user root -password admin
```

Returns the device type details for device type CISCO_1760, including the category and user-friendly name defined for this type.

Options

Table 18-16 Options and Arguments—vcb devicetype view

Option *Argument*	Description
devicetype *device type*	The name of the device type user-experience attributes (including name and device category) that you want to view. **Tip** Enter **all** as the *devicetype* to view a list of all device types defined in the system.
category all	Returns all defined device categories (router, switch, and so on) and their numeric equivalents.

Note For the list of global options, see Global Command Options, page 18-21.

Error Codes

Table 18-17 Error Codes—vcb devicetype view

Code	Description
511	deviceType not found

Note For the list of general VCB error codes, see General Error Codes, page 18-22.

vcb devicetype modify

The **vcb devicetype modify** command modifies the user-experience attributes associated with specific device types.

Synopsis

```
vcb devicetype modify -devicetype device type name [-category prime network device
category] [-name device name] [-key device type key]-user username -password password
```

Description

The **vcb devicetype modify** command overwrites the user-experience settings defined for a device type, including the name, icon, and device category as they appear in Prime Network.

Usage Example

```
vcb devicetype modify -devicetype CISCO_1760 -category DSLAM -name "Cisco 1760 DSLAM"
-user root -password admin
```

Modifies the category and name assigned to the specified device type. Any VNE that uses the U-VNE-driver associated with this device type inherits these modified user-experience attributes.

Options

Table 18-18 Options and Arguments—vcb devicetype modify

Option *Argument*	**Description**
devicetype *device type*	The name of the device type user-experience attributes that you want to modify.
category *prime network device category*	(Optional) Modifies the category assigned to the device type. Enter the category as a string (router, switch, unknown, and so on).
name *device name*	(Optional) Modifies the name that is displayed for this device type in Prime Network.
key *device type key*	(Optional) Modifies the unique ID of the device type. This value is not displayed in Prime Network. **Note** We recommend that you *not* use this option, and let the VCB define the key instead.

Note For the list of global options, see Global Command Options, page 18-21.

Error Codes

Table 18-19 Error Codes—vcb devicetype modify

Code	**Description**
502	Category value does not match any of the possible values for this enum
503	Key for this device type is not unique
504	Device type name contains illegal characters
511	Device type not found

Note For the list of general VCB error codes, see General Error Codes, page 18-22.

vcb devicetype delete

The **vcb devicetype delete** command deletes the user-experience attributes that are defined for the specified device type.

Synopsis

```
vcb devicetype delete -devicetype device type -userdefined -user username
-password password
```

Description

The **vcb devicetype delete** command deletes the user-friendly name, icon, and grouping that were defined for the specified device type from the site.xml file. It does not delete or otherwise modify the U-VNE template from which the U-VNE for this device type was created.

Usage Example

```
vcb devicetype delete -devicetype CISCO_1760 -userdefined -user root -password admin
```

Deletes the user-experience attributes defined for the CISCO_1760 device type.

Options

Table 18-20 Options and Arguments—vcb devicetype delete

Option *Argument*	Description
devicetype *device type*	The name of the device type from which you want to delete user-experience attributes.

Note For the list of global options, see Global Command Options, page 18-21.

Error Codes

Table 18-21 Error Codes—vcb devicetype delete

Code	Description
511	Device type not found.

Note For the list of general VCB error codes, see General Error Codes, page 18-22.

Testing and Certifying U-VNEs

After using the VCB to perform device customizations, you must test the VNE or U-VNE to verify that the device and its components can be managed by Prime Network to your satisfaction.To test the results of your VCB commands, we recommend that you add a single instance of the customized VNE to its own AVM in Prime Network.

Depending on the approach that you selected to create a U-VNE, by cloning or by referencing a template, obtain technology and application support information for the cloned-from device (see the *Cisco Prime Network 3.8 Administrator Guide*) or the template (see U-VNE Templates, page 22-1). Use the references to do the following:

- Rigorously assess compatibility issues.
- Identify potential side effects on your device, network, and management application (Prime Network).

Examples of the support you should test for include:

- Physical containment
- Logical inventory
- Topology
- Events

We recommend that you record the results of your tests in a compliance report, which certifies the capabilities of the customized VNE. See Preparing Compliance Reports, page 18-39. After you successfully complete the testing process, create additional VNE instances to manage the rest of the devices of this type in the network.

Note When performing the tests described in this chapter, bear in mind that unlike developed VNEs, U-VNEs are not optimized for a particular device type. The VCB enables you to extend VNEs and create U-VNEs as specified in this document. Other Prime Network features, such as Soft Properties and Command Builder, offer further extensibility to the U-VNE and developed VNE drivers. Prime Network users who need more management capability—or who prefer to have developed VNE drivers for devices that are not already in the Prime Network VNE support scope—can contact their Cisco account representative for any available Cisco Advanced Services alternative.

For details about testing U-VNEs, please see the following sections:

- Setting Up the Test Environment, page 18-37
- Procedures forTesting and Certifying U-VNEs, page 18-39
- Troubleshooting Device Type Customizations, page 18-43

Setting Up the Test Environment

Before beginning the testing process, you must ensure that:

- All the mandatory processes—avm11, avm0, and avm100 (only for event support)—are running normally.
- You have access to the device.
- You complete the actions described in Table 18-22.

Table 18-22 Actions to Perform Before You Begin Testing

Action	Purpose	For More Information
Configure the device to send events to the Prime Network server.	This is required before you test whether Prime Network can receive events from the VNE.	Testing and Certifying Event Customizations, page 20-12
Create an AVM for hosting the VNE instance on which you will perform the tests.	By isolating the test VNE on its own AVM, you prevent any actions that might impact the functionality of your network.	Creating a Test AVM and VNE, page 18-38
Measure device performance (such as CPU and memory usage) on the device, with and without the VNE.	This enables you to determine whether the VNE is placing an unreasonable load on the device.	Measuring Device Performance, page 18-39

Table 18-22 Actions to Perform Before You Begin Testing (continued)

Action	Purpose	For More Information
Open the relevant VNE documentation.	The information contained in the VNE documentation helps you during the testing process to verify how well Prime Network handles the VNE customizations you created.	• U-VNE Templates, page 22-1 • *Cisco Prime Network 3.8 Reference Guide*
Prepare a compliance report	Fill in the compliance report during testing to track your results against the list of supported technologies and properties outlined in U-VNE Templates, page 22-1.	Preparing Compliance Reports, page 18-39

Creating a Test AVM and VNE

We recommend that you create an AVM for the sole purpose of hosting the VNE instance on which you will perform the tests described in this chapter. Placing the VNE instance in its own AVM enables you to test your customizations in a safe environment, where the logs and any errors generated by the VNE are isolated from the rest of your network. This enables you to proceed with testing without having your VNE customizations impact the network, until you are satisfied that the customizations function as required.

Testing a single VNE instance also helps you scale a rollout more easily. For example, if you have 100 devices of a certain type that you are introducing to your managed network, first create a single VNE instance on which to test your customizations. After testing is complete, create additional VNEs for the other 99 devices.

AVMs and VNEs can be created in the Prime Network Administration client. For further details about creating an AVM and adding a VNE, see “Creating an AVM” and “VNE Administration: VNE Lifecycle and Creating VNEs” in the *Cisco Prime Network 3.8 Administrator Guide*.

Checking the Communication and Investigation State of the Test Instance

After you create a test instance of the U-VNE, you must verify that Prime Network can communicate with the device (communication state) and that Prime Network can discover the physical and logical inventory of the device (investigation state).

To check the communication and investigation state of the test instance:

Step 1 Log into Prime Network Vision.

Step 2 Choose **Network Inventory > Network Elements**.

Step 3 Perform a search to locate the device.

Step 4 Check the Communication State column to see the status of the communication between Prime Network and the device:

- If the status is Device Reachable, this indicates that all of the enabled protocols on the device are responding.
- If the status is Device Unreachable, this indicates that at least one of the enabled protocols is not responding. If this occurs, troubleshoot the problem, as described in Device Unreachable, page 18-43.

Step 5 Check the Investigation State column to see how successfully the VNE has modeled the device it represents:

- If the status is Normal, proceed to the next test.
- If the status is Incomplete, this indicates that Prime Network is unable to model all the components in the device. You must investigate further to determine whether the components or properties that cannot be modeled prevent you from using the U-VNE in your network. For troubleshooting details, see Investigation State Not Operational, page 18-44.

Note For more information, see "Troubleshooting VNE Modeling" in the *Cisco Prime Network 3.8 Administrator Guide*.

Measuring Device Performance

We recommend that you use an element manager or similar application to measure CPU and memory usage on the device before you start managing it with the VNE. This provides a baseline for comparison when you later measure the load on the VNE as part of the testing process. You should simulate various scenarios, including those that place heavy loads on the device, so that you can later determine the effect of your VNE customizations on device performance. See Testing CPU and Memory Usage, page 18-42.

Procedures forTesting and Certifying U-VNEs

To test and certify U-VNEs, we recommend that you follow these procedures:

- Preparing Compliance Reports, page 18-39
- Testing the Physical Inventory, page 18-40
- Testing the Logical Inventory, page 18-41
- Testing CPU and Memory Usage, page 18-42
- Testing That Prime Network Receives Events from an NE and Generates Tickets, page 18-42

Preparing Compliance Reports

As you complete the tests described in this chapter, we recommend that you capture the results in a compliance report. A compliance report makes it easier for you to evaluate whether the U-VNE suits your needs or whether you will need to try again by cloning from a different device or using a different U-VNE template.

Create a compliance report using the information provided in *Cisco Prime Network 3.8 Reference Guide* (for a U-VNE that was cloned from an existing device family) or U-VNE Templates, page 22-1 (for a template-based U-VNE). Use the information to do the following:

1. Copy each technology table to a file (such as a spreadsheet).
2. Remove any columns that describe support for templates other than the one you are using.
3. Add a column for verifying whether each supported property is modeled in Prime Network.
4. Fill out the final column, identifying the supported and unsupported properties.

Assessing a customization requires that you test on a reasonable configuration (test-to-success) and that you evaluate extreme cases (test-to-fail).

Note You cannot use the VCB to change the templates themselves; for example, to support additional CLIs or MIBs or to modify the parsing of the device results.

If you find gaps in the modeling of the U-VNE, you can try using Soft Properties. If you need more management capability or you prefer to have developed VNE drivers for devices that are not already in the Prime Network VNE support scope, contact your Cisco account representative for any available Cisco Advanced Services alternative.

Testing the Physical Inventory

The physical inventory reflects the physical components of the managed device, including its:

- Port Connector—Port details.
- Shelf—Chassis or rack where various types of equipment may be placed or inserted.
- Slot—Details on where the modules are inserted.
- Module—Physical module or adapter card details; hardware description.

Note The list of components that are actually displayed for the U-VNE in the Prime Network client are dependent on the device from which the U-VNE was cloned or on the template that was used to create the U-VNE.

To view the physical inventory:

Step 1 Log into Prime Network Vision.

Step 2 Choose **Network Inventory > Network Elements**.

Step 3 Perform a search to locate the device.

Step 4 Double-click the device.

Step 5 Expand the **Physical Inventory** tree node.

Step 6 Compare the physical containment displayed in the Physical Inventory tree with the actual components contained in the device managed by the U-VNE. In particular, make sure that the following information is accurate:

- Slot Number
- Number of Ports
- Connector Type

Step 7 Look for signs that the U-VNE does not fully model the device, including:

- Investigation state is Currently Unsynchronized.
- A module without ports; for example, if the WS-X6724 module is described as having 24 ports, but no ports appear beneath the module in the tree.

Step 8 Select each component in the tree, then check the Properties window to verify that the physical inventory includes all the properties that are supported by the U-VNE (as defined in the U-VNE template on which it is based).

Step 9 Record your results in a compliance report. See Preparing Compliance Reports, page 18-39.

Step 10 If there are components or properties that are missing or incomplete, you must decide whether these components or properties prevent you from using the U-VNE in your network:

- If you determine that the U-VNE can be managed successfully by Prime Network without the unsupported component or property, proceed to the next test.

Tip In the case of a missing property, try adding it using the Soft Property Builder.

- If you need more management capability or you prefer to have developed VNE drivers for devices that are not already in the Prime Network VNE support scope, you can contact your Cisco account representative for any available Cisco Advanced Services alternative.

  (The VCB enables you to extend VNEs and create U-VNEs as specified in this chapter. Other Prime Network features, such as Soft Properties and Command Builder, described in separate chapters, offer further extensibility to the U-VNE and developed VNE drivers.)

Note To delete a U-VNE, see Deleting a VNE Driver, page 18-12.

Testing the Logical Inventory

The logical inventory reflects dynamic data such as configuration data, forwarding, and service-related components that affect traffic handling in the element. The list of components that are actually displayed in the logical inventory are dependent on the technologies supported either the device family from which the U-VNE was cloned or by the template that was used to create the U-VNE.

To view the logical inventory:

Step 1 Log into Prime Network Vision.

Step 2 Choose **Network Inventory > Network Elements**.

Step 3 Perform a search to locate the device.

Step 4 Double-click the device.

Step 5 Expand the **Logical Inventory** tree node.

Step 6 Compare the information displayed in the Logical Inventory tree with the actual technologies supported by the U-VNE, as defined by the template on which the U-VNE is based. See U-VNE Templates, page 22-1.

Step 7 Select each component in the tree, then check the Properties view to verify that the logical inventory includes all the properties that are supported by the U-VNE (as defined in the U-VNE template on which it is based).

Verify the IP interfaces by querying the device for its list of IP interfaces and verifying that all of them appear under IP Flow Points. Verify that all the IP interfaces configured on the device appear in the IP Interfaces tab under Routing Entity.

Step 8 Record your results in a compliance report. See Preparing Compliance Reports, page 18-39.

Step 9 If any technologies or properties are missing or incomplete, you must investigate further to determine whether these technologies or properties prevent you from using the U-VNE in your network:

- If you determine that the device can be managed successfully by Prime Network without the unsupported technology or property, proceed to the next test.
- If you need more management capability or you prefer to have developed VNE drivers for devices that are not already in the Prime Network VNE support scope, you can contact your Cisco account representative for any available Cisco Advanced Services alternative.

Testing CPU and Memory Usage

Note Ensure that you perform both positive testing (on a reasonable configuration) and negative testing (on simulations of expected network scenarios).

In addition to verifying how well the U-VNE models the device, we recommend that you measure the CPU and memory usage demands placed by the U-VNE on the device. During test preparation, before adding the VNE instance, you measured CPU and memory usage on the NE; (see Measuring Device Performance, page 18-39). Now, compare the usage on the NE against the usage for the VNE instance on the Prime Network AVM and unit as follows:

- To view CPU usage, look at the properties of the device in the Inventory window.
- To monitor additional information, such as memory usage, use the Prime Network diagnostic client. To access the diagnostic client, enter the following address in your web browser:

  https://<gateway IP address>:1311

  Username: Admin

  Password: The password used to access the Prime Network applications (Manage, Vision, Events).

Testing That Prime Network Receives Events from an NE and Generates Tickets

This topic provides steps for verifying that Prime Network receives events from an NE and that the VNE driver parses the events correctly, generating tickets if the events are ticketable.

Note Event parsing depends on the completeness and correctness of the modeling. Parsing BGP events, for example, depends on BGP being modeled correctly in the inventory.

Before You Begin

Make sure that you have configured the device to send events to the Prime Network server and created a Link Down event.

To check for events from the NE:

Step 1 Log into Prime Network Vision.

Step 2 Add the device to a map.

Step 3 Right-click on the device in the map and select Filter Tickets. The tickets pane below the map will show the tickets for the selected device.

Step 4 Check whether the Link Down event that you generated during your test preparations appears in the table.

Step 5 Generate additional events from the device using the list of supported events (listed in *Cisco Prime Network 3.8 Reference Guide* for the cloned-from device family or listed in Chapter 22, "U-VNE Templates" for a template-based U-VNE), then see if they appear in the events table.

Step 6 If the events do not appear in the events table, proceed as follows:

- Check for mistakes in the device configuration.
- Use an external tool, such as a MIB browser, to determine whether events are being sent by the device.
- Troubleshoot event customization, as described in Troubleshooting Event Customization (CLI), page 20-34.

Troubleshooting Device Type Customizations

This section describes basic troubleshooting procedures to perform when Prime Network cannot communicate with a U-VNE, and includes the following topics:

- Device Unreachable, page 18-43
- Investigation State Not Operational, page 18-44

Device Unreachable

If you cannot communicate with a U-VNE, try the following:

- Launch Prime Network Administration and verify the Admin Status and Operational Status of the VNE. The Admin Status should be Enabled and the Operational Status should be Up.
- Verify that you are using the correct SNMP and Telnet credentials for the device.
- Ping the device from the Prime Network server.

Note For more information, see the *Cisco Prime Network 3.8 Administrator Guide*.

Investigation State Not Operational

If the investigation state of the U-VNE is any value other than Operational, perform the actions described in the following table.

Table 18-23 Troubleshooting the Investigation State

State	Description	Action
Partially Synchronized	The U-VNE is encountering a problem, such as an exception caused by a particular discovery command or an unsupported module.	Examine the AVM log for messages about: • Failed commands or OIDs. • Unsupported modules. See Log Entry for Unsupported Module, page 18-44. **Note** For more information, see Prime Network Logs in *Cisco Prime Network 3.8 Administrator Guide*.
Unsupported	The U-VNE is encountering registration problems.	Use the **vcb sitechanges view** command to verify that the U-VNE registrations were created properly. In particular, make sure that you used the correct sysObjectID for this device type when using the VCB to create the U-VNE-driver. If the sysObjectID is not correct: • Delete the U-VNE. See Deleting a VNE Driver, page 18-12. • Recreate the U-VNE driver using the correct sysObjectID. See Creating a U-VNE - Procedure, page 18-10.

Log Entry for Unsupported Module

The log entry for an unsupported module looks like this:

```
ERROR [07 07 2010 17:07:47.810 IST] - PhysicalCommandHandler.isEntitySupported - 
192.168.20.1 can't create module with entry
value spec/physical/modules/.1.3.6.1.4.1.9.12.3.1.9.29.118.1/loaders will use
default module loader
```

If you have an unsupported module, your choices are to either use the VCB to add the module (see Enabling Support for a New Standard Module, page 19-1), or manage the device in Prime Network without managing that particular module.

Moving a U-VNE to Production

Tip Always perform customization in a maintenance window.

Export the configuration from your lab setup and import it into the production setup using the following procedures:

- Exporting VCB Registry Customizations, page 21-3

- Importing VCB Registry Customizations, page 21-3

After you import the changes to the production server:

1. Restart Prime Network or at a minimum, restart all VNEs and AVMs that need to support the customizations.
2. Repeat testing and certification to ensure that the customizations are functioning as expected in your production environment. See Procedures forTesting and Certifying U-VNEs, page 18-39.