



## VCB: CLI Commands and Template Reference

---

These topics provide reference information for VCB CLI commands and templates:

- [VCB CLI Reference: vcb and vcb sitechanges Commands, page B-1](#)
- [VCB CLI Reference: U-VNEs Commands, page B-3](#)
- [VCB CLI Reference: Standard and Pluggable Modules Commands, page B-17](#)
- [VCB CLI Reference: Events Commands, page B-31](#)
- [VCB Template Reference, page B-56](#)

### VCB CLI Reference: vcb and vcb sitechanges Commands

These topics provide reference information for the **vcb** and **vcb sitechanges** commands which are used to deploy, maintain, and remove VCB customizations.

- [vcb, page B-1](#)
- [vcb sitechanges, page B-3](#)

#### vcb

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

#### Syntax

```
{uvne|module|pluggablemodule|event|eventpattern|eventparsingrules|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:

- [VCB CLI Reference: U-VNEs Commands, page B-3](#)
- [VCB CLI Command Reference: Device Types Commands, page B-12](#)

- [VCB CLI Command Reference: Standard Modules, page B-18](#)
- [VCB CLI Command Reference: Pluggable Modules, page B-25](#)

## Global Command Options

[Table B-1](#) describes the global options and arguments that are common to all **vcb** commands and subcommands.

**Table B-1** 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, and debug. For example, if debuglevel is set to warn, all warning and error messages are saved to the log file.
logfile <i>logfile</i>	Logs the CLI output to the file specified in the <i>logfile</i> argument.
userdefined	Displays the registrations that have been added to the site.xml file using the VCB or any other tool.
user <i>username</i>	BQL username.
password <i>password</i>	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 B-2** 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.

Table B-2 General Error Codes—*vcb* (continued)

Code	Description
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.

### Syntax

```
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.

## VCB CLI Reference: U-VNEs Commands

These topics provide reference information for the **vcb uvne** commands you can use to create, view, modify, and delete U-VNEs:

- [vcb uvne add](#), page B-4
- [vcb uvne view](#), page B-6
- [vcb uvne modify](#), page B-9
- [vcb uvne delete](#), page B-11

## 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.

### Syntax

```
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 B-12.

### 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 B-12).

#### 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 B-6.)

#### 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) based on the device family 100xx. Also adds support for the new software version for this device family. (For a list of already supported software versions, see [vcb uvne view](#), page B-6.)

#### 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 B-3 Options and Arguments—vcb uvne add*

Option Argument	Description
sysoid <i>sysoid</i>	The sysObject ID of the device for which to create a U-VNE-driver using the implementation defined in the U-VNE template. <b>Note</b> Each sysOID value in the system must be unique.
template <i>template name</i>	The name of the U-VNE template from which to create the U-VNE-driver. See <a href="#">U-VNE Templates</a> , page B-56. <b>Note</b> This option is mutually exclusive with the <b>-clonesysoid</b> and <b>-clonedevicefamily</b> options.
group <i>template filename</i>	The name of the file in which the U-VNE template is located. U-VNE templates are located in the uvne-product file. <b>Note</b> Use this option with the <b>-template</b> option only.
devicetype <i>device type name</i>	(Optional) The U-VNE device type name. If not specified, the device type is: <ul style="list-style-type: none"> <li>Defined as Unknown when the option is not specified.</li> <li>Inherited from the device or device family when you use the <b>-clonesysoid</b> option.</li> </ul> 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 <b>vcb devicetype add</b> command. For more information, see <a href="#">vcb devicetype add</a> , page B-12.
clonesysoid <i>clone-sysoid</i>	The sysObject ID of an already supported VNE. To view sysOIDs for supported VNEs, use the <b>vcb uvne view -sysoid</b> all command.
softwareversion <i>software-version</i>	The software version that you want to add for a U-VNE. To obtain the software version string, use the <b>show version</b> command.
clonesoftwareversion <i>clone-softwareversion</i>	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 software versions, use the command <b>vcb uvne view -scheme &lt;ipcore\product&gt; -devicefamily &lt;device family&gt; -user user -password password</b> . For more information, see <a href="#">vcb uvne view</a> , page B-6.

Table B-3 Options and Arguments—*vcb uvne add* (continued)

Option Argument	Description
<code>clonedevicefamily</code> <i>devicefamily</i>	A supported device family. For a list of supported device families, use the command <b>vcb uvne view</b> <b>-scheme</b> <i>&lt;ipcore product&gt;</i> <b>-devicefamily</b> <i>devicefamily</i> <b>-user</b> <i>user</i> <b>-password</b> <i>password</i> . For more information, see <a href="#">vcb uvne view</a> , page B-6. <b>Note</b> This option is mutually exclusive with the <b>-template</b> option.



**Note** For the list of global options, see [Global Command Options](#), page B-2.

## Error Codes

Table B-4 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 B-2.

## vcb uvne view

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

### Syntax

```
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.....:9.7.6.5.4.3.1
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 B-5** Options and Arguments—*vcb uvne view*

Option Argument	Description
<i>sysoid sysoid</i>	Returns the configuration of the specified VNE, including the device type, the user-friendly name, and the template (for U-VNEs). <b>Tip</b> Enter <b>all</b> as the <i>sysoid</i> to view the configuration of all sysOIDs (U-VNEs and developed VNEs) configured in the system.
<i>group template filename</i>	Returns the configuration of all sysOIDs and templates contained in the specified group. <b>Tip</b> Enter <b>all</b> as the <i>template filename</i> to view the template associations for each group in the system.
<i>userdefined</i>	Lists the U-Vne created through vcb command only.
<i>details</i>	Shows following additional information: <ul style="list-style-type: none"> <li>• Module Spec Name associated with the sysoid.</li> <li>• Syslog and Trap Parsing rule name associated with the sysoid</li> </ul>



**Note**

For the list of global options, see [Global Command Options, page B-2](#).

## Error Codes

**Table B-6** 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 B-2](#).



## vcb uvne modify

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

### Syntax

```
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.
- Change the cloning reference of a U-VNE.

### Usage Examples

#### Example 1

```
vcb uvne modify -sysoid .1.2.3.4 -group uvne-product -template GenericUVNE -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 -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 B-7 Options and Arguments—*vcb uvne modify*

Option Argument	Description
<i>sysoid sysoid</i>	The sysObject ID of the U-VNE-driver configuration that you want to modify.
<i>template template name</i>	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.  <b>Note</b> Using this option does not overwrite other configuration changes made with the VCB, such as user-experience attributes that are defined with the <a href="#">vcb devicetype add</a> command.
<i>group template filename</i>	The name of the group that includes the U-VNE template, such as <i>uvne-product</i> .
<i>devicetype device type</i>	The device type associated with the U-VNE.
<i>clonesysoid clone-sysoid</i>	The sysObject ID of an already supported VNE. To view sysOIDs for supported VNEs, use the <b>vcb uvne view -sysoid all</b> command.
<i>softwareversion software-version</i>	The software version that you want to support for a U-VNE. To obtain the software version string, use the <b>show version</b> command.
<i>clonesoftwareversion clone-softwareversion</i>	An already supported software version that you want to clone. The software version must already be supported for a particular device family.
<i>clonedevicefamily deviceFamily</i>	A supported device family. For a list of supported device families, use the command <b>vcb uvne view -sysoid all -user user -password password</b> . For more information, see <a href="#">vcb uvne view, page B-6</a> .

**Note**

For the list of global options, see [Global Command Options, page B-2](#).

## Error Codes

Table B-8 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 B-2](#).

## vcb uvne delete

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

### Syntax

```
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 software configuration for U-VNE driver configured for the device with sysOID .1.2.3.4. Use the command in Example 1 to delete the U-VNE.

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 B-9** Options and Arguments—*vcb uvne delete*

Option Argument	Description
<code>sysoid <i>sysoid</i></code>	The sysObject ID of the U-VNE configuration that you want to delete.  <b>Note</b> 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 B-2](#).

## Error Codes

**Table B-10** 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 B-2](#).

# VCB CLI Command Reference: Device Types Commands

These topics provide reference information for the **vcb devicetype** commands you can use to create, view, modify, and delete user-experience attributes for device types:

- [vcb devicetype add, page B-12](#)
- [vcb devicetype view, page B-14](#)
- [vcb devicetype modify, page B-15](#)
- [vcb devicetype delete, page B-16](#)

## 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.

### Syntax

```
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 B-4](#).

## 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 B-11 Options and Arguments—vcb devicetype add*

Option Argument	Description
devicetype <i>device type</i>	The name to assign to the new device type. Each name must be unique. To see existing device types, use the <b>vcb devicetype view</b> command.
category <i>prime network device category</i>	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 <i>device name</i>	The name to display for this device type in Prime Network. By default, the name is empty. <b>Note</b> The name need not be unique. The VCB does not enforce a naming convention for this value.
key <i>device type key</i>	(Optional) The unique ID of the new device type. This value is not displayed in Prime Network. <b>Note</b> We recommend that you <i>not</i> use this option, and let the VCB define the key instead.



Note

For the list of global options, see [Global Command Options, page B-2](#).

## Error Codes

**Table B-12** 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 B-2](#).

## vcb devicetype view

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

### Syntax

```
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 B-13** Options and Arguments—*vcb devicetype* view

Option Argument	Description
<code>devicetype</code> <i>device type</i>	The name of the device type user-experience attributes (including name and device category) that you want to view.  <b>Tip</b> Enter <b>all</b> as the <i>devicetype</i> to view a list of all device types defined in the system.
<code>category all</code>	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 B-2](#).

## Error Codes

**Table B-14** 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 B-2](#).

## vcb devicetype modify

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

### Syntax

```
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 B-15** Options and Arguments—*vcb devicetype modify*

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



### Note

For the list of global options, see [Global Command Options, page B-2](#).

## Error Codes

**Table B-16** 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 B-2](#).

## vcb devicetype delete

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

## Syntax

```
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 B-17 Options and Arguments—vcb devicetype delete*

Option Argument	Description
devicetype <i>device type</i>	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 B-2](#).

## Error Codes

*Table B-18 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 B-2](#).

# VCB CLI Reference: Standard and Pluggable Modules Commands

You can use the VCB CLI to create and manage standard and pluggable modules as described in these topics:

- [VCB CLI Command Reference: Standard Modules, page B-18](#)
- [VCB CLI Command Reference: Pluggable Modules, page B-25](#)

## VCB CLI Command Reference: Standard Modules

These topics provide reference information for the **vcb module** commands you can use to create, view, and delete standard modules:

- [vcb module add, page B-18](#)
- [vcb module view, page B-20](#)
- [vcb module delete, page B-24](#)

### vcb module add

The **vcb module add** command adds support for a new module type by creating a new registration from the specified template. This support enables VNEs that contain this module to properly recognize and model it in Prime Network.

#### Syntax

```
vcb module add -module module identifier -template template name -group module group
[ -hardwaredesc Hardware Description of the Module ] -user username -password password
vcb module add -module module identifier -template template name -sysoid <sysobject ID
devicefamily under which module should be supported> -scheme <ipcore|product> [
-hardwaredesc Hardware Description of the Module ] -user username -password password
```

#### Description

The **vcb module add** command enables the VNE to automatically detect a physical module based on an implementation defined in a module template. It can be used to:

- Create a module definition based on the module identifier. When this option is used, any VNE that uses the same spec file for its module definitions can model the new module.
- Create a module definition based on an extension to the definition used by the driver of a specific device (defined by its sysOID). When this option is used, only this particular device can model the new module.

This command updates the site.xml file, in which customizations are stored. Doing so enables the tool to differentiate between factory defaults (changes supplied from DSPs) and changes initiated by the VCB.

#### Usage Examples

##### Example 1

```
vcb module add -module ".1.3.6.1.4.1.9.12.3.1.9.29.99" -template "1.2.3.4" -group
ciscoentitymibspec -hardwaredesc cevCat6kWsf6kpf3b -user root -password admin
```

Adds support for a module with the module ID .1.3.6.1.4.1.9.12.3.1.9.29.99, based on the 1.2.3.4 module template, which is listed in the spec group, ciscoentitymibspec. The hardware description is cevCat6kWsf6kpf3b.

**Example 2**

```
vcb module add -module ".1.3.6.1.4.1.9.12.3.1.9.29.99" -sysoid "7.7.7"
-template "1.2.3.4" -scheme product -user root -password admin
```

Adds support for a module with the ID .1.3.6.1.4.1.9.12.3.1.9.29.99, based on module template 1.2.3.4. The VCB looks up the module spec used by the VNE-driver with the sysOID 7.7.7, and registers the new module for this device only (not for all devices that share the same module spec file).

**Options**

*Table B-19 Options and Arguments—vcb module add*

<b>Option Argument</b>	<b>Description</b>
module <i>module identifier</i>	The module identifier, a number, string, or OID. The module identifier should be unique and new.  <b>Note</b> The value specified for this argument must match the value returned from the physical command investigation.
group <i>module group</i>	The module group is a repository that might be shared across multiple device types. It specifies where the template is located. For the module groups that you can extend, see <a href="#">Module Groups and Module Specification Files, page B-66</a> .  <b>Note</b> Regular modules and pluggable modules use different module spec files. For information about pluggable modules, see <a href="#">VCB CLI Command Reference: Pluggable Modules, page B-25</a> .
sysoid <i>sysoid</i>	The sysObjectID of the device that belongs to particular device family for which a concrete module is created using the implementation defined in the template.  <b>Note</b> This sysObjectID must already exist in the system.
scheme <i>scheme</i>	Scheme name which will be used while adding the VNE.
template <i>template name</i>	The name of the template that contains the implementation of the physical investigation of this module.  <b>Tip</b> The template option need not be specified if the module does not support any ports; SIP and other carrier cards are examples of modules that do not support ports.
hardwaredesc <i>hardware description</i>	(Optional) Hardware description of the given module.

**Note**

For the list of global options, see [Global Command Options, page B-2](#).

## Error Codes

*Table B-20 Error Codes—vcb module add*

Code	Description
103	No such template name in the templates file.
111	The SysOID specified does not exist in site.xml.
401	Module already exists in the spec file or in site.xml (thus is already modeled).
402	Module template file not found.



**Note** For the list of general VCB error codes, see [General Error Codes, page B-2](#).

## vcb module view

The **vcb module view** command returns an existing module configuration.

### Syntax

```
vcb module view -sysoid sysoid -module {module identifier | all} -user root -password admin -scheme <scheme>
```

```
vcb module view -group module group -module {module identifier | all} -user root -password admin
```

```
vcb module view -group module group -template {template name | all} -user root -password admin
```

### Description

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

- Display module information based on the driver associated with the defined sysOID.
- Display details about the specified module.
- Display available templates found in the specified module spec file.

### Usage Examples

#### Example 1

```
vcb module view -sysoid .1.2.3.4 -module all -user root -password admin -scheme product
```

Returns the list of supported modules for the device with sysOID 1.2.3.4.

#### Example 2

```
vcb module view -sysoid .1.2.3.4 -module 50068 -user root -password admin -scheme product
```

Returns the port layer definitions for module 50068, which was added to the VNE-driver for the device with sysOID 1.2.3.4.

**Example 3**

```
vcb module view -group ciscophysicalspec2 -module all -user root -password admin
```

Returns a list of all supported modules contained in the specified module group.

**Example 4**

```
vcb module view -group ciscophysicalspec2 -module 20091 -user root -password admin
```

Returns the port layer definitions for module 20091, which is part of the group ciscophysicalspec2.

**Example 5**

```
vcb module view -group ciscoentitymibspec -template all -user root -password admin
```

Returns a list of all templates defined in the specified module group.

**Example 6**

```
vcb module view -group ciscoentitymibspec -template ethernetDefault -user root -password admin
```

Returns the port layer definitions of the template if it is defined in the specified module group.

**Options**

*Table B-21 Options and Arguments—vcb module view*

<b>Option Argument</b>	<b>Description</b>
sysoid <i>sysoid</i>	The sysOID of the device that contains the module configuration that you want to view.
module <i>module identifier</i>	The module whose port layer configuration you want to view. The identifier can be a number, string, or OID, depending on the type of identifier used by the relevant device type. <b>Tip</b> Enter <b>all</b> as the <i>module identifier</i> to return a list of all modules in the defined device or group, listed by identifier.
group <i>module group</i>	The module spec group associated with the module or template whose details you want to view.
template <i>template name</i>	The name of the module template whose port layer configuration you want to view. <b>Tip</b> Enter <b>all</b> as the template name to return the list of all module templates contained in the specified module spec group.
- scheme <i>scheme</i>	Scheme name which will be used while adding the VNE.

**Note**

For the list of global options, see [Global Command Options, page B-2](#).

## Error Codes

**Table B-22** Error Codes—*vcb module view*

Code	Description
103	No such template name in the template file
402	Module template file not found
411	SysOID specified does not exist in site.xml or vendor file
412	Module identifier specified does not exist



**Note** For the list of general VCB error codes, see [General Error Codes, page B-2](#).

## vcb module modify

Use the **vcb module modify** command to modify the associated module template.

### Syntax

```
vcb module modify -module module identifier -group module group -template template name
[ -hardwaredesc Hardware Description of the Module -user username -password password
vcb module modify -module module identifier -sysoid device sysoid -scheme
<ipcore/product> -template template name [ -hardwaredesc Hardware Description of the
Module ] -user username -password password
```

### Description

The **vcb module modify** command can be used to:

- Associate a module with another module template.
- Change the association between a module and its module template for a device specified by its sysOID.

### Usage Examples

#### Example 1

```
vcb module modify -module 20091 -group ciscophysicalspec2
-template GE-fiber-optic-ethernet-default -user root -password admin
```

Modifies the registration for module 20091 by associating it with module template GE-fiber-optic-ethernet-default, which is part of group ciscophysicalspec2. This module definition replaces the one obtained from the template that was specified when the module was first added. This change affects all devices that contain this module.

#### Example 2

```
vcb module modify -sysoid .1.2.3.4 -module 50068 -group ciscophysicalspec2
-template oc3-ppp-default -user root -password admin -scheme product
```

Modifies the registration for module 50068 by associating it with module template cc3-ppp-default. This change affects only the device with sysOID 1.2.3.4.

## Options

**Table B-23** Options and Arguments—*vcb module modify*

Option Argument	Description
module <i>module identifier</i>	The module configuration that you want to modify. The identifier can be a number, string, or OID, depending on the type of identifier used by the relevant device type.  <b>Note</b> The value specified for this argument must match the value returned by the device when investigating this module.
group <i>module group</i>	The name of the group that contains the module template to apply to the module.
sysoid <i>sysoid</i>	The sysObject ID of the device that contains the module configuration that you want to modify.  <b>Note</b> You must specify a sysoid that was added using the VCB.
template <i>template name</i>	The name of the module template with which the defined module should be associated. Use this option to change the module template from which the module derives its configuration.
hardwaredesc <i>hardware description</i>	(Optional) Hardware description of the given module.
scheme <i>scheme</i>	Scheme name which will be used while adding the VNE.



**Note**

For the list of global options, see [Global Command Options, page B-2](#).

## Error Codes

**Table B-24** Error Codes—*vcb module modify*

Code	Description
103	No such template name in the template file
402	Module template file not found
411	SysOID specified does not exist in site.xml or vendor file
412	Module identifier specified not found or already exists in vendor spec file



**Note**

For the list of general VCB error codes, see [General Error Codes, page B-2](#).

## vcb module delete

Use the **vcb module delete** command to delete the association between a module and the implementation defined in a module template.

### Syntax

```
vcb module delete -module module identifier -group module group -user username -password
password
vcb module delete -module module identifier -sysoid device sysoid -scheme
<ipcore/product> -user username -password password
```

### Description



#### Note

Use the **vcb module delete** command to enable VNEs to use factory-supplied updates if they become available.

The **vcb module delete** command is useful when migrating from an interim solution to a more complete implementation. Removing the association to the module from the site.xml file enables Prime Network to automatically detect a deployed implementation in a DSP.

Use the **vcb module delete** command:

- To delete the association between a module and its implementation on a specific device based on a defined sysOID. Other VNEs that use this implementation will still be able to identify the module.
- To delete the association between a module and its implementation on all devices that use the module as specified by the relevant group name. When the **-group** option is used, any VNEs that used this implementation will no longer be able to identify the module.



#### Note

- You can delete only those modules that were originally added using the VCB. Module definitions added as part of a developed VNE cannot be deleted.
- You can delete the module association on a specific device only when the association was created for that specific device.
- Deleting the module configuration does not delete or otherwise affect the template from which the configuration was created.

### Usage Examples

#### Example 1

```
vcb module delete -module 20091 -group ciscophysicalspec2 -user root -password admin
```

Deletes the definition for module 20091 from the group ciscophysicalspec2. The module definition is deleted from all devices that contain this module.

#### Example 2

```
vcb module delete -sysoid .1.2.3.4 -module 50068 -user root -password admin -scheme
product
```

Deletes the definition for module 50068 from the device with sysOID 1.2.3.4. Other devices that contain this module are unaffected.



## Options

**Table B-25** Options and Arguments—*vcb module delete*

Option Argument	Description
module <i>module identifier</i>	The module configuration that you want to delete. The identifier can be a number, string, or OID, depending on the type of identifier used by the relevant device type.
sysoid <i>device sysoid</i>	The sysObject ID of the device that contains the module configuration that you want to delete.
group <i>module group</i>	The module spec group that contains the template whose implementation you want to delete from the module.
scheme <i>scheme</i>	Scheme name which will be used while adding the VNE.



Note

For the list of global options, see [Global Command Options, page B-2](#).

## Error Codes

**Table B-26** Error Codes—*vcb module delete*

Code	Description
402	Module template file not found.
411	sysOID does not exist in the site.xml file or vendor spec file.
412	Module identifier not found or already exists in the vendor spec file.



Note

For the list of general VCB error codes, see [General Error Codes, page B-2](#).

## VCB CLI Command Reference: Pluggable Modules

These topics provide reference information for the **vcb pluggablemodule** commands you can use to create, view, modify, and delete pluggable modules:

- [vcb pluggablemodule add, page B-26](#)
- [vcb pluggablemodule view, page B-28](#)
- [vcb pluggablemodule modify, page B-29](#)
- [vcb pluggablemodule delete, page B-30](#)



Note

To add, view, modify, or delete regular modules, see [VCB CLI Command Reference: Standard Modules, page B-18](#).

## vcb pluggablemodule add

Use the **vcb pluggablemodule add** command to create a pluggable module definition that is associated using the module identifier. Use this command for pluggable modules such as SFPs and XFPs.

### Syntax

```
vcb pluggablemodule add -group groupName -module moduleno -pid pid -mediatype MediaType
-pluggabletype pluggableType -user username -password password
vcb pluggablemodule add -group groupName -containeroid containeroid -user username
-password password
vcb pluggablemodule add -group groupName -basetype basetype -user username -password
password
```

### Description

Prime Network does not model pluggable modules, only the ports on them. Configuration details that are entered using this command are displayed at the port level to differentiate between pluggable and regular ports. Changes made through this command apply to all VNEs that use the same spec file for pluggable ports.

The container oid and base type value are additional configuration information needed to model pluggable module properly. When you add new pluggable module make sure that container oid and base type values are added in the system.

### Usage Example

```
vcb pluggablemodule add -module ".1.3.6.1.4.1.9.12.3.1.9.51.22"
-group pluggable-ports-spec -mediatype fiber -pluggabletype SFP -user root -password
admin
```

Adds support for a pluggable module of type SFP with the module ID .1.3.6.1.4.1.9.12.3.1.9.51.22, to the pluggable-ports-spec module specification file. The media type is fiber optic.

```
vcb pluggablemodule add -group pluggable-ports-spec -basetype .1.3.6.1.4.1.9.12.3.1.9.52
-user root -password admin
```

Add pluggable module base type configuration with the base type value .1.3.6.1.4.1.9.12.3.1.9.52, to the pluggable-ports-spec module specification file.

```
vcb pluggablemodule add -group pluggable-ports-spec -containeroid
.1.3.6.1.4.1.9.12.3.1.5.153 -user root -password admin
```

Add pluggable module container oid configuration with the container oid .1.3.6.1.4.1.9.12.3.1.9.52, to the pluggable-ports-spec module specification file.

## Options

**Table B-27** Options and Arguments—*vcb pluggablemodule add*

Option Argument	Description
module <i>moduleNo</i>	The identifier for the module that you want to add. The identifier can be a number, string, or OID, depending on the type of identifier used by the relevant device type.  <b>Note</b> The value specified for this argument must match the value returned by the device when investigating this module.
group <i>pluggable port group</i>	The name of the group.
pid <i>pid</i>	The pid for the module.
mediatype <i>Media Type</i>	The media type of the port on the pluggable module.
pluggable type <i>pluggableType</i>	The type of the pluggable module; one of SFP, XFP, X2, XENPAK.
containeroid	The oid of a pluggable module container where the pluggable module will be inserted.
basetype	The oid of a pluggable module without the last octet in the oid  For example, if the oid of a pluggable module is .1.3.6.1.4.1.9.12.3.1.9.52.10 then the basetype oid would be .1.3.6.1.4.1.9.12.3.1.9.52



Note

For the list of global options, see [Global Command Options, page B-2](#).

## Error Codes

**Table B-28** Error Codes—*vcb pluggablemodule add*

Code	Description
403	Pluggable module spec file not found
404	Invalid pluggable module type



Note

For the list of general VCB error codes, see [General Error Codes, page B-2](#).

## vcb pluggablemodule view

Use the **vcb pluggablemodule view** command to display details for a pluggable module.

### Syntax

```
vcb pluggablemodule view -group groupName -module moduleno | all [-userdefined] -user
username -password password
vcb pluggablemodule view -group groupName -containeroid oid | all [-userdefined] -user
username -password password
vcb pluggablemodule view -group groupName -basetype basetype | all [-userdefined] -user
username -password password
```

### Description

Use the **vcb pluggablemodule view** command to display the pluggable module details—such as pid, media type, and pluggable type—for one or all modules in the given pluggable port group.

### Usage Examples

```
vcb pluggablemodule view -group pluggable-port-spec -module
".1.3.6.1.4.1.9.12.3.1.9.51.22" -user root -password admin
```

Returns the pluggable module details—such as pid, media type, and pluggable type—for the module with the given vendor oid.

```
vcb pluggablemodule view -group pluggable-port-spec -containeroid all -user root
-password admin
```

Returns the pluggable module container oid details, if the list has userdefined container oid, it is marked with # after the oid.

```
vcb pluggablemodule view -group pluggable-port-spec -basetype all -user root -password
admin
```

Returns the pluggable module details.

### Options

**Table B-29** Options and Arguments—*vcb pluggablemodule view*

Option Argument	Description
module <i>moduleno</i>	The identifier for the module that you want to view. The identifier can be a number, string, or OID, depending on the type of identifier used by the relevant device type.  <b>Tip</b> Enter <b>all</b> as the <i>module no</i> to return the list of all modules contained in the specified pluggable port group.
group <i>groupname</i>	The name of the group.



#### Note

For the list of global options, see [Global Command Options, page B-2](#).

## Error Codes

**Table B-30** Error Codes—*vcb pluggablemodule view*

Code	Description
403	Pluggable module spec file not found.



Note

For the list of general VCB error codes, see [General Error Codes, page B-2](#).

## vcb pluggablemodule modify

Use the **vcb pluggablemodule modify** command to change the pluggable module configuration.

### Syntax

```
vcb pluggablemodule modify -group groupName -module moduleno -pid pid {[-mediatype
Media Type] | [-pluggabletype pluggabletype]}
```

### Description

The **vcb pluggablemodule modify** command enables you to modify the configuration for a pluggable module.

### Usage Example

```
vcb module modify -module ".1.3.6.1.4.1.9.12.3.1.9.51.22" -group pluggable-ports-spec
-mediatype "fiber optic" -user root -password admin
```

Modifies the mediaType attribute of the pluggable module with identifier .1.3.6.1.4.1.9.12.3.1.9.51.22.

### Options

**Table B-31** Options and Arguments—*vcb pluggablemodule modify*

Option Argument	Description
module <i>moduleno</i>	The identifier for the pluggable module configuration that you want to modify. The identifier can be a number, string, or OID, depending on the type of identifier used by the relevant device type.
group <i>pluggable port group</i>	The name of the group.
pid <i>pid</i>	The pid for the module.
mediatype <i>Media Type</i>	(Optional) The media type of the port on the pluggable module.
pluggable type <i>pluggableType</i>	(Optional) The type of the pluggable module; one of SFP, XFP, X2, AND XENPAK.



Note

For the list of global options, see [Global Command Options, page B-2](#).

## Error Codes

**Table B-32** Error Codes—*vcb pluggablemodule modify*

Code	Description
403	Pluggable module spec file not found
404	Invalid pluggable module type



### Note

For the list of general VCB error codes, see [General Error Codes, page B-2](#).

## vcb pluggablemodule delete

Use the **vcb pluggablemodule delete** command to delete a pluggable module that was previously added using the VCB.

### Syntax

```
vcb pluggablemodule delete -group pluggableportgroupN -module moduleno -user username
  -password password
vcb pluggablemodule delete -group pluggableportgroup -containeroid container oid -user
  username -password password
vcb pluggablemodule delete -group pluggableportgroup -basetype basetype oid -user
  username -password password
```

### Description

Use the **vcb pluggablemodule delete** command to delete pluggable modules that were created using the VCB.



**Note** Use the **vcb pluggablemodule delete** command to enable VNEs to use factory-supplied updates when they are available.

### Usage Examples

```
vcb pluggablemodule delete -module ".1.3.6.1.4.1.9.12.3.1.9.51.22" -group
  pluggable-ports-spec -user root -password admin
```

Removes the definition for the pluggable module with identifier .1.3.6.1.4.1.9.12.3.1.9.51.22.

```
vcb pluggablemodule delete -basetype ".1.3.6.1.4.1.9.12.3.1.9.52" -group
  pluggable-ports-spec -user root -password admin
```

Removes the definition for the pluggable module basetype with identifier .1.3.6.1.4.1.9.12.3.1.9.52.

```
vcb pluggablemodule delete -containeroid ".1.3.6.1.4.1.9.12.3.1.5.153" -group
  pluggable-ports-spec -user root -password admin
```

Removes the definition for the pluggable module container oid with identifier .1.3.6.1.4.1.9.12.3.1.5.153.

## Options

*Table B-33 Options and Arguments—`vcb pluggablemodule delete`*

Option Argument	Description
module <i>module no</i>	The identifier for the pluggable module that you want to delete. The identifier can be a number, string, or OID, depending on the type of identifier used by the relevant device type.
group <i>pluggable port</i> <i>group</i>	The name of the group from which to delete the pluggable module.



Note

For the list of global options, see [Global Command Options, page B-2](#).

# VCB CLI Reference: Events Commands

This section describes the CLI commands that can be used to customize events, as follows:

- [VCB CLI Reference: `vcb event` Commands, page B-31](#)
- [VCB CLI Reference: `vcb eventparsingrules` Commands, page B-40](#)
- [VCB CLI Reference: `vcb eventpattern` Commands, page B-47](#)
- [VCB CLI Reference: `vcb eventarg` Command, page B-53](#)

## VCB CLI Reference: `vcb event` Commands

These topics provide reference information for the **vcb event** commands you can use to create, view, modify, and delete events:

- [vcb event add, page B-31](#)
- [vcb event view, page B-34](#)
- [vcb event modify, page B-37](#)
- [vcb event delete, page B-39](#)

### vcb event add

Use the **vcb event add** command to create an event definition for a syslog or a trap in Prime Network based on user input.



Note

To create a script to add unsupported traps from a MIB, use the **vcb event view** command with the **-generatecli** option. To list the traps in a MIB that are supported and those that are not, use the **vcb event view** command with the **-mibfile** option. For more information, see [vcb event view, page B-34](#).

## Syntax

```

vcb event add -eventtype {syslog/trap} -eventname eventName [-alarmid alarmId]
[-subtype1 subtype1Name [-ticketable1]
[-severity1 critical/major/minor/warning/info/cleared>]
[-shortdesc1 short description string] [-autoclear1 false/true]
. . .
[-subtypen subtypenName] [-ticketablen] [-severityn
critical/major/minor/warning/info/cleared ]
[ -shortdescn short description string] [-autoclearn false/true]

-user username -password password

```

## Description

The **vcb event add** command creates an event definition in Prime Network based on the user input. Afterwards, the VNE-driver can create specific instances of this event for incoming traps or syslogs, persist them in the event database, and forward them to interested clients.

## Usage Example

```

vcb event add -eventtype syslog
-eventname "stack switch status syslog"
-subtype1 "stack switch removed syslog"
-severity1 minor
-subtype2 "stack switch added syslog"
-severity2 cleared
-user root -password admin
-faultnature ADAC
-faultcategory "QOS"

```

Adds a syslog event definition with the name “stack switch status syslog”. Two subtypes are added: “stack switch removed syslog” with severity minor and “stack switch added syslog” with severity cleared. By default, the subevents are not ticketable.

The syslog for which the event definition was added is:

```

STACKMGR-4-SWITCH_[ADDED|REMOVED]: Switch [dec] has been [ADDED to|REMOVED from] the
stack.

```

A device sends one syslog when a switch is added to a stacked device (clear alarm) and another syslog when a switch is removed from a stacked device (asserted minor alarm).

## Options

*Table B-34 Options and Arguments—vcb event add*

Option Argument	Description
eventtype <i>event type</i>	Type of event. Valid values are: <ul style="list-style-type: none"> <li>syslog</li> <li>trap</li> </ul>
eventname <i>event name</i>	Unique string that identifies the event within Prime Network.



Table B-34 Options and Arguments—*vcb event add* (continued)

Option Argument	Description
alarmid <i>alarm ID</i>	(Optional) Unique integer identifier for the event. <b>Note</b> Recommendation—Do not provide this argument; the VCB automatically generates a unique number.
subtype <sub>n</sub> <i>subtypen name</i>	Unique string that identifies the subevent with the Prime Network event.
ticketable <sub>n</sub>	(Optional) Optional parameter for subtype. If specified, indicates that a ticket should be generated for this subevent. By default, no ticket is generated for the subtype.
-autoclear <sub>n</sub> <i>false true</i>	(Optional) Optional parameter for subtype. If the event is ticketable, setting autoclear to false causes the subevent to remain asserted until the clear alarm arrives or the user manually acknowledges or clears the subevent. <b>Note</b> Root cause events are not autocleared even when autoclear is set to false. By default, autoclear is true for user-defined event definitions.
severity <sub>n</sub> <i>severity level</i>	Optional parameter for subtype. The severity of the subevent. Possible values are critical, major, minor, warning, info, and cleared. Info is the default severity value for a subevent.
shortdesc <sub>n</sub> <i>short description</i>	Optional parameter for subtype. A short description of the subevent. This string is stored in the event database. By default, the subtype value is used as the default shortdesc value.
-faultnature	Parameter that indicates how the fault is cleared, either manually or automatically. Possible values are: <ul style="list-style-type: none"> <li>ADAC (Automatically Detected Automatically Cleared) - The event is automatically detected and automatically cleared by the system. For example, “link down” event.</li> <li>ADMC (Automatically Detected Manually Cleared - The event must be manually cleared by the user. For example, “DWDM fatal error” syslog.</li> </ul>

Table B-34 Options and Arguments—*vcb event add* (continued)

Option Argument	Description
-faultcategory	Event category (3GPP standards). Possible values are: <ul style="list-style-type: none"> <li>• COMMUNICATIONS</li> <li>• QOS</li> <li>• PROCESSING</li> <li>• EQUIPMENT</li> <li>• ENVIRONMENTAL</li> <li>• UNDETERMINED</li> <li>• INTEGRITYVIOLATION</li> <li>• OPERATIONALVIOLATION</li> <li>• PHYSICALVIOLATION</li> <li>• SECURITYORSERVICEMECHANISMVIOATION</li> <li>• TIMEDOMAINVIOLATION</li> </ul>

**Note**

For the list of global options, see [Global Command Options, page B-2](#).

**Error Codes**Table B-35 Error Codes—*vcb event add*

Code	Description
201	Event name already exists in Prime Network.
202	Alarm ID already exists in Prime Network.

**Note**

For the list of general VCB error codes, see [General Error Codes, page B-2](#).

**vcb event view**

Use the **vcb event view** command to list event definition registrations.

**Syntax**

```
vcb event view -eventname {eventName | all} [-substringmatch] -user username
-password password -eventtype {trap | syslog | service}
vcb event view -genericevents all | trap | syslog [-ipaddress vneip] [-date yyyy-mm-dd]
[-time hh:mm:ss] [-maxrecords num] -user username -password password
vcb event view -mibfile complete-path-mibFilename [-generatecli -repository
ParsingrulesHive - group PatternsHive] -user username -password password
```

## Description

The **vcb event view** command enables you to view event definitions including event properties such as alarm ID, event subtypes, severity, and ticketability.

## Usage Examples

```
vcb event view -userdefined -eventname all -user root -password admin
```

Returns all the event definitions that were added to Prime Network using the VCB.

```
vcb event view -eventname bgp -substringmatch -user root -password admin
```

Returns all BGP event definitions in Prime Network, including those that were added using the VCB.

```
vcb event view -user root -password admin -eventname bgp -substringmatch -eventtype service
```

Returns all BGP service events.

```
vcb event view -mibfile /mibs/IF-MIB -user root -password admin
```

Returns lists of supported events and unsupported events based on the traps in the IF-MIB file.

```
vcb event view -mibfile /mibs/IF-MIB -generateeventcli -group cisco-trap-product-parsing-rules -repository cisco-trap-repository -user root -password admin
```

Creates, but does not run, a script /Main/VcbEventCommand.sh. The script contains three **vcb** commands for each unsupported trap; the commands add an event (and provide an event ID), event parsing rules, and an event pattern. Optionally, edit the script.

To run the script, change permissions on the file to ensure that it is executable and supply a username and password as input; see this example:

```
chmod 755 VcbEventCommand.sh
./VcbEventCommand.sh -user root -password admin
```



### Note

The Prime Network gateway maintains a known list of MIBs that are used to provide translation for trap varbinds when displayed in the UI. When an event is added from a MIB that is unknown to the gateway, the VCB does not add the MIB to the known MIB list. As a result, the varbinds for this trap might not be translated to user-friendly names.

## Options

**Table B-36** Options and Arguments—*vcb event view*

Option Argument	Description
eventname <i>eventName</i>	Unique string that represents the event. <b>Tip</b> Enter <b>all</b> as the <i>eventName</i> to display information on all the event definitions in Prime Network. Use this argument with caution because the number of events can potentially be very large.
substringmatch	(Optional) Indicates that the event name argument is not an exact match.
eventtype <i>trap   syslog   service</i>	Displays events of a specific type - traps, syslogs or service events.

Table B-36 Options and Arguments—*vcb event view* (continued)

Option Argument	Description
<code>generic</code> <i>events generic event type</i>	Displays all events from the Prime Network database (in raw format). <b>Tip</b> Enter <b>all</b> as the <i>generic event type</i> to display information on all the events in Prime Network.
<code>ipaddress</code> <i>neip</i>	(Optional) Generic events filter. IP address for the NE for which you want to see generic events.
<code>date</code> <i>yyyy-mm-dd</i>	(Optional) Generic events filter. The date after which the events arrived. (Returns events that arrived after the given day.)
<code>time</code> <i>hh:mm:ss</i>	(Optional) Generic events filter. The time after which the events arrived. (Returns events that arrived after the given time.)
<code>maxrecords</code> <i>num</i>	(Optional) Generic events filter. The maximum number of events that you want to display. The default value is 100.
<code>mibfile</code> <i>complete-path-mibFilename</i>	Loads MIB modules and compares the traps defined in the MIB against the events that are supported in Prime Network. Displays lists of supported traps and unsupported traps. <b>Note</b> Before using this command option, copy the MIB and dependent MIB files to a local folder. Rename each MIB file, removing the .my file extension from it.
<code>generatecli</code>	(Optional) When provided, produces a script, <i>NETWORKHOME/Main/VcbEventCommand.sh</i> . The script contains commands to add basic event support for each unsupported trap that was identified through the <b>-mibfile</b> option.
<code>repository</code> <i>ParsingrulesHive</i>	Mandatory <b>-generatecli</b> option. Hive that includes event parsing rules for traps.
<code>group</code> <i>PatternsHive</i>	Mandatory <b>-generatecli</b> option. Hive that includes event patterns for traps.

**Note**

For the list of global options, see [Global Command Options, page B-2](#).

**Error Codes**Table B-37 Error Codes—*vcb event view*

Code	Description
231	No such event exists in the events file

**Note**

For the list of general VCB error codes, see [General Error Codes, page B-2](#).

## vcb event modify

Use the **vcb event modify** command to modify events that were previously defined using the VCB or to modify event attributes for factory-defined events (by using the **-override** option). This command can also be used to drop events.

### Syntax

```
vcb event modify -eventname eventName [-alarmid alarmId] [-override]
{-subtype1 subtype1Name [-ticketable1] [-autoclear1 false/true]-severity1
critical/major/minor/warning/info/cleared
-shortdesc1 short description string..
{-subtypeN subtypeNName [-ticketableN] [-autoclear1 false/true]
-severityN critical/major/minor/warning/info/cleared
[-shortdescN] short description string -user username -password password
[-eventtype {trap | syslog | service | drop}
```

### Description

The **vcb event modify** command modifies an event definition in Prime Network. It can also be used to instruct the system to drop a specific event.



#### Note

Support for modifying an event is limited due to the complexity involved. When additional changes are required—such as changing the name of an event or a subtype—the supported procedure is to delete the entire event definition and add it afresh:

- Delete the event, event pattern, and associated event parsing rules.
- Add the event, event pattern, and event parsing rules.

### Usage Examples

```
vcb event modify -eventname "stack switch status syslog"
-subtype1 "stack switch removed syslog" -severity1 major -ticketable1 -user root
-password admin
```

Updates the event definition for the stack switch status syslog, changing the severity of the specified subtype to major and making the subtype ticketable. For a corresponding example of how this event was added, see [Usage Example, page B-32](#) for the **vcb event add**.

```
vcb event modify -eventname "bgp trap" -override -eventtype drop -user root -password admin
```

Drops the “bgp trap” event. This overrides the system-defined event.

### Options

*Table B-38 Options and Arguments—vcb event modify*

Option Argument	Description
eventname <i>eventName</i>	Unique string identifies the event within Prime Network.
eventtype <i>trap   syslog   service   drop</i>	Modifies an event of a specific type or instructs the system to drop an event.

Table B-38 Options and Arguments—*vcb event modify* (continued)

Option Argument	Description
alarmid <i>alarmId</i>	(Optional) Unique integer identifier for the event. If not provided, the VCB automatically generates a unique number. <b>Note</b> We recommend that you do not provide an input alarm ID.
subtypen <i>subtypeName</i>	Unique string that identifies the subevent with the Prime Network event.  To retain ticketability for any ticketable subtype—whether you want to modify the subtype or not—you must enter the subtype option and argument along with the ticketable option (below).
ticketable <i>n</i>	(Optional) Parameter for subtype. Indicates whether a ticket should be generated for this subtype. If not specified, no ticket is generated. <b>Note</b> To retain ticketability, supply the ticketable option for all subtypes that are currently defined as ticketable events (even for subtypes that you do not intend to modify). Otherwise, the subtypes are modified to be non-ticketable events.
-autoclear <i>n false true</i>	(Optional) Parameter for subtype. If the event is ticketable, setting autoclear to false causes the subevent to remain asserted until the clear alarm arrives or the user manually acknowledges or clears the subevent. <b>Note</b> Root cause events are not autocleared even when autoclear is set to false.  By default, autoclear is true for user-defined event definitions.
severity <i>n value</i>	(Optional) Parameter for subtype. Specifies the severity of the subevent. Possible values are critical, major, minor, warning, info, and cleared.
shortdesc <i>n short description string</i>	(Optional) Parameter for subtype. A short description.
override	(Optional) Indicates that you expect to override attributes for a factory-defined event.
-faultnature	Parameter that indicates how the fault is cleared, either manually or automatically. Possible values are: <ul style="list-style-type: none"> <li>ADAC (Automatically Detected Automatically Cleared) - The event is automatically detected and automatically cleared by the system. For example, “link down” event.</li> <li>ADMC (Automatically Detected Manually Cleared - The event must be manually cleared by the user. For example, “DWDM fatal error” syslog.</li> </ul>

Table B-38 Options and Arguments—*vcb event modify* (continued)

Option Argument	Description
-faultcategory	Event category (3GPP standards). Possible values are: <ul style="list-style-type: none"> <li>• COMMUNICATIONS</li> <li>• QOS</li> <li>• PROCESSING</li> <li>• EQUIPMENT</li> <li>• ENVIRONMENTAL</li> <li>• UNDETERMINED</li> <li>• INTEGRITYVIOLATION</li> <li>• OPERATIONALVIOLATION</li> <li>• PHYSICALVIOLATION</li> <li>• SECURITYORSERVICEMECHANISMVIOIATION</li> <li>• TIMEDOMAINVIOLATION</li> </ul>

**Note**

For the list of global options, see [Global Command Options, page B-2](#).

**Error Codes**Table B-39 Error Codes—*vcb event modify*

Code	Description
202	Alarm ID already exists in Prime Network.
251	Event name does not exist in Prime Network.
252	Event subtype name does not exist for the event.

**Note**

For the list of general VCB error codes, see [General Error Codes, page B-2](#).

**vcb event delete**

Use the **vcb event delete** command to delete an event definition. The **vcb event delete** does not delete or change the event template from which the event definition was cloned.

**Syntax**

```
vcb event delete -eventname eventName -user username -password password
```

## Description

The **vcb event delete** command removes event definitions created using the VCB and removes event attribute overrides from factory-defined events. Deleting a factory-defined event removes event attribute overrides only and not the event itself; the original event attributes are then applied to future events.

## Usage Example

```
vcb event delete -eventname "stack switch status syslog" -user root -password admin
```

Deletes the event definition. All registry entries added as a part of the event add command are removed from the site.xml file.

## Options

**Table B-40** Options and Arguments—*vcb event delete*

Argument	Description
eventname <i>eventName</i>	Name of the event to be deleted



### Note

For the list of global options, see [Global Command Options, page B-2](#).

## Error Codes

**Table B-41** Error Codes—*vcb event delete*

Code	Description
231	No such event exists in the events file



### Note

For the list of general VCB error codes, see [General Error Codes, page B-2](#).

## VCB CLI Reference: vcb eventparsingrules Commands

These topics provide reference information for the **vcb eventparsingrules** commands you can use to create, view, modify, and delete event parsing rules:

- [vcb eventparsingrules add, page B-41](#)
- [vcb eventparsingrules view, page B-43](#)
- [vcb eventparsingrules modify, page B-45](#)
- [vcb devicetype modify, page B-15](#)
- [vcb eventparsingrules delete, page B-46](#)



## vcb eventparsingrules add

Use the **vcb eventparsingrules add** command to create a VNE-driver registration for adding parsing rules to support a new trap or syslog by customizing a specified set of event templates.

### Syntax

```
vcb eventparsingrules add -templates templateName1, templateName2, ..., templateNameN
  -group repository -rulename rulename [-enable]
  { [-arg1 -arg1Value...-argN argNValue] } -user username -password password
```

### Description

The **vcb eventparsingrules add** command creates a VNE-driver event registration based on the templates chosen by the user. This enables Prime Network to identify and associate the event to a particular device component instead of classifying the event as a generic event.

The command does the following:

- Creates a separate registry configuration (a copy) for customizing the event. Parameters that you input using the command affect the copy.
- Creates rules for handling the event based on the event template and user input.
- Updates the site.xml file, so that Prime Network can differentiate customizations created using the VCB from changes supplied in VNE-driver registration files.

### Usage Example

```
vcb eventparsingrules add -enable
  -templates
  syslog-identification, syslog-subtype-from-expression, create-managedelement-key, create-ana-syslog-event
  -group cisco-syslog-repository
  -rulename stack-switch-status-syslog
  -syslog_identification_testmessage "STACKMGR-4-SWITCH_ADDED: Switch 2 has been added to the stack"
  -syslog_identification_expression "STACKMGR-4-SWITCH_%%subtypekey%%: Switch %%uniqueid%% has been ."
  -syslog_subtype_from_expression_replacing_rules "ADDED-stack switch added syslog, REMOVED-stack switch removed syslog"
  -create_ana_syslog_event_type "stack switch status syslog"
  -user root -password admin
```

Adds parsing rules to identify the syslog correctly, associates it with the correct device component, and creates the corresponding Prime Network event and subevent.

Four event templates are entered:

- *syslog-identification*—Rules that pertain to syslog identification.
- *syslog-subtype-from-expression*—Rules that map the syslog values (in this case, ADDED and REMOVED) to the event subtype names: stack switch added syslog, stack switch removed syslog. The example includes two rules, separated by commas.



**Note** Rules must be comma-separated. Each rule must include a value and event subtype, separated by a hyphen: value-event subtype.

- create-managedelement-key—Indicates that the syslog should be associated with the managed element. (There are no input parameters for this template.)
- create-ana-syslog-event—Provides rules for creating instances of the corresponding Prime Network event (defined with the **vcb event add** command).

Input parameters for the event templates are variable arguments that depend on the templates selected.

- syslog\_identification\_expression—The actual syslog message with input that is of interest to the user and is masked with special keys, such as %%subtypekey%%, %%uniqueid%%, and %%entityid%%, depending on which is applicable. In the previous example, only subtypekey and uniqueid parameters are relevant.



**Note** Only a substring of the message is used in the example, because whatever comes afterward is of no interest to the user.

- syslog\_subtype\_from\_expression\_replacing\_rules—Specifies the mapping from the subtypekey to the subevent name. The subevent string should exactly match one of the subevent names that was defined using the **vcb event add** command.
- create\_ana\_syslog\_event\_type—Specifies the event name.



**Note** This parameter should exactly match the event name defined using the **vcb event add** command.

## Options

**Table B-42** Options and Arguments—*vcb eventparsingrules add*

Option Argument	Description
templates <i>template name1, ... template namen</i>	Comma-separated list of event template names. Event templates are divided into categories that correspond to the function they fulfill (identification, association, and so on.) Depending upon the trap or syslog that you are adding, select no more than one template from each template category.
group <i>repository</i>	Specifies the vendor-specific trap or syslog repository file under which the customizations should be made.
rule <i>rulename</i>	String that is used as a key name for event rule definition.
enable	(Optional) Indicates whether the rule is enabled or disabled. Only enabled rules are used to parse incoming traps and syslogs.
variable arguments	Arguments vary from one event template to another event template.
syslog_identification_ testmessage	(Optional) Parameter valid for syslogs only. An example syslog message used to check the correctness of the regular expression that the VCB creates automatically based on user input.



**Note** For the list of global options, see [Global Command Options, page B-2](#).

## Error Codes

**Table B-43** Error Codes—*vcb eventparsingrules add*

Code	Description
211	Event template file not found.
103	No such template name in template file.
212	Only one template can be selected from each template category.
213	Invalid expression for syslog.



**Note** For the list of general VCB error codes, see [General Error Codes, page B-2](#).

## vcb eventparsingrules view

The **vcb eventparsingrules view** command displays event registrations. Use it to verify that you successfully added an event parsing rule or to view parameters (to fill them in based on the example).

### Syntax

```
vcb eventparsingrules view -group repository name -rulename { rulename | all } [-detail]
[-userdefined] -user username -password password
vcb eventparsingrules view -template templateName | all -inputparam -user username
-password password
```

### Description

The **vcb eventparsingrules view** command shows configuration settings. Use it to list:

- Details of the events repository for a particular rulename or for all the events in the file. The information displayed includes the parsing rules and important parameters in each rule.
- Input parameters in the event template specified by -template option. If all is specified, the user input parameters of all the templates are displayed.

### Usage Examples

#### Example 1

```
vcb eventparsingrules view -template syslog-identification -inputparam -user root
-password admin
```

Displays the syslog-identification event template definition, including a detailed description of the input parameters required when using the template to add event parsing rules.

**Example 2**

```
vcb eventparsingrules view -group cisco-syslog-repository -userdefined -rulename all
-user root -password admin
```

Displays all event parsing rules that were defined using the VCB under the cisco-syslog-repository hive.

**Example 3**

```
vcb eventparsingrules view -group cisco-syslog-repository
-rulename stack-switch-status-syslog -user root -password admin
```

Displays the event parsing rules for the event “stack-switch-status-syslog” which was created using the VCB.

**Example 4**

```
vcb eventparsingrules view -group cisco-syslog-repository -rulename all -user root
-password admin
```

Displays all the event parsing rules present in the hive cisco-syslog-repository, including those added using the VCB.

**Options**

**Table B-44** Options and Arguments—*vcb eventparsingrules view*

Option Argument	Description
group <i>repository name</i>	The trap or syslog repository filename.
rulename <i>ruleName</i>	The unique string that is used to represent the event parsing rules. <b>Tip</b> Enter <b>all</b> as the <i>ruleName</i> to display information on all the rules in Prime Network.
detail	(Optional) Lists the entire rule contents including the parsing rule entry details.
template <i>templateName</i>	The event template name. <b>Tip</b> Enter <b>all</b> as the <i>templateName</i> to display information on all event templates in Prime Network.
inputparam	(Optional) Lists template definition entries that require user input when creating event parsing rules.

**Note**

For the list of global options, see [Global Command Options, page B-2](#).

**Error Codes**

**Table B-45** Error Codes—*vcb eventparsingrules view*

Code	Description
103	No such template name in the templates file.
222	Parsing rules repository not found.
231	No such rule name in the site.xml.

## vcb eventparsingrules modify

Use the **vcb eventparsingrules modify** to modify the parsing rule definitions. The most common use case for this command is to select one or more different templates because the certification of the customization failed.

### Syntax

```
vcb eventparsingrules modify -templates templateName1, templateName2, ..., templateNameN
-group repository -rulename rulename [-enable] { [-arg1 -arg1Value...-argN argNValue] } -user
username -password password
```

### Description

The **vcb eventparsingrules modify** command changes parsing rule definitions based on the templates chosen by the user. The command can also be used to add parsing rules that were inadvertently omitted when adding the parsing rule. For example, use the command to add the rules for extracting the uniqueid parameter.

### Options

*Table B-46 Options and Arguments—vcb eventparsingrules modify*

Option Argument	Description
templates <i>template name1, ... template nameN</i>	Comma-separated list of event template names. Event templates are divided into categories that correspond to the function they fulfill (identification, association, and so on.) Depending upon the trap or syslog that you are adding, select no more than one template from each template category.
group <i>repository</i>	The hive under which the customizations should be made. The hive is the vendor-specific trap or syslog repository file.
rulename <i>ruleName</i>	String that is used as a key name for event rule definition.
enable	(Optional) Indicates whether the rule should be enabled or disabled. Only enabled rules are used to parse incoming traps and syslogs.
variable arguments	Each event template can require different input and a different number of input parameters from none to more than one. See <a href="#">Event Templates Input Summary—Required and Optional Input</a> , page B-83.



#### Note

**vcb eventparsingrules modify** command is not supported in Prime Network. To modify the parsing rules use *repository* and *rule name* option. For the list of global options, see [Global Command Options](#), page B-2.

## Error Codes

**Table B-47** Error Codes—*vcb eventparsingrules modify*

Code	Description
103	No such template name in the templates file.
211	Event template file not found.
212	Only one template can be selected from each template category.



### Note

For the list of general VCB error codes, see [General Error Codes, page B-2](#).

## vcb eventparsingrules delete

Use the **vcb eventparsingrules delete** command to delete the parsing rule definitions of an event. Doing so does not delete or change the event template from which that event definition was cloned.

### Syntax

```
vcb eventparsingrules delete -group repository hive -rulename rulename -user username
                             -password password
```

### Description

The **vcb eventparsingrules delete** command removes event parsing rule definitions created from an event template. It does not change or delete the event template itself.

### Usage Example

```
vcb eventparsingrules delete -group cisco-syslog-repository
                             -rulename stack-new-master-syslog
```

This example deletes the stack-new-master-syslog rule from the cisco-syslog-repository hive.

### Options

**Table B-48** Options and Arguments—*vcb eventparsingrules delete*

Option Argument	Description
group <i>repository hive</i>	The hive from which to remove the event parsing rule.
rulename <i>ruleName</i>	The rule to delete.



### Note

For the list of global options, see [Global Command Options, page B-2](#).

## Error Codes

*Table B-49 Error Codes—vcb eventparsingrules delete*

Code	Description
222	Parsing rules repository not found.
241	No such rule name in the site.xml.



Note

For the list of general VCB error codes, see [General Error Codes, page B-2](#).

## VCB CLI Reference: vcb eventpattern Commands

These topics provide reference information for the **vcb eventpattern** commands you can use to create, view, modify, and delete event patterns:

- [vcb eventpattern add, page B-47](#)
- [vcb eventpattern view, page B-49](#)
- [vcb eventpattern modify, page B-50](#)
- [vcb eventpattern delete, page B-52](#)

### vcb eventpattern add

Use the **vcb eventpattern add** command to create a VNE-driver registration that points from the parsing rules hive, which is scheme or VNE-specific, to the parsing rules defined in the repository file.



Note

Only those events that have this pointer are deemed as supported events. Other events are deemed generic events despite having parsing rules and event definitions.

### Syntax

```
vcb eventpattern add [-patternid patternId] -group parsing rules hive
-repository parsing rules repository hive -rulename rulename -user username
-password password
```

### Description

The **vcb eventpattern add** command creates a pointer from the parsing-rules hive to the repository where the actual parsing rules are defined.

### Usage Examples

```
vcb eventpattern add
-patternid 202 -group cisco-syslog-product-parsing-rules
-repository cisco-syslog-repository
-rulename stack-switch-status-syslog -user username -password password
```

Adds a pointer from the parsing rules file to the actual definitions in the parsing-rules hive with pattern ID 202. It points to the key (rule) named stack-switch-status-syslog in the cisco-syslog-repository file.

## Options

**Table B-50** Options and Arguments—*vcb eventpattern add*

Option Argument	Description
patternid <i>patternId</i>	(Optional) (Recommendation: do not provide.) Unique integer to identify the supported event to VNEs. If not provided, VCB generates this number automatically.  <b>Note</b> Omitting this option and argument enables the VCB to ensure that the patternid is unique and that it does not overlap with other file definitions due to registry inheritance.
group <i>parsing rules hive</i>	The hive to which this pattern should be added. The parsing-rules hives are generally scheme-specific. Device type-specific definitions can also be made.
repository <i>parsing rules repository hive</i>	The trap or syslog repository where the actual parsing rules are defined.  <b>Note</b> Enter the same hive that was specified when creating parsing rules registrations using the <b>vcb eventparsingrules add</b> command.
rulename <i>rulename</i>	String that is used as a key name for event rule definition.  <b>Note</b> Enter exactly the same string as the one that was specified when creating parsing rules registrations using the <b>vcb eventparsingrules add</b> command.



**Note**

For the list of global options, see [Global Command Options, page B-2](#).

## Error Codes

**Table B-51** Error Codes—*vcb event pattern add*

Code	Description
221	Parsing rules hive not found.
222	Parsing rules repository not found.



**Note**

For the list of general VCB error codes, see [General Error Codes, page B-2](#).



## vcb eventpattern view

Use the **vcb eventpattern view** command to display event registrations. It is useful when you need to verify successful completion of an add command or to help find a similar case for filling in parameters on other commands.

### Syntax

```
vcb eventpattern view -group parsingrules hive -rulename { rulename | all }  
[-substringmatch] [-full] -user username -password password
```

### Description

The **vcb eventpattern view** command shows the actual set of events that are supported by a particular NE type or scheme. It displays the pattern ID and the repository file where the event parsing rules are defined. When the substringmatch option is used, only rules that contain a certain substring are displayed; use this option, for example, to obtain rules for a technology name such as MPLS.

### Usage Examples

#### Example 1

```
vcb eventpattern view -group cisco-syslog-parsing-rules -rulename  
stack-switch-status-syslog -user root -password admin
```

Displays the event pattern definition for the specified rulename; that is, the pattern ID, and the pattern is pointing to the parsing rules repository.

#### Example 2

```
vcb eventpattern view -group cisco-syslog-parsing-rules -rulename all -user root  
-password admin
```

This example shows all the event pattern definitions in the specified hive.

#### Example 3

```
vcb eventpattern view -group cisco-syslog-parsing-rules -userdefined  
-rulename bgp -substringmatch -full -user root -password admin
```

This example shows the entire event definition for all BGP events (including those defined using the VCB) defined in the cisco-syslog-parsing-rules hive. The following information is displayed:

- Pattern definitions—Parsing rules repository, pattern ID
- Parsing rules definitions—All rules in the definition that require user input, and the values set for these parameters
- Event definitions—Event attributes such as eventname, subevent names, ticketability, severity and so on

## Options

**Table B-52** Options and Arguments—*vcb eventpattern* view

Option Argument	Description
<code>group parsingrules hive</code>	The parsing-rules filename used by the NE type or scheme.
<code>rulename rule name</code>	Unique string that represents the event parsing rules. <b>Tip</b> Enter <b>all</b> as the <i>rule name</i> to list all event parsing rules defined in the repository.
<code>substringmatch</code>	(Optional) Indicates that the rule name provided is not an exact match. This option is useful when you want to know the names of all rules that belong to a particular technology, such as BGP.
<code>full</code>	(Optional) Displays the entire details of the event, from the pattern definition and parsing rules to the event definition. Provides the complete picture of the how an event is supported in Prime Network. <b>Note</b> Avoid this option when using the “all” argument because it can result in a very large output.



### Note

For the list of global options, see [Global Command Options, page B-2](#).

## Error Codes

**Table B-53** Error Codes—*vcb eventpattern* view

Code	Description
221	Parsing rules hive not found.
241	No such rule name in the site.xml.



### Note

For the list of general VCB error codes, see [General Error Codes, page B-2](#).

## `vcb eventpattern modify`

Use the `vcb eventpattern modify` command to modify the pointer to the parsing rules.

### Syntax

```
vcb eventpattern modify -patternid patternId -group parsing rules hive [-repository
parsing rules repository hive] [-rulename ruleName] -user username -password password
```

### Description

The `vcb eventpattern modify` command modifies the pointer from the parsing-rules hive to the repository where the actual parsing rules are defined.

## Usage Examples

```
vcb eventpattern modify
  -patternid 202
  -group cisco-syslog-product-parsing-rules
  -repository cisco-router-syslog-repository -user root -password admin
```

This example assumes that we are starting with the eventpattern with ID 202 that points to the cisco-syslog-repository (as shown in [Usage Examples](#) for the **vcb eventpattern add** command). In this example, we modify the repository for the eventpattern with ID 202 to the cisco-router-syslog-repository.



**Note** **vcb eventpattern modify** cannot be performed on the **group** as this is the identifier for the pattern. You can modify only the **repository** or **rule name**.

## Options

**Table B-54** Options and Arguments—*vcb eventpattern modify*

Option Argument	Description
patternid <i>patternId</i>	Unique integer to identify the supported event to a VNE.
group <i>parsing rules hive</i>	The hive in which this pattern is to be modified. The parsing-rules hives are generally scheme-specific. VNE-specific definitions can also be made using this hive.
repository <i>parsing rules repository hive</i>	(Optional) The hive where the actual parsing rules are defined (the trap/syslog repository).  Enter the same hive that was specified when creating parsing rules registrations using the <b>vcb eventparsingrules add</b> command.
rulename <i>ruleName</i>	String that is used as a key name for event rule definition.  <b>Note</b> Enter exactly the same string as the one that was specified when creating parsing rules registrations using the <b>vcb eventparsingrules add</b> command.



**Note** For the list of global options, see [Global Command Options, page B-2](#).

## Error Codes

**Table B-55** Error Codes—*vcb eventpattern modify*

Code	Description
221	Parsing rules hive not found.
222	Parsing rules repository not found.
271	Pattern with ID not found

**Note**

For the list of general VCB error codes, see [General Error Codes, page B-2](#).

## vcb eventpattern delete

Use the **vcb eventpattern delete** command to delete the parsing rule from the list of supported event patterns. Doing so does not delete the parsing rules in the repository file.

### Syntax

```
vcb eventpattern delete -group parsing rule hive -patternid pattern ID -user username
                        -password password
```

### Description

The **vcb eventpattern delete** command removes the pointer to the parsing rule defined in the repository file.

### Usage Examples

```
vcb eventpattern delete -group cisco-syslog-parsing-rules -patternid 202 -user root
                        -password admin
```

Deletes the parsing rules pattern with ID 202. All registry entries added as a part of the **vcb eventpattern add** command will be removed from site.xml.

### Options

*Table B-56 Options and Arguments—vcb eventpattern delete*

Argument	Description
patternid <i>pattern ID</i>	Unique integer to identify the supported event to a VNE.
group <i>parsing rules hive</i>	The hive in which this pattern is to be modified. The parsing-rules hives are generally scheme-specific. VNE-specific definitions can also be made using this hive.

**Note**

For the list of global options, see [Global Command Options, page B-2](#).

### Error Codes

*Table B-57 Error Codes—vcb event pattern delete*

Code	Description
221	Parsing rules hive not found.
271	Pattern with ID not found



**Note** For the list of general VCB error codes, see [General Error Codes, page B-2](#).

## VCB CLI Reference: `vcb eventarg` Command

### `vcb eventarg view`

Use the `vcb eventarg view` command to display event parsing rule arguments and descriptions.

#### Syntax

```
vcb eventarg view -user username -password password
```

#### Description

The `vcb eventarg view` command option displays all the VCB event parsing rules template variable arguments along with descriptions.

## Troubleshooting Event Customization Using the VCB CLI

Errors that you receive from the VCB CLI are self-explanatory. Most errors make very clear what you need to do to correct the problem that has occurred. For example, if an event name already exists, you must enter a different event name. If an alarm ID or pattern ID is already in use, you should omit the related option and argument from your command and allow the VCB to generate a unique ID for you.



**Note** To get more information, add the `-debug` option to any `vcb` command; for more information, see [Global Command Options, page B-2](#).

Errors that occur in the server are not as obvious. If a newly supported event does not appear in Prime Network Events or Prime Network Vision, you need to perform some troubleshooting, as follows:

- Step 1** Ensure that the device is configured to send events to Prime Network gateway. Use any tool to snoop and check whether the simulated network events that you are sending are actually arriving at the Prime Network gateway. If not, fix the issue whether it is connectivity, firewall and so on, then proceed to next step.

For detailed information, see [VCB CLI Reference: Events Commands, page B-31](#).

- Step 2** Enable debug for event processing in the VNE. Execute the following commands for the AVM that has the VNE that you will be testing.

```
runRegTool.sh -gs localhost set 127.0.0.1
avm<avmid>/services/logger/log4j.category.com.sheer.metrocentral.framework.eventapplicatio
n.eventcorrelation.SendAlarmMessageUtil DEBUG
runRegTool.sh -gs localhost set 127.0.0.1
avm<avmid>/services/logger/log4j.category.com.sheer.metrocentral.framework.eventmanager.Ev
entManager DEBUG
runRegTool.sh -gs localhost set 127.0.0.1
avm<avmid>/services/logger/log4j.category.com.sheer.metrocentral.framework.eventapplicatio
n.parsing.ParsingApplication DEBUG
```

- Step 3** Allow the VNE to come up, then open the log file for the AVM. Check whether the newly added pattern is being loaded at VNE startup. Look for an entry in the log file that is similar to the following text:

```
DEBUG [06 21 2010 12:19:59.524 IST] - ParsingApplication.buildRulesMatrix - pattern
ATMLC-6-CLOCKING with index 5001 in the registry is now mapped into index 123 in the
parsing application.
```

The above DEBUG statement includes both the rulename (ATMLC-6-CLOCKING with) and the pattern id (5001) of the newly added event. If a similar statement is not printed for the newly added event, go to [Step 4](#); otherwise, go to [Step 5](#).

- Step 4** Review the **vcb eventparsingrules add** command that you used, checking whether you enabled the event using the **-enable** option. If the command was issued without the **-enable** option, delete the event parsing rules using the **vcb eventparsingrules delete** command and add the event parsing rules again, ensuring that you use with the **-enable** option.

- Step 5** Check statically whether the links between event pattern, event parsing rules, and event are OK. To perform this check, use the **vcb eventpattern view** command with the **-full** option (see [Example 3, page B-49](#)). The output should display details of all the three customizations. A typo in the rulename, event type name, or event subtype name can prevent the links from being established and result in a partial display. For example, if the rulename in the **vcb eventpattern** command does not match that used in the **vcb eventparsingrules** command, only event pattern details will be displayed; details for event parsing rules and the event will not be displayed.

If the output is OK (that is, it includes details for all three customizations), go to [Step 7](#). Otherwise, go to [Step 6](#).

- Step 6** Review the commands that have been issued and re-add or modify the customizations as required. Then go back to [Step 5](#).

- Step 7** After the static verification that you perform in step 5 succeeds, check whether the parsing itself is failing. Put a tail on the AVM log file and resend the simulated event. When parsing fails, the event is classed as a generic event. Log output similar to the following will appear.

```
DEBUG [06 21 2010 16:11:09.623 IST] - EventManager.filterEventApplications - Event
has been dropped by application
[com.sheer.metrocentral.framework.eventapplication.filter.GenericSyslogTypeFilterAp
pl]
##### com.sheer.metrocentral.framework.eventapplication.types.EventData
#####
# Id          : = 137611826381_1277116869542
# Unique source ID: = null
# Type        : = generic syslog
# SubType     : = generic syslog
# SourceOID   : = { [ManagedElement(Key=10.77.212.205)] [Syslog] }
# Event Time  : = 1277116869542
# Info        : = 7.212.205 %FAN-3-FAN_OK: Fan 3 had earlier reported a rotation
error. It is ok now
# CorrelationKeys: =
#           CK=(MC.DA-10.77.212.205)-25:52:0:0 [16]
# Adjacent XID : = null
# Source IP interface: = null

#####
```

**Step 8** Open the log file and go backwards from the end of the file until you come to the place where logs pertaining to the actual parsing process are available. Search for the string 'Testing pattern: handle *rulename*', where *rulename* is the string you used in the **vcb eventpattern add** command. Here you will find logs that report the results of testing each rule. Identify the rule that failed as shown in the following log.

```
DEBUG [06 21 2010 16:11:09.622 IST] - ParsingApplication.processEvent - Exception
during parsing correlation rules, at pattern-125, rule-2

Stack: [(uniqueid=>3), (syslog=>7.212.205 %FAN-3-FAN_OK: Fan 3 had earlier reported a
rotation error. It is ok now), (subtypekey=>0K), (.prulescache=>[]), (counter=>0)]

Event Data :

##### com.sheer.metrocentral.framework.eventapplication.parsing.types
.RawSyslogEventData #####
# Id          : = 4311876356_1277116869490
# Unique source ID: = null
# Type       : = raw event
# SubType   : = raw syslog
# SourceOID  : = null
# Event Time : = 1277116869494
# Info      : = 7.212.205 %FAN-3-FAN_OK: Fan 3 had earlier reported a rotation
error. It is ok now
# syslog = 7.212.205 %FAN-3-FAN_OK: Fan 3 had earlier reported a rotation error. It
is ok now

#####
#####
java.lang.reflect.InvocationTargetException

    at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method) at
sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl
.java:39)
    at
sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAcce
ssorImpl.java:27)
    at java.lang.reflect.Constructor.newInstance(Constructor.java:513)
    at
com.sheer.metrocentral.framework.correlation.parsing.ChangeArgumentValue.execute
(ChangeArgumentValue.java:68)
```

In the above example, the parsing rule that failed is `ChangeArgumentValue`. The failure implies that the replacing rules that map the network event parameters to the Prime Network event subtypes are failing. Review the `replacing_rules` arguments used in the **vcb eventparsingrules** command and make the necessary changes.

The list of parsing rules (classes) and the corresponding option in `vcb` are given in the following table. Review the parameter values of the failing option and make appropriate changes.

Repeat the above steps until all errors are resolved and the event is parsed correctly and the Prime Network event is generated as expected.

# VCB Template Reference

Use the information in this section to determine which template best matches the device, module, or event that you want to manage with Prime Network. Information in this section is also useful when you test the customizations that you have made.

Topics include:

- [U-VNE Templates, page B-56](#)
- [Module Templates, page B-65](#)
- [Event Templates, page B-80](#)

## U-VNE Templates

Features, advantages, and limitations of template-based U-VNEs are template-dependent. The GenericUVNE template uses the same set of MIB-II based instrumentation for logical inventory discovery as is used by the Prime Network Generic SNMP VNE. The advantage of the U-VNE—created using the VCB and the GenericUVNE template—over the Generic SNMP VNE is that you can identify the device type for the U-VNE and further extend the U-VNE for additional event recognition using the VCB.

For more information, see the following sections:

- [Methods for Creating U-VNEs—Overview, page 3-7](#)
- [Comparison of Generic SNMP VNEs, U-VNEs, and Developed VNEs, page 3-3](#)
- [GenericUVNE Template, page B-56](#)

## GenericUVNE Template

The GenericUVNE template is applicable to Cisco and non-Cisco NEs and supports event customization with event association to Managed Element only.

Use the GenericUVNE template to model any NE that is not currently supported by Prime Network. A U-VNE created this way is very similar to the Generic SNMP VNE. It provides basic information, such as the physical interfaces available on the device and their status, rudimentary logical modeling, and parsing of basic traps; see [GenericUVNE—Supported Traps, page B-59](#). Using the VCB, however, you can configure additional traps and syslog recognition for a U-VNE created using the GenericUVNE template.

This U-VNE models NEs using SNMP MIB-II, which is the most generic and widely used management interface. This U-VNE does not consider the device vendor, device type, or software version of the NE that it models. Using the VCB, however, you can update device type attributes for a U-VNE created using the GenericUVNE template.

[Table B-58](#) summarize the features and advantages of the GenericUVNE template.



Table B-58 GenericUVNE Summary

Features	Advantages	Limitations
<ul style="list-style-type: none"> <li>• Same physical and logical inventory as a Generic SNMP VNE.</li> <li>• Applicable to Cisco and non-Cisco NEs.</li> <li>• User-defined device type attributes: <ul style="list-style-type: none"> <li>– Device category—Determines the icon that is displayed.</li> <li>– Element type.</li> </ul> </li> <li>• Event recognition, enabling Prime Network to forward events from unsupported devices to OSS applications.</li> </ul>	<p>When compared with a Generic SNMP VNE, this U-VNE provides:</p> <ul style="list-style-type: none"> <li>• Simplified trap and syslog recognition (using the VCB).</li> <li>• Application of soft properties to a specific U-VNE (as opposed to a device type).</li> </ul>	<p>Event association to Managed Element only</p>

**Note**

Expedite Legend—The Expedited column in the service event tables in this chapter can contain these values:

Y—Indicates that the service event is expedited by a syslog or trap generated by the device. This means that the syslog or trap causes the VNE to poll the device without waiting for the usual polling cycle, thus enabling quicker detection of the event.

N—Indicates that the service event is not expedited. The service event is not expedited. This means that the VNE will poll this device during the next regularly scheduled polling cycle.

**Note**

In some of the following tables, attributes, protocols, technologies, etc. are listed as supported. Supported denotes that SNMP queries are made to the NE for those attributes, etc. Whether values are available in response to the queries depends on whether the instrumentation supported in the NE works.

See the following sections:

- [GenericUVNE—Physical Inventory Model, page B-58](#)
- [GenericUVNE—Logical Inventory Model, page B-58](#)
- [GenericUVNE—Supported Traps, page B-59](#)
- [GenericUVNE—Supported Events, page B-59](#)
- [GenericUVNE—Limitations, page B-60](#)
- [GenericUVNE—Supported Topologies, page B-60](#)
- [GenericUVNEs—Supported Technologies, page B-60](#)
- [GenericUVNE—Supported Service Events, page B-65](#)

## GenericUVNE—Physical Inventory Model

A U-VNE created using the GenericUVNE template uses a static model for the device chassis. The rest of the physical inventory is modeled using the ifTable. Since modules are not modeled, this U-VNE creates a single generic module on which all of the physical interfaces reside.

[Table B-59](#) describes which MIB tables are used to model the physical inventory components that are supported by a U-VNE created using the GenericUVNE template.

**Table B-59** MIBs Used for Physical Inventory Model of GenericUVNE

Logical Component	MIB Table	Columns/Tables Used For Modeling
Interfaces	ifTable	<ul style="list-style-type: none"> <li>ifDescr</li> <li>ifType</li> <li>ifOperStatus</li> </ul>
Ports		<ul style="list-style-type: none"> <li>ifOperStatus and ifAdminStatus</li> </ul>
Port status	ifTable	<ul style="list-style-type: none"> <li>ifSpeed</li> </ul>
Port speed	ifTable	<ul style="list-style-type: none"> <li>ifPhysAddress (Ethernet ports)</li> </ul>
MAC address	ifTable	



### Note

Certain general properties on the managed element, such as system description, are modeled using the RFC1213-MIB.

## GenericUVNE—Logical Inventory Model

[Table B-60](#) describes which MIB tables are used to model the logical inventory components that are supported by a U-VNE created using the GenericUVNE template. Attributes in [Table B-60](#) are taken from MIB-II.

**Table B-60** MIBs Used for Logical Inventory Model of GenericUVNE

Logical Component	MIB Table	Columns/Tables Used For Modeling
IP Interfaces	ipAddrTable	<ul style="list-style-type: none"> <li>ipAdEntIfIndex</li> <li>ipAdEntNetMask</li> </ul>
ARP table	ipNetToMediaTable	<ul style="list-style-type: none"> <li>ipNetToMediaPhysAddress</li> <li>ipNetToMediaType</li> </ul>
Routing table	ipRouteTable	<ul style="list-style-type: none"> <li>ipRouteDest</li> <li>ipRouteIfIndex</li> <li>ipRouteNextHop</li> <li>ipRouteType</li> <li>ipRouteMask</li> </ul>
Bridging table	dot1dTpFdbTable	—
Default bridge	dot1dBridge	<ul style="list-style-type: none"> <li>dot1dBaseBridgeAddress</li> <li>dot1dBaseType</li> </ul>

## GenericUVNE—Supported Traps

A U-VNE created using the GenericUVNE template can parse the standard MIB-II and Bridge-MIB traps listed in [Table B-61](#).

*Table B-61 Supported Traps for GenericUVNE*

Standard MIB-II Traps	
authenticationFailure	mplsTunnelReoptimized
bgpBackwardTransition	mplsTunnelRerouted
bgpEstablished	mplsTunnelUp
coldStart	ospfIfAuthFailure
entConfigChange	ospfIfConfigError
linkDown	ospfIfRxBadPacket
linkUp	ospfIfStateChange (down)
mplsL3VpnVrfDown	ospfIfStateChange (up)
mplsL3VpnVrfNumVrfRouteMaxThreshExceeded	ospfMaxAgeLsa
mplsL3VpnVrfRouteMidThreshExceeded	ospfNbrStateChange (down)
mplsL3VpnVrfUp	ospfNbrStateChange (up)
mplsLdpInitSessionThresholdExceeded	ospf-if-packet-retransmit
mplsLdpSessionDown	ospfOriginateLsa
mplsLdpSessionUp	ospfTxRetransmit
mplsTunnelDown	warmStart
Bridge-MIB Traps	
dot1dBaseBridgeAddress	dot1dBaseType

A U-VNE created using the GenericUVNE template can identify traps, but it cannot correlate them. This is because this U-VNE does not include the model entities required by higher trap parsing levels.

For example, if Prime Network receives an mplsTunnelDown trap from a device modeled with the GenericUVNE template, Prime Network can identify the Tunnel Down trap, but it cannot perform correlation on the trap. The reason is that this U-VNE does not investigate tunnels, which means that there is no Device Component in the model to which Prime Network can attach a correlation flow.

For this U-VNE, event association is always to the **Managed Element**.

## GenericUVNE—Supported Events

A U-VNE created using the GenericUVNE template supports the service events listed in [Table B-62](#).

*Table B-62 Supported Service Events for GenericUVNE*

Event Name	Supported	Expedited
Device Unreachable	Y	N
Discard Packets	Y	N

**Table B-62** *Supported Service Events for GenericUVNE (continued)*

Event Name	Supported	Expedited
Dropped Packets	Y	N
Port Flapping	Y	N
Port Down	Y	N

### GenericUVNE—Limitations

A U-VNE created using the GenericUVNE template uses MIB2 to cover the widest possible range of NEs. Although MIB2 is a widely accepted industry standard, most network equipment vendors augment MIB2 with other Management Interfaces such as private MIBs, Telnet, XML, and so on. In addition, different vendors sometimes have different implementations of standard MIBs. As a result, even the limited model created by this U-VNE is dependent on the vendor's adherence to general network management standards.

### GenericUVNE—Supported Topologies

A U-VNE created using the GenericUVNE template supports the topologies listed in [Table B-63](#).

**Table B-63** *Supported Topologies for GenericUVNE*

Topology Type	Link Type	Supported
Ethernet	Ethernet	Y
Physical Layer	Ethernet	Y

### GenericUVNEs—Supported Technologies

The following sections list the objects and attributes that are recognized on a U-VNE created using the GenericUVNE template, per technology:

- [IP](#), page B-61
- [Ethernet \(IEEE 802.3\)](#), page B-62
- [Base Logical Components](#), page B-63
- [Common](#), page B-64

**IP**

[Table B-64](#) lists the IP attribute support on a U-VNE created using the GenericUVNE template.



**Note**

[Table B-64](#) includes the supported technologies only.

**Table B-64** IP Attribute Support on GenericUVNEs

Attribute	Supported
<b>IMO Name—IIPInterface</b>	
IP Address	Y
Subnetwork Mask	Y
IP Interface Addresses Array	
Interface Name	
Interface Description	Y
IP Interface State	Y
OSPF Interface Cost	
Broadcast Address	
MTU	
Lookup Method	
Address Resolution Type	
ARP Timeout	
Secured ARP	
ICMP Mask Reply	
IGMP Proxy	
HSRP Groups	
IP Multiplexing Table	
IANA Type	
Containing CTPs	
Contained CTPs	
<b>IMO Name—IRoutingEntity</b>	
Routing Table	Y
ARP Entity	Y
Routing Table Changes	
Name	Y
Logical Sons	Y

**Table B-64** IP Attribute Support on GenericUVNEs (continued)

Attribute	Supported
<b>IMO Name—IRoutingEntry</b>	
Destination IP Subnet	Y
Next Hop IP Address	Y
Type	Y
Routing Protocol Type	Y
Outgoing Interface Name	Y
<b>IMO Name—IARPEntity</b>	
ARP Table	Y
<b>IMO Name—IARPEntity</b>	
IP Address	Y
MAC Address	Y
Port	Y
Entry Type	Y

**Ethernet (IEEE 802.3)**

[Table B-65](#) lists the Ethernet (IEEE 802.3) attribute support on a U-VNE created using the GenericUVNE template.

**Table B-65** Ethernet (IEEE 802.3) Attribute Support on GenericUVNEs

Attribute	Supported
<b>IMO Name—IEthernet</b>	
MAC Address	Y
Duplex Mode	
Output Flow Control	
Input Flow Control	
IANA Type	
Containing CTPs	
Contained CTPs	
Port Type	

### Base Logical Components

Table B-66 lists the base logical attribute support on a U-VNE created using the GenericUVNE template.



Note

Table B-66 includes the supported technologies only.

**Table B-66** Base Logical Components Attribute Support on GenericUVNEs

Attribute	Supported
<b>IMO Name—IManagedElement</b>	
IP Address	Y
Communication State	Y
Investigation State	Y
Element Category	Y
Element Type and Key	Y
Device Name	Y
System Name	Y
System Description	Y
Up Time	Y
Software Version	Y
Vendor Identity	
Memory and CPU Usage	
DRAM Free	
DRAM Used	
Flash Device Size	
NVRAM Size	
Processor DRAM	
Sys Contact	
Sys Location	
Serial Number	
File Systems	
<b>IMO Name—ISystemService</b>	
Type	
Status	
Up Time	

**Common**

[Table B-67](#) lists the common attribute support on a U-VNE created using the GenericUVNE template.

**Note**

[Table B-67](#) includes the supported technologies only.

**Table B-67** Common Attribute Support on GenericUVNEs

Attribute	Supported
<b>IMO Name—IPhysicalLayer</b>	
Media Type	
Clocking Source	
Maximum Speed	Y
Is Internal Port	
Discarded Bandwidth	
Dropped Bandwidth	
Input Bandwidth	
Output Bandwidth	
Discarded and Received Input Data Counters	Y
Dropped and Forward Output Data Counters	Y
Administrative Status	Y
Operational Status	Y
Last Changed	Y
IANA Type	
Containing CTPs	
Contained CTPs	
Port Alias	
Location	
Sending Alarms	
Connector Description	
Part ID	
Connector Serial Num	
Product	
Status	
Managed	



*Table B-67 Common Attribute Support on GenericUVNEs*

Attribute	Supported
<b>IMO Name—IBridgeEntry</b>	
Destination MAC	Y
Outgoing Interface	Y

### GenericUVNE—Supported Service Events

Table B-68 lists the supported service events on a U-VNE created using the GenericUVNE template.

*Table B-68 Supported Service Events for GenericUVNE*

Event Name	Supported	Expedited
Device Unreachable	Y	N
Discard Packets	Y	N
Dropped Packets	Y	N
Port Flapping	Y	N
Port Down	Y	N

## Module Templates

Module templates define a set of port layers—from the connector at Layer 0 to encapsulation at Layer 2—that are applicable to a module. These templates ensure that each port is modeled with the correct port layer information based on the ifType obtained from the SNMP MIB output.

Module templates are applicable to standard modules only (not pluggable modules). You do not need to use a module template to add a pluggable module.

When adding support for a new module using the VCB, you must identify the module template that matches the capabilities of the module.

For example, the atm-default module template contains the following port layer definitions, which are typical port layers for an OC3 ATM card:

- Layer 0—Fiber optic
- Layer 1—OC3
- Layer 2—ATM

These definitions make this template suitable for modules with ports that:

- Use fiber optic cable.
- Support the OC-3 data transfer rates over SONET.
- Use ATM encapsulation for transporting IP traffic between two peers.




---

**Note** You cannot add modules to a U-VNE created using the GenericUVNE template.

---

Module templates are collected into groups, such as the `ciscophysicalspec2` group for Cisco modules. The information contained in the module specification files is summarized in the [Module Groups and Module Specification Files, page B-66](#).




---

**Note** Module definitions that you create with the VCB are added to the module group that contains the template on which the definition is based.

---

After you obtain the module identifier and research the capabilities of the module, use [Module Templates by Technology, page B-69](#) to identify the module template that best matches the module. You can then add support for the module using the VCB.

This section contains the following topics:

- [Module Groups and Module Specification Files, page B-66](#)
- [Module Templates by Technology, page B-69](#)

## Module Groups and Module Specification Files




---

**Note** Unlike the modeling that Prime Network does for standard modules, Prime Network models only the ports for pluggable modules. The only module group for pluggable modules is the `pluggable-ports-spec` file. The remainder of this section applies to standard modules only (not pluggable modules).

---

A module group is the name of a vendor-specific module specification file that is stored in the Prime Network registry. A module specification file is an XML file that lists supported modules and other properties, such as port layers and `sysOID`. When you use **vcb module** commands to add, modify, or delete a module:

- You provide the name of a module specification file as an argument to the **-group** option. (For more information, see [VCB CLI Command Reference: Standard Modules, page B-18](#).)
- The VCB modifies the module specification file: adding, updating, or deleting the module definition.




---

**Note** The VCB allows you to update and delete only those modules that you added using the VCB.

---

Prime Network enables you to extend the following module specification files:

- `ciscophysicalspec2`
- `ciscocatalyst3400spec`
- `cisco-catalyst-spec`

[Table B-69](#) summarizes the technologies that are supported and the module templates that are provided in the module specification files. For more information about a module template, use the link in the Technologies column.

Table B-69 Module Group Summary for Standard Modules

Module Group	Technologies	Template Names
ciscophysicalspec2	<a href="#">Ethernet (Fixed), page B-69</a>	<ul style="list-style-type: none"> <li>• A10GigaEthernet</li> <li>• ethernet-default-over-optic</li> <li>• ethernetDefault</li> <li>• gigaEthernet</li> <li>• EthernetChannelSwitchDefault</li> </ul>
	<a href="#">Ethernet (Multiloader), page B-70</a>	<ul style="list-style-type: none"> <li>• 10Gigaethernet-</li> <li>• Gigaethernet-Fiber</li> <li>• GE-fiberoptic-ethernet-default</li> <li>• ethernetDefault-RJ45-or-Fiber2</li> <li>• E1orGigabitTechnology2</li> <li>• GE-over-OC12-pos-default</li> <li>• GE-over-OC3-pos-default3</li> <li>• ethernet-or-oc-pos-default3</li> <li>• ethernet-or-OC12-</li> <li>• pos-default3</li> <li>• ethernet-or-oc48-</li> <li>• pos-default</li> <li>• DWDM10GigaEthernet</li> </ul>
	<a href="#">POS (Fixed), page B-72</a>	<ul style="list-style-type: none"> <li>• PPPdefaultOC48</li> <li>• PPPdefault</li> </ul>
	<a href="#">POS (Multiloader), page B-72</a>	<ul style="list-style-type: none"> <li>• POS-OC3-default</li> <li>• PPPdefaultOC12</li> <li>• PPPdefaultOC192</li> <li>• PPPdefaultOC3</li> <li>• PPPdefaultOC768</li> <li>• DWDMOC768</li> </ul>

Table B-69 Module Group Summary for Standard Modules (continued)

Module Group	Technologies	Template Names
ciscophysicalspec2 (continued)	<a href="#">Channelized T1/E1 (Fixed), page B-73</a>	<ul style="list-style-type: none"> <li>RJ45-T1E1Channelized</li> <li>T1E1Channelized</li> <li>T1E1Channelized-ATMorCEM</li> <li>E3Default</li> <li>E3Loader</li> <li>E1Channelized</li> <li>E1Default</li> </ul>
	<a href="#">Channelized OCXX (Fixed), page B-74</a>	<ul style="list-style-type: none"> <li>ChannelizedOC3</li> <li>ChannelizedOC12</li> <li>ChannelizedOC12xx</li> <li>ChannelizedOCxx</li> </ul>
	<a href="#">ATM (Fixed), page B-74</a>	<ul style="list-style-type: none"> <li>atmDefault</li> <li>atmOverOC12</li> <li>atm-over-e3ds3</li> <li>T1E1_ATM-IMA</li> <li>ds1Default</li> <li>ds3Default</li> <li>adslDefault</li> </ul>
	<a href="#">ATM (Multiloader), page B-75</a>	<ul style="list-style-type: none"> <li>T3Channelized</li> <li>T3Loader</li> <li>layer2-over-ds1</li> <li>layer2-over-ds3</li> <li>layer2-over-ds3-over-bnc</li> <li>pppOverDS3Default</li> <li>layer2-over-e1</li> <li>HSSIDefault</li> </ul>
	<a href="#">Multitechnology, page B-76</a>	<ul style="list-style-type: none"> <li>36xxMultiTechnologiesModuleDefault</li> <li>8xxMultiTechnologiesModuleDefault</li> <li>MultiTechnologiesModuleLayers</li> <li>MultiTechnologiesModuleDefault2</li> </ul>
	<a href="#">Serial, page B-77</a>	<ul style="list-style-type: none"> <li>PPPwithRJ11</li> <li>multichannelDefault</li> <li>serialPPPDDefault</li> </ul>
	<a href="#">ISDN, page B-77</a>	<ul style="list-style-type: none"> <li>ds1T1Default</li> <li>BRIDefault</li> </ul>

Table B-69 Module Group Summary for Standard Modules (continued)

Module Group	Technologies	Template Names
ciscophysicalspec2 (continued)	<a href="#">Generic</a> , page B-78	<ul style="list-style-type: none"> <li>TSLineDefault</li> <li>generic-port1</li> <li>voiceEMDefault</li> </ul>
ciscocatalyst3400spec	<a href="#">Ethernet (Cisco Catalyst 3400)</a> , page B-79	<ul style="list-style-type: none"> <li>cisco-3400-MultiTechnologiesModuleDefault</li> <li>cisco-3400-ethernetDefault-</li> <li>RJ45-or-Fiber</li> </ul>
cisco-catalyst-spec	<a href="#">Ethernet (Cisco Catalyst)</a> , page B-79	<ul style="list-style-type: none"> <li>EthernetDefault</li> <li>FastEthernetDefault</li> <li>GigaEthernetDefault</li> <li>GigaEthernetOnCopper</li> <li>giga-ethernet</li> </ul>

## Module Templates by Technology

This section presents module templates organized by technology:

- [Ethernet \(Fixed\)](#), page B-69
- [Ethernet \(Multiloader\)](#), page B-70
- [POS \(Fixed\)](#), page B-72
- [POS \(Multiloader\)](#), page B-72
- [Channelized T1/E1 \(Fixed\)](#), page B-73
- [Channelized OCXX \(Fixed\)](#), page B-74
- [ATM \(Fixed\)](#), page B-74
- [ATM \(Multiloader\)](#), page B-75
- [Multitechnology](#), page B-76
- [Serial](#), page B-77
- [ISDN](#), page B-77
- [Generic](#), page B-78
- [Ethernet \(Cisco Catalyst 3400\)](#), page B-79
- [Ethernet \(Cisco Catalyst\)](#), page B-79

### Ethernet (Fixed)

[Table B-70](#) lists module templates that support EthernetCSMA/CD at Layer 1 and a single connector type at Layer 0. When there is more than one Layer 2 option, Layer 2 is modeled based on transmission rate.

**Module Group**

These templates are defined in the ciscophysicalspec2 module group.

**Table B-70** *Module Templates—Ethernet (Fixed)*

Template Name	Layer 0	Layer 1	Layer 2	Example Modules
A10GigaEthernet	Fiber optic	EthernetCSMA/CD	10 Gigabit Ethernet	<ul style="list-style-type: none"> <li>WS-SUP32-10GE-3B</li> <li>7600-ES+2TG</li> </ul>
ethernet-default-over-optic	Fiber optic	EthernetCSMA/CD	<ul style="list-style-type: none"> <li>Ethernet</li> <li>Fast Ethernet</li> <li>Gigabit Ethernet</li> </ul>	<ul style="list-style-type: none"> <li>WS-6700-DFC3B</li> <li>WS-6700-DFC3BXL</li> </ul>
ethernetDefault	RJ45	EthernetCSMA/CD	<ul style="list-style-type: none"> <li>Ethernet</li> <li>Fast Ethernet</li> <li>Gigabit Ethernet</li> </ul>	<ul style="list-style-type: none"> <li>8FE-TX-RJ45</li> <li>SPA-8X1FE-TX-V2</li> </ul>
gigaEthernet	Fiber optic	EthernetCSMA/CD	Gigabit Ethernet	<ul style="list-style-type: none"> <li>WS-X4624-SFP-E</li> <li>7600-ES+3C</li> </ul>
EthernetChannelSwitchDefault	RJ45	EthernetCSMA/CD	EtherChannel	NM-16ESW

**Ethernet (Multiloader)**

[Table B-71](#) lists Ethernet module templates that support multiple options at more than one port layer.

**Module Group**

These templates are defined in the ciscophysicalspec2 module group.

**Note**

In addition to Ethernet, some module templates in [Table B-71](#) also support POS or DWDM ports. (See the footnotes for [Table B-71](#).)

**Table B-71** *Module Templates—Ethernet (Multiloader)*

Template Name	Layer 0	Layer 1	Layer 2	Example Modules
10Gigaethernet-Gigaethernet-Fiber <sup>1</sup>	<ul style="list-style-type: none"> <li>Fiber optic</li> <li>RJ45</li> </ul>	EthernetCSMA/CD	<ul style="list-style-type: none"> <li>10 Gigabit Ethernet</li> <li>Gigabit Ethernet</li> </ul>	<ul style="list-style-type: none"> <li>76-ES+XC-20G3C</li> <li>WS-X45-SUP6-E</li> <li>WS-X4606-X2-E</li> </ul>
GE-fiberoptic-ethernet-default <sup>2</sup>	RJ45	EthernetCSMA/CD	<ul style="list-style-type: none"> <li>Ethernet</li> <li>Fast Ethernet</li> </ul>	catalyst375024ME (cevModuleCat375024M)
	Fiber optic		<ul style="list-style-type: none"> <li>Gigabit Ethernet</li> </ul>	
ethernetDefault-RJ45-or-Fiber <sup>2</sup>	<ul style="list-style-type: none"> <li>Fiber optic</li> <li>RJ45</li> </ul>	EthernetCSMA/CD	<ul style="list-style-type: none"> <li>Ethernet</li> <li>Fast Ethernet</li> <li>Gigabit Ethernet</li> </ul>	<ul style="list-style-type: none"> <li>WS-X4232-RJ-XX</li> <li>WS-X4524-GB-RJ45V</li> </ul>

Table B-71 Module Templates—Ethernet (Multiloader) (continued)

Template Name	Layer 0	Layer 1	Layer 2	Example Modules
E1orGigabitTechnology <sup>2</sup>	<ul style="list-style-type: none"> <li>RJ45</li> <li>Fiber optic</li> </ul>	EthernetCSMA/CD	<ul style="list-style-type: none"> <li>Ethernet</li> <li>Fast Ethernet</li> </ul>	Motherboard for 2941 (cevCpu2941)
	<ul style="list-style-type: none"> <li>RJ48</li> </ul>	DS1/E1	<ul style="list-style-type: none"> <li>Gigabit Ethernet</li> </ul>	
GE-over-OC12-pos-default <sup>3</sup>	Fiber optic	OC12	<ul style="list-style-type: none"> <li>PPP</li> <li>HDLC</li> </ul>	OSM-4OC12-POS-SI+
		EthernetCSMA/CD	<ul style="list-style-type: none"> <li>Gigabit Ethernet</li> </ul>	
GE-over-OC3-pos-default <sup>3</sup>	Fiber optic	OC3	<ul style="list-style-type: none"> <li>PPP</li> <li>HDLC</li> </ul>	OSM-4OC3-POS-SI+
	RJ45	EthernetCSMA/CD	<ul style="list-style-type: none"> <li>Gigabit Ethernet</li> </ul>	
ethernet-or-oc-pos-default <sup>3</sup>	Fiber optic	OC3	<ul style="list-style-type: none"> <li>PPP</li> <li>HDLC</li> </ul>	OSM-2+4GE-WAN+
		EthernetCSMA/CD	<ul style="list-style-type: none"> <li>Fast Ethernet</li> <li>Gigabit Ethernet</li> <li>10 Gigabit Ethernet</li> </ul>	
ethernet-or-OC12-pos-default <sup>3</sup>	Fiber optic	OC12	<ul style="list-style-type: none"> <li>PPP</li> <li>HDLC</li> </ul>	<ul style="list-style-type: none"> <li>OSM-2OC12-POS-SI</li> <li>OSM-2OC12-POS-SI+</li> </ul>
		EthernetCSMA/CD	<ul style="list-style-type: none"> <li>Fast Ethernet</li> <li>Gigabit Ethernet</li> <li>10 Gigabit Ethernet</li> </ul>	
ethernet-or-oc48-pos-default	Fiber optic	OC48	<ul style="list-style-type: none"> <li>PPP</li> <li>HDLC</li> </ul>	OSM-1OC48-POS-SI+
		EthernetCSMA/CD	<ul style="list-style-type: none"> <li>Fast Ethernet</li> <li>Gigabit Ethernet</li> <li>10 Gigabit Ethernet</li> </ul>	
DWDMA10GigaEthernet <sup>4</sup>	Fiber optic	<ul style="list-style-type: none"> <li>DWDM</li> <li>EthernetCSMA/CD</li> </ul>	10 Gigabit Ethernet	<ul style="list-style-type: none"> <li>76-ES+XT-2TG3CXL</li> <li>76-ES+XT-4TG3C</li> </ul>

1. The connector type is modeled as RJ45 only for a Gigabit Ethernet port that is not pluggable. The connector type for other Gigabit Ethernet and 10 Gigabit Ethernet ports is modeled as fiber optic.
2. The connector type is modeled based on transmission rate.
3. This module template supports either Ethernet or POS ports.
4. This module template supports either Ethernet or DWDM ports.

**POS (Fixed)**

Table B-72 lists POS module templates that support fixed Layer 0 (fiber optic) and Layer 1 (OC3, OC12, or OC48) options.

**Module Group**

These templates are defined in the ciscophysicalspec2 module group.

*Table B-72 Module Templates—POS (Fixed)*

Template Name	Maximum Transmission Rate (Mbps) Supported	Layer 0	Layer 1	Layer 2	Example Modules
PPPdefaultOC48	2488.32	Fiber optic	OC48	<ul style="list-style-type: none"> <li>• PPP</li> <li>• HDLC</li> <li>• Frame relay</li> </ul>	<ul style="list-style-type: none"> <li>• SFP-OC48-IR1</li> <li>• gsr-e-qoc48-sm-lr-sc</li> <li>• 16OC48-POS/DPT</li> </ul>
PPPdefault	155.52	Fiber optic	OC3	<ul style="list-style-type: none"> <li>• PPP</li> <li>• HDLC</li> <li>• Frame relay</li> </ul>	<ul style="list-style-type: none"> <li>• GSR-SFC6</li> <li>• GSR-CSC</li> </ul>

**POS (Multiloader)**

Table B-73 lists templates for POS modules with multiple Layer 2 options.

**Module Group**

These templates are defined in the ciscophysicalspec2 module group.

*Table B-73 Module Templates—POS (Multiloader)*

Template Name	Maximum Transmission Rate (Mbps) Supported	Layer 0	Layer 1	Layer 2	Example Modules
POS-OC3-default	155.52	Fiber optic	OC3	<ul style="list-style-type: none"> <li>• PPP</li> <li>• HDLC</li> <li>• Frame Relay</li> </ul>	—
PPPdefaultOC12	622.08	Fiber optic	OC12	<ul style="list-style-type: none"> <li>• PPP</li> <li>• HDLC</li> <li>• Frame Relay</li> </ul>	SPA-8XOC12-POS
PPPdefaultOC192	9,953.28	Fiber optic	OC192	<ul style="list-style-type: none"> <li>• PPP</li> <li>• HDLC</li> </ul>	SPA-OC192POS-LR



Table B-73 Module Templates—POS (Multiloader) (continued)

Template Name	Maximum Transmission Rate (Mbps) Supported	Layer 0	Layer 1	Layer 2	Example Modules
PPPdefaultOC3	155.52	Fiber optic	OC3	<ul style="list-style-type: none"> <li>• PPP</li> <li>• HDLC</li> <li>• Frame Relay</li> </ul>	<ul style="list-style-type: none"> <li>• SFP-OC3-SR</li> <li>• SFP-OC3-IR1</li> </ul>
PPPdefaultOC768	39,813.12	Fiber optic	OC768	<ul style="list-style-type: none"> <li>• PPP</li> <li>• HDLC</li> <li>• Frame Relay</li> </ul>	1OC768-ITU/C
DWDMOC768	39,813.12	Fiber optic	OC768	<ul style="list-style-type: none"> <li>• PPP</li> <li>• HDLC</li> </ul>	—

**Channelized T1/E1 (Fixed)**

Table B-74 lists templates for channelized T1/E1 modules where port layers are fixed.

**Module Group**

These templates are defined in the ciscophysicalspec2 module group.

Table B-74 Module Templates—Channelized T1/E1 (Fixed)

Template Name	Maximum Transmission Rate (Mbps) Supported	Layer 0	Layer 1	Layer 2	Example Modules
RJ45-T1E1Channelized	1.544	RJ45	T1E1	—	SPA-8XCHT1/E1
T1E1Channelized	1.544	RJ48	T1E1	—	<ul style="list-style-type: none"> <li>• NM-2CE1T1-PRI</li> <li>• PA-MC-4T1</li> </ul>
T1E1Channelized-ATMorCEM <sup>1</sup>	1.544	RJ48	T1E1	—	HWIC-4T1/E1
E3Default	44.736	BNC	DS3	—	ESR-8E3/DS3
E3Loader	44.736	BNC	DS3	—	—
E1Channelized	1.544	RJ48	E1	—	PA-8CE1
E1Default	1.544	RJ45	E1T1	—	VWIC2-1MFT-T1E1

1. Layer 2—ATM or CEM—is built at runtime.

**Channelized OCXX (Fixed)**

Table B-75 lists templates for Optical Carrier modules where port layers are fixed.

**Module Group**

These templates are defined in the ciscophysicalspec2 module group.

**Table B-75** Module Templates—Channelized OCXX (Fixed)

Template Name	Maximum Transmission Rate (Mbps) Supported	Layer 0	Layer 1	Layer 2	Example Modules
ChannelizedOC3	155.52	Fiber optic	OC3	—	—
ChannelizedOC12	622.08	Fiber optic	OC12	—	—
ChannelizedOC12xx	622.08	Fiber optic	OC12	—	SPA-1XCHOC12/DS0
ChannelizedOCxx	155.52	Fiber optic	OC3	—	<ul style="list-style-type: none"> <li>SPA-1XCHSTM1/OC3</li> <li>SPA-CHOC3-CE-ATM</li> </ul>

**ATM (Fixed)**

Table B-76 lists templates for ATM modules where port layers are fixed.

**Module Group**

These templates are defined in the ciscophysicalspec2 module group.

**Table B-76** Module Templates—ATM (Fixed)

Template Name	Maximum Transmission Rate (Mbps) Supported	Layer 0	Layer 1	Layer 2	Example Modules
atmDefault	155.52	Fiber optic	OC3	ATM	GSR-SFC12410
atmOverOC12	—	Fiber optic	OC12	ATM	<ul style="list-style-type: none"> <li>SPA-1XOC12-ATM-V2</li> <li>SPA-1XOC12-ATM</li> </ul>
atm-over-e3ds3	—	RJ48	DS3	ATM	—
T1E1_ATM-IMA	1.544	RJ48	T1E1	ATM	—
ds1Default	—	BNC	DS1	ATM	—
ds3Default	—	BNC	DS3	ATM	PA-A3-T3
adslDefault	—	RJ11	ADSL	ATM	<ul style="list-style-type: none"> <li>WIC-1SHDSL</li> <li>WIC-1ADSL</li> <li>WIC-1ADSL-DG</li> </ul>

**ATM (Multiloader)**

Table B-77 lists templates for ATM modules with multiple Layer 2 options.

**Module Group**

These templates are defined in the ciscophysicalspec2 module group.

**Table B-77** Module Templates—ATM (Multiloader)

Template Name	Maximum Transmission Rate (Mbps) Supported	Layer 0	Layer 1	Layer 2	Example Modules
T3Channelized <sup>1</sup>	—	BNC	DS3	<ul style="list-style-type: none"> <li>• PPP</li> <li>• HDLC</li> <li>• Frame relay</li> <li>• ATM</li> </ul>	<ul style="list-style-type: none"> <li>• SPA-2XCT3/DS0</li> <li>• PA-2T3/E3-EC</li> </ul>
T3Loader	—	BNC	DS3	<ul style="list-style-type: none"> <li>• PPP</li> <li>• HDLC</li> <li>• Frame relay</li> </ul>	—
layer2-over-ds1	1.544	RJ48	DS1	<ul style="list-style-type: none"> <li>• PPP</li> <li>• HDLC</li> <li>• Frame relay</li> <li>• ATM</li> </ul>	VWIC-2MFT-T1-DIR
layer2-over-ds3	44.736	RJ48	DS3	<ul style="list-style-type: none"> <li>• PPP</li> <li>• HDLC</li> <li>• Frame relay</li> <li>• ATM</li> </ul>	<ul style="list-style-type: none"> <li>• NM-1A-E3</li> <li>• NM-1T3/E3</li> </ul>
layer2-over-ds3-over-bnc	44.736	BNC	DS3	<ul style="list-style-type: none"> <li>• PPP</li> <li>• HDLC</li> <li>• Frame relay</li> <li>• ATM</li> </ul>	—
pppOverDS3Default	44.736	BNC	DS3	<ul style="list-style-type: none"> <li>• PPP</li> <li>• HDLC</li> </ul>	<ul style="list-style-type: none"> <li>• copper-6ds3</li> <li>• copper-12ds3</li> <li>• 2DS3-SMB</li> <li>• NM-4T</li> </ul>

Table B-77 Module Templates—ATM (Multiloader) (continued)

Template Name	Maximum Transmission Rate (Mbps) Supported	Layer 0	Layer 1	Layer 2	Example Modules
layer2-over-e1	1.544	RJ48	E1	<ul style="list-style-type: none"> <li>• PPP</li> <li>• HDLC</li> <li>• Frame relay</li> <li>• ATM</li> </ul>	NM-1CE1T1-PRI
HSSIDefault	—	DB50	HSSI	<ul style="list-style-type: none"> <li>• PPP</li> <li>• HDLC</li> <li>• Frame relay</li> </ul>	—

1. This module template supports both full and channelized T3 and ATM over T3.

## Multitechnology

Table B-78 lists multitechnology templates, including the following, which support modules where the connector type is not determined:

- MultiTechnologiesModuleLayers
- MultiTechnologiesModuleDefault



**Note** Do not use these templates unless no other template matches the modules to be added.

### Module Group

These templates are defined in the ciscophysicalspec2 module group.

Table B-78 Module Templates—Multitechnology

Template Name	Layer 0	Layer 1	Layer 2	Example Modules
36xxMultiTechnologiesModuleDefault <sup>1</sup>	RJ45	• EthernetCSMA/CD	<ul style="list-style-type: none"> <li>• Fast Ethernet</li> <li>• Gigabit Ethernet</li> </ul>	—
		• Serial	<ul style="list-style-type: none"> <li>• PPP</li> <li>• HDLC</li> <li>• Frame relay</li> </ul>	
8xxMultiTechnologiesModuleDefault	RJ45	• EthernetCSMA/CD	<ul style="list-style-type: none"> <li>• Fast Ethernet</li> <li>• Gigabit Ethernet</li> </ul>	—
		• Serial	<ul style="list-style-type: none"> <li>• PPP</li> <li>• HDLC</li> <li>• Frame relay</li> </ul>	

Table B-78 Module Templates—Multitechnology (continued)

Template Name	Layer 0	Layer 1	Layer 2	Example Modules
MultiTechnologiesModule Layers <sup>2</sup>	<ul style="list-style-type: none"> <li>• RJ11</li> <li>• RJ45</li> <li>• RJ48</li> <li>• Fiber optic</li> <li>• DB60</li> </ul>	<ul style="list-style-type: none"> <li>• EthernetCSMA/CD</li> <li>• DS1</li> <li>• E1</li> <li>• OC3</li> <li>• ADSL</li> <li>• Serial</li> </ul>	<ul style="list-style-type: none"> <li>• Fast Ethernet</li> <li>• Gigabit Ethernet</li> <li>• PPP</li> <li>• HDLC</li> <li>• Frame relay</li> <li>• ATM</li> </ul>	OSM-1CHOC12/T3-SI
MultiTechnologiesModule Default <sup>2</sup>	<ul style="list-style-type: none"> <li>• RJ11</li> <li>• RJ45</li> <li>• RJ48</li> <li>• Fiber optic</li> <li>• DB60</li> </ul>	<ul style="list-style-type: none"> <li>• EthernetCSMA/CD</li> <li>• DS1</li> <li>• E1</li> <li>• OC3</li> <li>• ADSL</li> <li>• Serial</li> </ul>	<ul style="list-style-type: none"> <li>• Fast Ethernet</li> <li>• Gigabit Ethernet</li> <li>• PPP</li> <li>• HDLC</li> <li>• Frame relay</li> <li>• ATM</li> </ul>	gsr-sfc16-oc192

1. This module template primarily supports Cisco 3600 series modules.
2. Use this template only when no other template matches the modules to be supported.

## Serial

Table B-79 lists templates that provide support for modules with serial interfaces when the information for Layer 1 is not clear.

### Module Group

These templates are defined in the ciscophysicalspec2 module group.

Table B-79 Module Templates—Serial

Template Name	Layer 0	Layer 1	Layer 2	Example Modules
PPPwithRJ11	RJ11	Serial	PPP	WIC-1AM-V2
multichannelDefault	RJ48	Serial	PPP	—
serialPPPDefault	<ul style="list-style-type: none"> <li>• RJ45</li> <li>• RJ48</li> <li>• Fiber optic</li> <li>• DB60</li> <li>• Generic connector</li> </ul>	Serial	<ul style="list-style-type: none"> <li>• PPP</li> <li>• HDLC</li> <li>• Frame relay</li> </ul>	<ul style="list-style-type: none"> <li>• HWIC-4T</li> <li>• NM-2W</li> <li>• WIC-SERIAL-1T</li> </ul>

## ISDN

Table B-80 lists templates for ISDN modules.

### Module Group

These templates are defined in the ciscophysicalspec2 module group.

Table B-80 Module Templates—ISDN

Template Name	Maximum Transmission Rate (kbps) Supported	Layer 0 <sup>1</sup>	Layer 1	Layer 2	Example Modules
ds1T1Default	—	<ul style="list-style-type: none"> <li>• RJ45</li> <li>• RJ48</li> <li>• DB60</li> <li>• Fiber optic</li> </ul>	<ul style="list-style-type: none"> <li>• DS1</li> <li>• E1</li> <li>• EthernetCSMA/CD</li> </ul>	ISDN	—
BRIDefault	64	RJ45	ISDN layer 1	ISDN layer 2	WIC-1B-U-V2

1. The connector type is modeled based on the type of port.

## Generic

Table B-81 lists generic templates. Use them to configure modules for technologies that Prime Network does not support.

### Module Group

These templates are defined in the ciscophysicalspec2 module group.

Table B-81 Module Templates—Generic

Template Name	Layer 0	Layer 1	Layer 2	Example Modules
TSLineDefault <sup>1</sup>	Generic connector	Generic Layer 1	—	<ul style="list-style-type: none"> <li>• NM-16A</li> <li>• NM-32A</li> </ul>
generic-port <sup>1</sup>	Generic connector	Generic Layer 1	Generic Layer 2	—
voiceEMDefault <sup>2</sup>	RJ45	Generic Layer 1	Generic Layer 2	<ul style="list-style-type: none"> <li>• VIC-EM</li> <li>• VIC-4VP-FXS-DID</li> </ul>

1. This template provides support for modules whose technologies are not currently supported in Prime Network.

2. This template provides support for voice modules.

## Ethernet (Cisco Catalyst 3400)

Table B-82 lists templates that support modules for the Cisco Catalyst 3400 device group.

### Module Group

These templates are defined in the ciscocatalyst3400spec module group.

**Table B-82** Module Templates—Ethernet (Cisco Catalyst 3400)

Template Name	Layer 0	Layer 1	Layer 2	Example Modules
cisco-3400-MultiTechnologiesModuleDefault	RJ45	EthernetCSMA/CD	Ethernet	3400 fixed modules
<b>Note</b> Not recommended for 10/100/1000 Ethernet ports. Instead, use cisco-3400-ethernetDefault-RJ45-or-Fiber.				
cisco-3400-ethernetDefault-RJ45-or-Fiber	<ul style="list-style-type: none"> <li>• RJ45</li> <li>• Fiber optic</li> </ul>	EthernetCSMA/CD	<ul style="list-style-type: none"> <li>• Ethernet</li> <li>• FastEthernet</li> <li>• GigaEthernet</li> </ul>	3400 fixed modules

## Ethernet (Cisco Catalyst)



**Note** For modules in the Cisco Catalyst 3400 device group, see [Ethernet \(Cisco Catalyst 3400\)](#).

Table B-83 lists templates that support modules for Cisco Catalyst devices.

### Module Group

These templates are defined in the cisco-catalyst-spec module group.

**Table B-83** Module Templates—Ethernet (Cisco Catalyst)

Template Name	Layer 0	Layer 1	Layer 2	Example Modules
EthernetDefault	RJ45	EthernetCSMA/CD	Ethernet	<ul style="list-style-type: none"> <li>• WS-C3560G-24TS</li> <li>• WS-SUP720-3BXL</li> </ul>
FastEthernetDefault	RJ45	EthernetCSMA/CD	FastEthernet	<ul style="list-style-type: none"> <li>• WS-X4148-RJ</li> <li>• ws-c2924-x1-v</li> </ul>
GigaEthernetDefault	RJ45	EthernetCSMA/CD	GigaEthernet	<ul style="list-style-type: none"> <li>• WS-F6K-MSFC2A</li> <li>• OSM-2+4GE-WAN+</li> </ul>
GigaEthernetOnCopper	UTP	EthernetCSMS/CD	GigaEthernet	Catalyst 6500 Supervisor Module 720 base board
giga-ethernet	Fiber optic	EthernetCSMA/CD	GigaEthernet	<ul style="list-style-type: none"> <li>• cat6k-wsx-6066-slb-apc</li> <li>• wsx6ksup1a2ge</li> <li>• wsx6ksup22ge</li> </ul>

## Event Templates

Event templates work together to extract information from a syslog or a trap and to generate the keys and the location ID for associating a Prime Network event with a managed element device component. For more information, see the following sections:

- [Terminology Used in Event Templates](#), page B-80
- [Supported Interface Types](#), page B-80
- [Event Templates Functional Summary](#), page B-81
- [Event Templates Input Summary—Required and Optional Input](#), page B-83

For information about specific template types, see the following sections:

- [Event Identification Templates](#), page B-84
- [Unique ID Templates—for Traps Only](#), page B-86
- [Event Subtype Templates](#), page B-87
- [Entity ID Templates—for Traps Only](#), page B-89
- [Entity Key Templates](#), page B-90
- [Prime Network Event Templates](#), page B-92

## Terminology Used in Event Templates

The following terms are used in the tables in this section.

### Entity ID

The entity ID identifies the entity in the VNE with which to associate the event. For example, for interface-based events, `ifIndex` or `ifName` can be used as the entity ID.

### Unique ID

The unique ID is used to create a unique location for the event when association to the exact entity is not possible. For example, when associating BGP traps to the Managed Element, the neighbor IP address can be used as the unique ID.

## Supported Interface Types

The VCB can automatically identify the following interface types and associates events to them:

- Ether Channel
- GRE Tunnel
- DSO Bundle
- MPLSTunnel
- IMA Group
- MLPPP
- CEM Group
- IpInterface (Loopback, Vlan, all other subinterfaces)

For other interface types, the VCB associates the event to the layer 1 device component.



## Event Templates Functional Summary

Event templates extract information and generate keys to associate an event with the correct VNE or U-VNE component. [Table B-84](#) lists event templates (by type) and explains what each template does.

**Table B-84** Event Template Functions

Template	Performs This Function for a...	
	Syslog	Trap
<b>Event Identification Templates—Mandatory. Use one.</b>		
snmp-trap-identification	—	Extracts information that identifies an event
syslog-identification	Extracts: <ul style="list-style-type: none"> <li>Information that identifies an event</li> <li>Event subtypes</li> <li>Entity ID</li> <li>Unique ID</li> </ul>	—
<b>Event Subtype Templates—Optional. Use one if there are subtypes for the event.</b>		
snmp-trap-subtype-from-oid	—	<ul style="list-style-type: none"> <li>Extracts subtype information from the event</li> <li>Maps the subtype values to the event subtypes defined in Prime Network</li> </ul>
snmp-trap-subtype-from-trapoid		
snmp-trap-subtype-from-value		
syslog-subtype-from-expression	Maps the subtype values—extracted by the syslog-identification template—to the event subtypes defined in Prime Network	—
<b>Unique ID Templates—Optional. Use one when you cannot associate the event to an exact entity.</b>		
snmp-trap-identifier-from-oid	—	Extracts the <a href="#">unique ID</a>
snmp-trap-identifier-from-value		
<b>Entity ID Templates—Optional. Use one when you can associate the event to an exact entity.</b>		
snmp-trap-entity-from-oid	—	Extracts the <a href="#">entity ID</a>
snmp-trap-entity-from-value		

Table B-84 Event Template Functions (continued)

Template	Performs This Function for a...	
	Syslog	Trap
<b>Entity Key Templates—Mandatory. Use one.</b>		
create-managedelement-key	<ul style="list-style-type: none"> <li>Creates the device component key</li> <li>Associates the event with the Managed Element device component</li> </ul>	
create-interface-key-from-ifindex	—	<p>Creates the interface device component key from the ifIndex, using the entity ID that was extracted by an entity ID template (snmp-trap-entity-from-oid or snmp-trap-entity-from-value)</p> <p>Associates the event with the appropriate interface layer; see <a href="#">Supported Interface Types, page B-80</a></p>
create-interface-key-from-ifname	<ul style="list-style-type: none"> <li>Creates the interface device component key from the ifname, using the entity ID that was extracted by: <ul style="list-style-type: none"> <li>syslog-identification template for a syslog</li> <li>an entity ID template (snmp-trap-entity-from-oid or snmp-trap-entity-from-value) for a trap</li> </ul> </li> <li>Associates the event with the appropriate interface layer; see <a href="#">Supported Interface Types, page B-80</a></li> </ul> <p><b>Note</b> This template is more frequently used with syslogs than with traps.</p>	
<b>Prime Network Event Templates—Mandatory. Use one.</b>		
create-ana-trap-event	—	<p>Creates both of the following:</p> <ul style="list-style-type: none"> <li>A unique location for the event, based on the <a href="#">entity ID</a> (device component key) or the <a href="#">unique ID</a> (extracted by an entity ID or a unique ID template)</li> <li>Prime Network event, using subtypes extracted by an event subtype template: <ul style="list-style-type: none"> <li>snmp-trap-subtype-from-oid</li> <li>snmp-trap-subtype-from-trapoid</li> <li>snmp-trap-subtype-from-value</li> </ul> </li> </ul>
create-ana-syslog-event	<p>Creates both of the following:</p> <ul style="list-style-type: none"> <li>A unique location for the event, based on <a href="#">entity ID</a> (device component key) or <a href="#">unique ID</a> (extracted by the syslog-identification template).</li> <li>Prime Network event, using subtypes extracted by the syslog-identification template</li> </ul>	—

For the input required for each template, see [Event Templates Input Summary—Required and Optional Input](#), page B-83.

## Event Templates Input Summary—Required and Optional Input

Table B-85 summarizes event templates and the mandatory and optional input arguments for them.

Table B-85 Event Template Variables

Templates	Variable Name	Variable Type	Variable Description
<b>Event Identification Templates</b>			
syslog-identification	expression	Mandatory	Regular expression to match against the incoming syslog message.
	testmessage	Optional	String that is an example of the actual syslog message.
snmp-trap-identification	oid	Mandatory	SNMP trap OID.
<b>Unique ID Templates</b>			
snmp-trap-identifier-from-oid	inOID	Mandatory	Partial string that uniquely identifies the required OID in the varbind list.
	index	Optional	Used to create a unique location ID for the event when association to the exact entity is not possible. Default value is 1.
snmp-trap-identifier-from-value	inOID	Mandatory	Partial string that uniquely identifies the required OID in the varbind list.
<b>Event Subtype Templates</b>			
snmp-trap-subtype-from-oid	inOID	Mandatory	Partial string that uniquely identifies the required OID in the varbind list.
	replacing-rules		Defines mapping between event subtype and value in the trap that indicates the subtype.  The subtype names provided in replacing-rules must match the subtype names used to add the event using the <b>vcb event add</b> command. To view event subtypes, use the <b>vcb event view</b> command.
	index	Optional	Used to create a unique location ID for the event when association to the exact entity is not possible. Default value is 1.
snmp-trap-subtype-from-trapoid	replacing-rules	Mandatory	Defines the mapping between event subtype and value in the trap that indicates the subtype.
snmp-trap-subtype-from-value	inOID	Mandatory	Partial string that uniquely identifies the required OID in the varbind list.
	replacing-rules		Defines the mapping between event subtype and value in the trap that indicates the subtype.
syslog-subtype-from-expression	replacing-rules	Mandatory	Defines the mapping between event subtype and value in the trap that indicates the subtype.

Table B-85 Event Template Variables (continued)

Templates	Variable Name	Variable Type	Variable Description
<b>Entity ID Templates</b>			
snmp-trap-entity-from-oid	inOID	Mandatory	Partial string that uniquely identifies the required OID in the varbind list.
	index	Optional	Used to create a unique location ID for the event when association to the exact entity is not possible. Default value is 1.
snmp-trap-entity-from-value	inOID	Mandatory	Partial string that uniquely identifies the required OID in the varbind list.
<b>Entity Key Templates</b>			
create-interface-key-from-ifindex	—	—	—
create-interface-key-from-ifname	—	—	—
create-managedelement-key	—	—	—
<b>Prime Network Event Templates</b>			
create-ana-trap-event	type	Mandatory	Event name, a string that must match the event name used to create the Prime Network event using the <b>vcb event add</b> command.
	subtype	Mandatory for INFO events only	Event subtype name, a string that must match the subtype name used to create the Prime Network event using the <b>vcb event add</b> command.
create-ana-syslog-event	type	Mandatory	Event name, a string that must match the event name used to create the Prime Network event using the <b>vcb event add</b> command.
	subtype	Mandatory for INFO events only	Event subtype name, a string that must match the subtype name used to create the Prime Network event using the <b>vcb event add</b> command.

## Event Identification Templates

Event identification templates are mandatory. You must use one of these templates:

- [snmp-trap-identification, page B-84](#)
- [syslog-identification, page B-85](#)

### snmp-trap-identification

This template supports SNMP V1, V2, and V3 traps. Any incoming SNMP V1 traps are converted automatically to SNMP V2 and then parsed as SNMP V2 traps. (The first rule in this template is the conversion rule.)

#### Mandatory Input

**oid**—The trap OID. [Table B-86](#) describes how to format the OID for different traps.

Table B-86 Template Input—snmp-trap-identification

If the OID...	Format Input Like This...	Example Input String
Is for a V1 trap	Supply the enterprise OID appended by 0 and then by the specific type.	—
Contains subtype information as these do: <ul style="list-style-type: none"> <li>mplsLdpLibLspUp - 1.3.6.1.4.1.9.10.65.2.0.5</li> <li>mplsLdpLibLspDown - 1.3.6.1.4.1.9.10.65.2.0.6</li> </ul>	Remove subtype information from the input string.	1.3.6.1.4.1.9.10.65.2.0
Is an informational trap as this one is: mplsLdpPathVectorLimitMismatch - 1.3.6.1.4.1.9.10.65.2.0.2	Supply the entire OID.	1.3.6.1.4.1.9.10.65.2.0.2
Does not contain subtype information; however, subtype information is contained in one of the varbinds: demandNbrLayer2Change - 1.3.6.1.4.1.9.9.26.2.0.3 The subtype for this trap is a varbind: isdnLapdOperStatus.	Supply the entire OID.	1.3.6.1.4.1.9.9.26.2.0.3

## syslog-identification

This template supports syslogs. This template contains a single rule that extracts not only event identification information, but also subtype, unique ID, and entity ID when they are available for the event.

### Mandatory Input

**expression**—A regular expression to be matched against the incoming syslog message.

The input string need not be a complete regular expression. It can be a partial syslog message with the parameters in which you are interested marked using keywords. Replace event-specific parameters in the syslog message with the following keywords:

- %%subtypekey%%—Use when the substring represents an event subtype.
- %%uniqueid%%—Use when the substring represents a parameter that uniquely identifies the event.
- %%entityid%%—Use when the substring represents the associated entity of the event.

For example, for the following syslogs:

```
%C6KENV-4-CLOCKFAILED: clock [dec] failed
```

```
%C6KENV-4-CLOCKOK: clock [dec] operational
```

you could provide the following input string:

```
%C6KENV-4-CLOCK%%subtypekey%%: clock %%uniqueid%%
```

There are no `%%entityid%%` parameters in this example, because this syslog must be associated with the ManagedElement device component.

The VCB uses the input string to automatically create a regular expression:  
`“.*%C6KENV-4-CLOCK(\S+):clock (\S+).*”`

There are instance when we require more than one input to uniquely identify the entity in the VNE. VCB allows you to subscript the keywords `%%entityid%%` and `%%uniqueid%%` with integers in order to specify more than one input that constitutes the entityid, for example, `%%entityid1%%`, `%%entityid2%%`.

For example, for EFP syslogs:

`%ETHER_SERVICE-6-UPDOWN: Service instance 111 on interface GigabitEthernet10/0/3 changed to down.`

Here, service instance id and interface name are inputs required to uniquely identify the EFP instance in the VNE. Read the template documentation to determine which variable should be marked as `entityid1` and which should be marked `entityid2`.

### Optional Input

**testmessage**—An input string that supplies an example of the actual syslog message. For example:

`%C6KENV-4-CLOCKFAILED: clock 1 failed`

If supplied, VCB checks this test message against the automatically created regular expression.

### Input Format

`-syslog_identification_expression regular_expression`

`-syslog_identification_testmessage message`

## Unique ID Templates—for Traps Only



### Note

For syslogs, unique ID information is extracted by the event identification template. For more information, see [Event Identification Templates, page B-84](#).

The unique ID differentiates a particular instance of an event from other events of the same type. This parameter is required when:

- It is not possible to associate the event to the appropriate device component in the VNE for some reason, possibly one of the following:
  - The device component is not modeled due to lack of technology support.
  - The corresponding key generation template is not available.
- A unique ID is not required at this time. For example, when Prime Network associates interface traps to a managed element device component, the `ifIndex` or `ifName` is the unique ID. Prime Network automatically appends this unique ID to the location, thereby creating a unique location for each interface despite associating the event with a common entity, the managed element device component.

These templates differ in the way the information is extracted from the trap; select and use only one of the following:

- [snmp-trap-identifier-from-oid, page B-87](#)
- [snmp-trap-identifier-from-value, page B-87](#)

### snmp-trap-identifier-from-oid

The rules in this template extract the unique identifier from one of the OIDs in the varbind list of the trap.

#### Mandatory Input

**inOID**—A partial string that uniquely identifies the required OID in the varbind list.

#### Optional Input

**index**—The location of the required value from the end of the OID. Default value is 1.

#### Input Format

```
-snmp_trap_identifier_from_oid_inOID OID  
-snmp_trap_identifier_from_oid_index indexValue
```

### snmp-trap-identifier-from-value

The rules in this template extract the unique identifier from the value of one of the OIDs in the varbind list.

#### Mandatory Input

**inOID**—A partial string that uniquely identifies the required OID in the varbind list.

#### Input Format

```
-snmp_trap_identifier_from_value_inOID OID
```

## Event Subtype Templates

Use an event subtype template when configuring a trap or a syslog that includes event subtypes. For example, you should use an event subtype template when events arrive as a multistatus set or in asserted and cleared pairs, as is the case with link status traps. Link status traps send Link Down and Link Up traps, two subevents that are related to the same event:

- An asserted event indicates that the link is down.
- A clearing event indicates that the link status has changed to up.

Event subtype templates do the following:

- Extract event subtype information for traps. (For syslogs, event subtype information is extracted by the syslog-identification template; see [syslog-identification](#), page B-85.)
- Map the subtype value to the event subtype name defined in Prime Network.

When subtypes exist for a trap, use one of these event subtype templates:

- [snmp-trap-subtype-from-oid](#), page B-88
- [snmp-trap-subtype-from-trapoid](#), page B-88
- [snmp-trap-subtype-from-value](#), page B-88

When subtypes exist for a syslog, use this template:

- [syslog-subtype-from-expression](#), page B-89.

### snmp-trap-subtype-from-oid

The rules in this template extract event subtype information from one of the OIDs in the varbind list of the trap.

#### Mandatory Input

**inOID**—A partial string that uniquely identifies the required OID in the varbind list.

**replacing-rules**—Replacing rules define the mapping between the event subtype and the value in the trap that indicates the subtype.

The format for a rule is *value-event subtype*. The hyphen between the value and the event subtype is mandatory. Rules must be comma-separated.

**Note**

Supply the same event subtype that was defined for the event with the **vcb event add** command. Use the **vcb event view** command to obtain a list of subtypes.

#### Optional Input

**index**—The location of the required value from the end of the OID. Default value is 1.

#### Input Format

```
-snmp_trap_subtype_from_oid_inOID OID
-snmpt_trap_subtype_from_oid_index indexValue
-snmpt_trap_subtype_from_oid_replacing_rules value-subtype, value-subtype, value-subtype
```

**Note**

The hyphen between value and subtype is required.

### snmp-trap-subtype-from-trapoid

The rules in this template extract event subtype information from one of the OIDs in the varbind list of the trap.

#### Mandatory Input

**replacing-rules**—Replacing rules define the mapping between the event subtype and the value in the trap that indicates the subtype.

The format for a rule is *value-event subtype*. Rules must be comma-separated.

**Note**

Supply the same event subtype that was defined for the event with the **vcb event add** command. Use the **vcb event view** command to obtain a list of subtypes.

#### Input Format

```
-snmp_trap_subtype_from_trapoid_replacing_rules value-subtype, value-subtype, value-subtype
```

### snmp-trap-subtype-from-value

The rules in this template extract the event subtype information from the value of one of the OIDs in the varbind list.



**Mandatory Input**

- **inOID**—A partial string that uniquely identifies the required OID in the varbind list.
- **replacing-rules**—Replacing rules define the mapping between the event subtype and the value in the trap that indicates the subtype.

The format for a rule is *value-event subtype*. Rules must be comma-separated.

**Note**

Supply the same event subtype that was defined for the event with the **vcb event add** command. Use the **vcb event view** command to obtain a list of subtypes.

**Input Format**

```
-snmp_trap_subtype_from_value_inOID inOID
-snmp_trap_subtype_from_value_replacing_rules value-subtype, value-subtype,
value-subtype
```

**syslog-subtype-from-expression**

The rules in this template extract the event subtype information from the value of one of the OIDs in the varbind list.

**Mandatory Input**

**replacing-rules**—Replacing rules define the mapping between the event subtype and the value in the trap that indicates the subtype.

The format for a rule is *value-event subtype*. Rules must be comma-separated.

**Note**

Supply the same event subtype that was defined for the event with the **vcb event add** command. use the **vcb event view** command to obtain a list of subtypes.

**Input Format**

```
-syslog_subtype_from_expression_replacing_rules value-subtype, value-subtype,
value-subtype
```

**Entity ID Templates—for Traps Only****Note**

For syslogs, entity ID information is extracted by an event identification template. For more information, see [Event Identification Templates, page B-84](#).

The entity ID specifies the device component to which the event should be associated in the VNE. These templates differ in the way that the information is extracted from the trap:

- [snmp-trap-entity-from-oid, page B-89](#)
- [snmp-trap-entity-from-value, page B-90](#)

**snmp-trap-entity-from-oid**

The rules in this template extract the entityID from the one of OIDs in the varbind list of the trap.

**Mandatory Input**

**inOID**—A partial string that uniquely identifies the required OID in the varbind list

**Optional Input**

**index**—The location of the required value from the end of the OID. Default value is 1.

**Input Format**

```
-snmp_trap_entity_from_oid_inOID inOID
```

**snmp-trap-entity-from-value**

The rules in this template extract the entity ID from the value of one of the OIDs in the varbind list.

**Mandatory Input**

**inOID**—A partial string that uniquely identifies the required OID in the varbind list

**Input Format**

```
-snmp_trap_entity_from_value_inOID inOID
```

## Entity Key Templates

Selecting a template from this category is mandatory. Entity key templates use the entity ID information—extracted using other templates—to generate a key to uniquely identify the device component in the VNE. The event is later associated with the corresponding device component.

For templates that extract the entity ID, see [syslog-identification, page B-85](#) and [Entity ID Templates—for Traps Only, page B-89](#).

Select one of the following templates:

- [create-interface-key-from-ifindex, page B-90](#)
- [create-interface-key-from-ifname, page B-90](#)
- [create-managedelement-key, page B-91](#)
- [create-efp-key-from-ifname-serviceid, page B-91](#)
- [create-logical-container-key, page B-91](#)
- [create-moduleDC-key-given-entPhysicalIndex, page B-91](#)
- [create-moduleDC-with-slotSubslot-value-key, page B-91](#)
- [create-pw-interface-key-from-tunnelindex, page B-92](#)

**create-interface-key-from-ifindex**

This template creates the interface device component key from the ifIndex and associates the event with the appropriate interface layer.

No input is required.

**create-interface-key-from-ifname**

This template creates the interface device component key from an ifName and associates the event with the appropriate interface layer.

No input is required.

### create-managedelement-key

This template creates the managed element device component key, associating the event with the managed element device component.

No input is required.

### create-efp-key-from-ifname-serviceid

This template creates the EFP DC key from the Service Instance+ifName.



Note

---

The service instance ID should be marked as %%entityid1%% and interface name as %%entityid2%% in the syslog expression given by the user.

---

The event will be associated with particular EFP DC. For example, consider the following:

Syslog Feb 8 11:04:18 MSK: %ETHER\_SERVICE-6-UPDOWN: Service instance 214 on interface TenGigabitEthernet3/3 changed to up

The syslog expression for this should be given as %ETHER\_SERVICE-6-UPDOWN: Service instance %%entityid1%% on interface %%entityid2%% change to %%subtypekey%%.

No input is required.

### create-logical-container-key

This template associates syslog/trap events to the designated containers, as preferred.

Containers list:

CfmService,BfdService,MPBgp,REPSservice,StpService,SbcService,EthernetLMI,ISISSystem,LSE,ClockService. Provide the desired logical container string, from the containers list.

Requires mandatory user input.

### create-moduleDC-key-given-entPhysicalIndex

This template associates trap events to the designated moduleDC, when its observed entPhysicalIndex of Entity-MIB, through mib instrumentation queries from the device; Is given as the input for the desired module entity.

No input is required.

### create-moduleDC-with-slotSubslot-value-key

This template associates syslog events to the corresponding module, knowing the residing slot number. For example, Syslog message "%OIR-6-REMCARD: Card removed from slot 4, interfaces disabled" USER\_INPUT\_MANDATORY <- %OIR-6-REMCARDCARD: Card removed from slot%%entityid%%, interfaces disabled entityid <- 4

No input is required.

### create-pw-interface-key-from-tunnelindex

This template associates trap events to the designated pseudowire tunnel interface. To achieve this, provide the appropriate oid of the var bind for the trap to be associated with this pseudowire tunnel interface while providing inputs for "snmp-trap-entity-from-oid" template. The tunnel interface index provided by "snmp-trap-entity-from-oid" template, becomes the automatic input to the current "create-pw-interface-key-from-tunnelindex" template.

No input is required.

## Prime Network Event Templates

These templates create a unique location ID for the event using the device component key and unique ID information created in the previous rules.

### create-ana-trap-event

Template to create a unique location ID for a Prime Network trap event.

#### Mandatory Input

**type**—A string that specifies the event name.




---

**Note** This string should match the event name that was used to create the Prime Network event using the **vcb event add** command. View the event name using the **vcb event view** command.

---

#### Optional Input

**subtype**—A string that specifies the event subtype name.




---

**Note** The subtype parameter is mandatory for INFO events because INFO events do not have subtypes. For events that have subtypes, the subtype parameter is not needed.

---

#### Input Format

```
-create_ana_trap_event_type type
-create_ana_trap_event_subtype subtype
```

### create-ana-syslog-event

Template to create a unique location ID for a Prime Network syslog event.

#### Mandatory Input

**type**—A string that specifies the event name. Mandatory for an INFO event only.




---

**Note** This string should match the event name used for creating the Prime Network event using the **vcb event add** command.

---

#### Optional Input

**subtype**—A string that specifies the event subtype name.



---

**Note** The subtype parameter is mandatory for INFO events because INFO events do not have subtypes. For events that have subtypes, the subtype parameter is not needed.

---

#### Input Format

```
-create_ana_syslog_event_type type  
-create_ana_syslog_event_subtype subtype
```



---

**Note** For more information about **vcb eventparsingrules** commands, see [VCB CLI Reference: vcb eventparsingrules Commands](#), page B-40.

---

