CHAPTER 21

Managing Additional Event Types Using the VCB

This chapter contains the following sections:

Scenarios in Which to Add Additional Event Types

Use the VCB to customize events, for example, when you want to do any of the following:

- Enable Cisco ANA to recognize:
  - Traps that are specific to a new technology or device defined in a MIB. See Adding Unsupported Traps from a MIB to Cisco ANA as Events, page 21-1.
  - A syslog that you have defined. See Configuring a Custom Syslog, page 21-2.
- Change the settings for a factory-defined Cisco ANA event with respect to ticketing and whether or not the ticket autoclears. See Making a Factory-Defined Event Ticketable and Preventing it from Autoclearing, page 21-4.

For additional and advanced examples, see Event Customization—Additional Examples, page 21-44.

Adding Unsupported Traps from a MIB to Cisco ANA as Events

This procedure shows you, briefly, how to add unsupported traps as events in Cisco ANA based on a particular MIB definition file.

Step 1 Obtain and copy MIB modules to a folder on the Cisco ANA server. You must include all dependent MIB modules.

Step 2 Remove the .my file extensions from the MIB module filenames and ensure that the MIB filename and MIB module name are the same.

Step 3 Execute the **vcb event view** command to confirm that an input MIB includes traps that are not supported as events in Cisco ANA. For example, when you have copied MIBs to a /mibs folder on the Cisco ANA server, check the OSPF-TRAP-MIB by entering this command:

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

Step 4 To generate a VCB script file that will add the unsupported traps from the OSPF-TRAP-MIB, execute the **vcb event view** command with the **–generatecli** option:

```
vcb event view -mibfile /mibs/OSPF-TRAP-MIB -generateeventcli -user root -password admin
```

Run the output script file:

```
$ANAHOME/Main/VcbEventCommand.sh root admin
```

Note For changes to take effect, you must restart the VNEs that are affected.

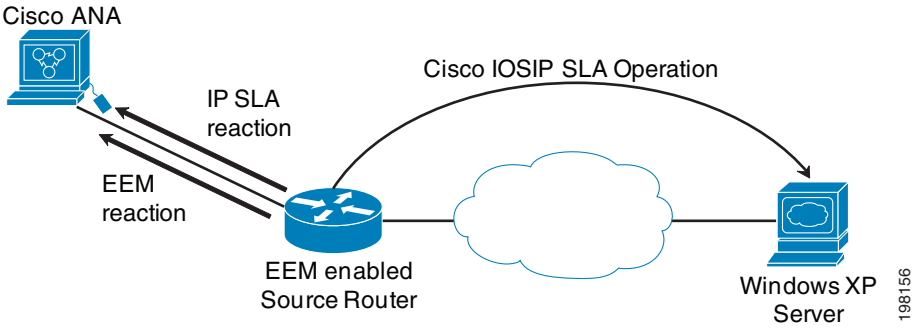
For a more detailed example, see Supporting Traps from a MIB—A Step-by-Step Example, page 21-44.

Configuring a Custom Syslog

Figure 21-1 depicts a scenario where Cisco IOS IP SLA is being used to monitor a Windows XP server from a source router where Embedded Event Manager (EEM) is also enabled. When the server is not reachable from the router, IP SLA triggers a syslog, %HA_EM-6-LOG: IPSLA-XP: Windows-XP no reachable, which has been configured on the source router. The source router, using EEM, generates a customized syslog. Cisco ANA will parse the syslog to create an event and a ticket in Cisco ANA NetworkVision.

Note How to enable EEM and create a customized syslog—or define any other action on the source router—is outside of the scope of this document.

Figure 21-1 Using EEM to Send a Customized Syslog to Cisco ANA



The following procedure shows the VCB commands to use to enable Cisco ANA to recognize the customized syslog.

Step 1 Add a ticketable event for the customized syslog by entering this command:

```
vcb event add -user root -password admin -eventtype syslog -eventname MonitoringXP
-subtype1 "Server monitoring syslog" -ticketable1 -autoclear1 false -severity1 major
-shortdesc1 "Video Service in Windows XP server is not accessible" -user root
-password admin
```

Where:

- eventname—Must be a unique value.

  To search for an event name, list all events using the **vcb event view -eventname** *all* command and use the grep command to search the list as shown here:

  ```
  vcb event view -eventname all -user root -password admin | grep MonitoringXP
  ```

- eventtype—Must be *syslog*.

  For more information, see vcb event add, page 21-23.

Step 2 Add event parsing rules for the customized syslog. To create event parsing rules, the VCB uses multiple event templates to extract information and create keys at runtime. (For more information, see Why Use Event Templates?, page 21-4.) Enter this command:

```
vcb eventparsingrules add -enable -rulename IP-SLA-Rule
-templates syslog-identification,create-managedelement-key,create-ana-syslog-event
-group cisco-syslog-repository -syslog_identification_expression "%HA_EM-6-LOG:
IPSLA-XP: Windows-XP no reachable" -create_ana_syslog_event_type MonitoringXP
-create_ana_syslog_event_subtype "Server monitoring syslog" -user root -password admin
```

Where:

- syslog_identification_expression—Matches the customized syslog, %HA_EM-6-LOG: IPSLA-XP: Windows-XP no reachable that was created on the source router.
- create_ana_syslog_event_type and create_ana_syslog_event_subtype—Match the event name and event subtype, respectively, provided in Step 1.
- group—cisco-syslog-repository is a standard Cisco syslog repository

For more information, see Event Templates, page 22-27.

Step 3 Add an event pattern to associate the event parsing rule with a device family or a scheme. Enter this command:

```
vcb eventpattern add -user root -password admin -rulename IP-SLA-Rule
-group cisco-syslog-ipcore-parsing-rules -repository cisco-syslog-repository
```

Where:

- rulename—Matches the rule name, IP-SLA-RULE, provided in Step 2.
- group—Associates the event pattern with cisco-syslog-ipcore-parsing-rules.
- repository—Links the event pattern to the actual parsing rules in cisco-syslog-repository.

Note For changes to take effect, you must restart the VNEs that are affected.

Making a Factory-Defined Event Ticketable and Preventing it from Autoclearing

Out-of-the-box, Cisco ANA does not ticket the "mep missing trap" event. To ticket the event and prevent the ticket from autoclearing, use the **vcb event modify** command with the **-override** option. Enter the following command:

```
vcb event modify -eventname "cisco cfm crossconnect mep missing trap"
-subtype1 "mep missing trap" -ticketable -autoclear1 false -override -user username
-password password
```

Where:

- eventname—cisco cfm crossconne ct mep missing trap is the event type
- subtype1—mep missing trap is the event state (event subtype) for which we want to add a ticket and set autoclear to false

To look up factory-defined event types and event subtypes, see *Cisco Active Network Abstraction 3.7.1 Reference Guide* or use the **vcb event view** command (see vcb event view, page 21-26).

To restore the event to factory defaults, use the **vcb event delete** command. Deleting a factory-defined event removes only the event attribute overrides, not the event itself. For more information, see vcb event delete, page 21-30.

Note For changes to take effect, you must restart the VNEs that are affected.

Why Use Event Templates?

Event templates simplify and streamline the process of event parsing. Event templates can be characterized as follows; they:

- Enable the creation of user-defined parsing rules for a specific network event.
- Are vendor-independent and common for all VNEs.
- Differ from U-VNE and module templates:
  - Multiple event templates must be selected, whereas for a U-VNE or a module, a single template is sufficient to add support.
  - Event templates are collated into a single template at runtime.
  - Event templates contain placeholders for event-specific data.

There are six template categories. To create event parsing rules, the VCB takes event-specific data that you supply and event templates that you select and processes them to create event-specific parsing rules; see Figure 21-2.

Figure 21-2 Event Template Process

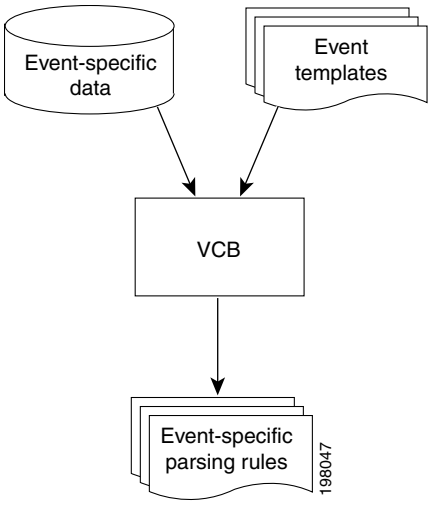


Table 21-1 lists the event template categories and their usage.

Table 21-1 Purpose of each Event Template Category

Template Category	Templates in this Category Contain Rules to	For a Syslog	For a Trap
event-identification-templates	Extract event ID	X	X
	Extract: • Event subtype • Entity ID • Unique ID	X	—
event-subtype-templates	Extract event subtype	—	X
	Perform event subtype mapping	X	X
entity-id-templates	Extract entity ID	—	X
entity-key-templates	Create device component key from the entity ID **Note** Uses the entity ID that was extracted by a template in either the event-identification-templates or the entity-id-templates categories	X	X
uniqueid-templates	Extract unique ID	—	X
ana-event-templates	Create the source OID and the Cisco ANA event.	X	X

Note Event templates are common across vendors and device types.

In addition to the functions that they serve, event template categories can be further divided into those that are mandatory and those that are optional; see Table 21-5.

Performing Event Customization

This section describes how to perform the customizations that enable you to manage additional traps and syslogs as events in Cisco ANA and includes the following topics:

Table 21-2 provides the overall process that you should follow.

Table 21-2 Event Customization Process

	Phase	Procedures
Step 1	Investigation—Includes: • Identifying the events to add—Such as, traps and syslogs that Cisco ANA receives but does not process; unsupported traps in a MIB; and so on. • Gathering information about the trap or the syslog—Such as, event subtypes and a unique identifier for the event.	• Planning—Researching Events, page 21-8 • For examples, see the following: – Investigate the cevcEvcCreationNotification Trap, page 21-49 – Investigate the FWSM-5-111004 Syslog, page 21-53
Step 2	Customization—Create three definitions using the following three **vcb** commands:	
	• **vcb event add**	• Adding an Event, page 21-9
	• **vcb eventparsingrules add**	• Adding Event Parsing Rules, page 21-11
	• **vcb eventpattern add**	• Adding Event Patterns, page 21-16
	Note Examples that show using all three commands are included in this chapter.	• For a basic example, see Configuring a Custom Syslog, page 21-2 • For advanced examples, see Event Customization—Additional Examples, page 21-44
	Note When you customize, ensure that you follow best practices.	Documenting Your Customization for Ease-of-Use, page 21-7.
Step 3	Test and certification—Confirm in a lab environment that Cisco ANA recognizes and processes the event and that the event parameters—type, subtype, severity, and so on—are correct.	Testing and Certifying Event Customizations, page 21-17

Understanding Events, Event Parsing Rules, and Event Patterns

This topics provides definitions for event, event parsing rules, and event pattern:

- Event—System-level event definition that includes an alarm ID, event subtypes, if any, and event subtype parameters such as severity, ticketability, autoclear, and short description. The alarm types and Cisco ANA events that you define are added at the VNE framework level. See Configuring Events, page 21-7.
- Event Parsing Rules—Vendor-level definitions that specify how to parse the incoming network event to:
  - Uniquely identify the syslog or trap.
  - Associate the event with the appropriate entity in the VNE.
  - Create an instance of a Cisco ANA event.

  Note Application-specific parameters such as correlation, flapping, and expedite are not currently supported.

  See Configuring Event Parsing Rules, page 21-10.
- Event Pattern—Scheme or VNE type-level definition that associates event parsing rules with a device family or a scheme. See Configuring Event Patterns, page 21-16.

Documenting Your Customization for Ease-of-Use

VCB commands for event customization are slightly long, especially for the **vcb eventparsingrules** command. Before executing VCB event support commands on the Cisco ANA server, document the data and write the complete **vcb** commands that you plan to use in a text file. Writing commands in a text file, enables you to:

- Copy-paste the commands from the text file into the Cisco ANA server command window.
- Quickly correct any syntax errors in the text file and execute the command again in the CLI.
- Maintain a list of the customizations that you have done.
- Easily copy input that you must reuse across commands. For example, if subtype name is required input for an event template, you must supply the same subtype name that you used in the **vcb event add** command.

For an example that walks you through the process, see Adding a Trap—A Step-by-Step Example, page 21-48.

Configuring Events

See the following topics:

- Planning—Researching Events, page 21-8
- Viewing an Event, page 21-8
- Adding an Event, page 21-9

Planning—Researching Events

To obtain lists of unsupported traps and syslogs, look at the information in Cisco ANA EventVision and VNE documentation. Also, use **vcb** commands to list unsupported traps for a MIB and to list all traps and syslogs that have been sent to Cisco ANA:

- To list unsupported traps for a MIB, use the **vcb event view -mibfile** command; for an example, see Adding Unsupported Traps from a MIB to Cisco ANA as Events, page 21-1.
- To list all traps and syslogs that have been sent to Cisco ANA, use the **vcb event view -genericevents** command, as follows:

  vcb event view -genericevents *all | trap | syslog* [**-ipaddress** *vneip*] [**-date** *yyyy-mm-dd*] [**-time** *hh:mm:ss*] [**-maxrecords** *num*]

  The traps and syslogs are displayed in raw format. By default, the command returns the most recent 100 records.

The following example illustrates using the **vcb event view -genericevents** command option to list traps for a particular NE:

```
ana37@vne-ch-dev-v210 [~/Main]% vcb event view -genericevents trap -ipaddress
10.77.204.185 -time 18:39:26 -user root -password admin -maxrecords 5


Event ID: trap-13127
  NEIP: 10.77.204.185
  Type: .1.3.6.1.4.1.9.9.613.0.0.1
  Time: 2009-12-02 18:39:26
  Raw Event:
.1.3.6.1.2.1.1.3.0=429 days, 14 hours, 47 minutes, 16 seconds.
.1.3.6.1.6.3.1.1.4.1.0=.1.3.6.1.4.1.9.9.613.0.0.1
.1.3.6.1.4.1.9.9.613.1.3.2.1.1=4
```

The example requests up to 5 traps for the NE with IP address 10.77.204.185 at 18:39:26. One trap is returned; the trap OID is 1.3.6.1.4.1.9.9.613.0.0.1. For more information about the **vcb event view** command, see Viewing an Event, page 21-8.

After you have obtained the trap OID or the syslog mnemonic, find out whether there are subtypes for the event, and look for a unique ID. Continue to research the event using other sources such as:

- SNMP Object Navigator—For an example, see Investigate the cevcEvcCreationNotification Trap, page 21-49.
- MIB—Look at a MIB or use a MIB browser.
- Vendor documentation and data sheets.
- Vendor web sites—For an example, see Investigate the FWSM-5-111004 Syslog, page 21-53.

Viewing an Event

Use the **vcb event view** command to view a list of supported events and to obtain event properties for a single supported event, all supported events, or a subset of supported events that match a substring. Use this command to view supported events for a particular technology by entering a substring such as bgp.

Also, use this command to view a list of unsupported traps, or all traps and syslogs that Cisco ANA has received.

To view event configurations, enter one or more of the commands listed in Table 21-3.

Table 21-3 VCB Event View Commands

To View...	Use This Command...
All supported events that match a substring	**vcb event view -eventname** *event* **-substringmatch -user** *username* **-password** *password*
All supported events including those that were added using the VCB	vcb event view -eventname *all* **-user** *username* **-password** *password*
All traps and syslogs that Cisco ANA has received, supported or not, shown in raw format	**vcb event view -genericevents** *all* **-user** *username* **-password** *password*
All syslogs that Cisco ANA has received, supported or not, shown in raw format	**vcb event view -genericevents** *syslog* **-user** *username* **-password** *password*
All traps that Cisco ANA has received, supported or not, shown in raw format	**vcb event view -genericevents** *trap* **-user** *username* **-password** *password*
All traps that Cisco ANA has received, supported or not, for a particular VNE (shown in raw format	**vcb event view -genericevents trap -ipaddress** *vneip* **-user** *username* **-password** *password*
All syslogs that arrived on a particular date supported or not, shown in raw format	**vcb event view -genericevents** *syslog* **-date** *yyyy-mm-dd* **-user** *username* **-password** *password*

Tip Use the **| more** filter with view commands to display the output one screen at a time. For example, enter the command **vcb event view -eventname** *all* **-user** *username* **-password** *password***| more** to display the list of all supported events. Press **Enter** to display the next screen of information.

Adding an Event

When you add an event to Cisco ANA, you supply event attributes such as subtype, severity, and ticketability. Enter the following VCB commands on the Cisco ANA gateway:

vcb event add -eventtype *syslog|trap* **-eventname** *eventName* **-subtypen** *subtypeName* **-severityn** *severityvalue* **-shortdescn** *shortdescription* **...-user** *username* **-password** *password*

Where:

- syslog|trap—Is either syslog or trap, depending on the type of event that it is.
- eventName—Is a unique string that you must provide to identify the event in Cisco ANA.
- subtype*n*—Is a string that you provide. The value of *n* is a number from 1 to 10, enabling you to add a maximum of 10 subevents to an event.

Note If you do not supply a value for the **-severity***n* option, the default severity, info, is applied to the subtype.

- shortdescription—Is a short description that is displayed in the Cisco ANA UI.

Some command options and arguments were not supplied in the command above. They are set to default values as follows:

- alarmid—The VCB generates a unique alarm ID.

Note Cisco recommends that you do not supply an alarm ID when you add an event; the VCB can then look up existing alarm IDs and provide one that is not already in use.

- -ticketable—Because this option was not provided for either subtype, Cisco ANA does not write a ticket for either subevent.
- -autoclear—Because this option was not provided, by default the value is set to true. Because the event is not ticketable, the autoclear setting has no effect.

After you add an event, view it to ensure that the configuration is what you intended; see Viewing an Event, page 21-8.

Related Topic

vcb event, page 21-23

Configuring Event Parsing Rules

Event parsing rules are vendor-level definitions that specify how to parse an incoming network event to:

- Uniquely identify the syslog or trap.
- Associate the event with the appropriate entity in the VNE (or U-VNE).
- Create an instance of a Cisco ANA event.

This section contains the following topics:

Viewing Event Parsing Rules

Use the **vcb eventparsingrules view** command to view the parsing rules in a repository for an event or all events. Use this command to view parsing rules in a template or all templates and to view input parameters for templates, including whether each parameter is mandatory or optional.

To view event parsing rules configurations, enter one or more of the commands listed in Table 21-4.

Table 21-4 VCB Eventparsingrules View Commands

To View...	Use This Command...
Event parsing rules for all events that are defined in a repository	**vcb eventparsingrules view -group** *repository name* **-rulename** *all* **-user** *username* **-password** *password*
Event parsing rules for an event that is defined in a repository	**vcb eventparsingrules view -group** *repository name* **-rulename** *ruleName* **-user** *username* **-password** *password*
Event parsing rules that are defined in a template	**vcb eventparsingrules view -template** *template name* **-user** *username* **-password** *password*
Event parsing rules defined in all templates	**vcb eventparsingrules view -template** *all* **-user** *username* **-password** *password*
Input parameters for an event template	**vcb eventparsingrules view -template** *templateName* **-inputparam**
Input parameters for all event templates	**vcb eventparsingrules view -template** *all* **-detail -user** *username* **-password** *password*

Tip Use the **| more** filter with view commands to display the output one screen at a time. For example, enter the command **vcb eventparsingrules view -template** *all* **-user** *username* **-password** *password***| more** to display the list of parsing rules defined in all templates. Press **Enter** to display the next screen of information.

Related Topic

vcb eventparsingrules, page 21-31.

Adding Event Parsing Rules

To add event parsing rules, you must select multiple event templates and, for each template, provide any required input. For an overview of event templates, see Why Use Event Templates?, page 21-4. For more information, see Event Templates, page 22-27.

Adding event parsing rules to Cisco ANA is only one of three steps required to support a new event in Cisco ANA. (For the other two steps, see Adding an Event, page 21-9 and Adding Event Patterns, page 21-16.) You must perform all three steps for an event to be supported.

Enter the following VCB commands on the Cisco ANA gateway:

vcb eventparsingrules add -templates *templateName1, templateName2, …, templateNamen* **-group** *hiveName* **-rulename** *ruleName* [**-enable** *true|false*]
{ [**-arg1** *arg1Value*...**-argN** *argNValue*] }**-user** *username* **-password** *password*

Where:

- templateName1, templateName2....,templateName*n* —Are event templates that contain rules to perform the following functions:
  - Extract information—For example, event identification, event subtypes, and entity ID.
  - Create information—For example, entity key, source key, and an event (to be created for the device component).

Depending on the event type and the information that is available for the event (from your investigation of it), you typically need to select and enter about four templates.

Note For more information, see Selecting Event Templates, page 21-12.

- hiveName—Is the vendor-specific trap or syslog repository file under which the customizations should be made. For more information, see Selecting Parsing Rules Files and Repositories, page 21-14.
- ruleName—Is a string that is used as a key name for event rule definition.
- enable—Is a toggle that declares whether or not the rules are used to parse incoming events. Default value is true.
- Arg1value, Arg2value, etc—Provide the input for each template; the amount of input can vary from none to one or more inputs. The kind of input required also varies.

  An argument name consists of the template name plus the entry. To view the entries for each template, use the **vcb eventparsingrules view -template** *all* **-inputparam** command or see Event Templates Input Summary—Required and Optional Input, page 22-30.

Related Topic

vcb eventparsingrules, page 21-31

Selecting Event Templates

To determine which event templates to use, you need to know that some template categories are mandatory and others are optional:

- Mandatory templates—You must select and use one template per each mandatory template category to add support for a trap or a syslog.
- Optional templates—When you have researched and found that the relevant information is available for the trap or syslog that you want to support, you must select and use optional templates (select one template per optional template category).

Note The more information that you provide, the better your customization can be.

Another factor that determines which template to use is whether the event you are adding is for a trap or a syslog. Table 21-5 lists the templates in each category and whether they are mandatory or optional and applicable to a trap or a syslog.

Table 21-5 Event Parsing Templates Selection Guide for Traps and Syslogs

	Templates That Are Applicable for...	
Template Categories	**Trap**	**syslog**
Mandatory Template Categories		
Select Up to One Template from Each Mandatory Template Category		
event-identification-templates	snmp-trap-identification—You must select this template to add support for a trap. It extracts information to identify the event.	syslog-identification—You must select this template to add support for a syslog. It extracts information to identify the event.

Table 21-5 Event Parsing Templates Selection Guide for Traps and Syslogs (continued)

Template Categories	Templates That Are Applicable for...	
	Trap	**syslog**
entity-key-templates	You must select one of these templates based on where to associate the event, the managed element device component or an interface device component: • create-managedelement-key—Use to associate the event to the ManagedElement device component. • create-interface-key-from-ifIndex—Use to associate interface-level traps to appropriate interface device components. See Supported Interface Types, page 22-28. • create-interface-key-from-ifName— Use to associate interface-level events (usually syslogs) to appropriate interface device components. See Supported Interface Types, page 22-28.	
ana-event-templates	create-ana-trap-event—When you add support for a trap, you must select this template to create the source OID and the Cisco ANA event.	create-ana-syslog-event—When you add support for a syslog, you must select this template to create the source OID and the Cisco ANA event.
Optional Template Categories[1]		
Select Up to One Template Based on Need from Each Optional Template Category		
event-subtype-templates	When the trap you want to support has subtypes, use one of these templates; select the template to use based upon the input that you have available: • snmp-trap-subtype-from-oid • snmp-trap-subtype-from-trapoid • snmp-trap-subtype-from-value	When the syslog that you want to support has subtypes, use this template to map the subtype value to the event subtype: • syslog-subtype-from-expression **Note** Subtypekey is extracted by the syslog-identification-template.
entity-id-templates	For interface-level events: • snmp-trap-entity-from-oid • snmp-trap-entity-from-value	— **Note** Entity ID is extracted by the syslog-identification-template.
unique-id-templates	• snmp-trap-identifier-from-oid • snmp-trap-identifier-from-value	— **Note** Unique ID is extracted by the syslog-identification template.

1. When selecting optional templates, take into account the information that is available for the event.

For the input that you need to provide with each template, see Event Templates Input Summary—Required and Optional Input, page 22-30. To determine whether to use optional templates and, if so, which of them to use, investigate the trap or the syslog; see Planning—Researching Events, page 21-8.

Related Topic

Why Use Event Templates?, page 21-4

Selecting Parsing Rules Files and Repositories

The **vcb eventparsingrules** and **vcb eventpattern** commands require you to enter values for the **-group** option as shown in Table 21-6. Supplying an argument for the **vcb eventparsingrules -group** option is necessary, but the selection is not critical. On the other hand, the group that you select for the **vcb eventpattern -group** argument is critical.

Table 21-6 The -group and -repository Options

Command Mode	Option	Argument	Comment
eventparsingrule	-group	Select from the Repository Files column of Table 21-7.	For event parsing rules, grouping is for convenience. The choice of group is not critical. Whichever group you choose, you must supply the same argument for the **vcb eventpattern -repository** option.
eventpattern	-group	Select from the Parsing Rules Files column of Table 21-7.	This value controls whether the event is supported. Guidelines for selecting the value are supplied in Table 21-7. **Note** If this value is not set correctly, the event is not supported because it is not associated with the correct NE types.
eventpattern	-repository	Must match the argument for the **vcb eventparsingrules -group** option.	For an example, see Adding Support for a Syslog with Event Subtypes—A Step-By-Step Example, page 21-53.

When you add event support, you add it at the group level. The event is supported for all VNEs (and U-VNEs) in the group. For an example of the process of adding event support at the group level, see Adding Support for an Event to an NE Type—An Advanced Example, page 21-57. In brief:

1. Identify the NE type in the Group column in Table 21-7.
2. Select among the repository files in Table 21-7. Separate repository files exist for syslogs and for traps. The default and most commonly-used repositories are:
   - cisco-trap-repository

     Note If you must select between cisco-trap-repository and mib2-trap-repository, choose mib2-trap-repository only when adding a standard MIB-II trap.

   - cisco-syslog-repository

     Note If you must select between cisco-syslog-repository and cisco-router-iox-syslog-repository, chose cisco-router-iox-syslog-repository when a particular syslog is applicable to IOX devices only.

   Supply the repository filename as an argument for the **vcb eventparsingrules** *command* **-group** option and for the **vcb eventpattern** *command* **-repository** option.
3. Select the parsing rules file based on the advice given in Table 21-7. Supply the parsing rules file as the argument for the **vcb eventpattern -group** option.

Table 21-7 lists the repositories and parsing rules files by group.

Table 21-7 Parsing Rules and Repositories

Group	Parsing Rules Files	Repository Files
Cisco Default Group		
Use these default hives and repositories unless the NE belongs to another group in this table	Select based on the scheme and whether you are adding support for a trap or a syslog: • product scheme: – cisco-syslog-product-parsing-rules – cisco-trap-product-parsing-rule • ipcore scheme: – cisco-syslog-ipcore-parsing-rules – cisco-trap-ipcore-parsing-rules **Note** For scheme definitions, see *Cisco Active Network Abstraction 3.7.1 Administrator Guide*.	Select based on whether you are supporting a syslog or a trap: • Syslog: – cisco-syslog-repository • Trap: – cisco-trap-repository – mib2-trap-repository[1]
Cisco Non-Default Groups		
Cisco CRS-1 Carrier Routing System	Select based on the scheme and whether you are adding support for a trap or a syslog: • ipcore scheme: – cisco-iox-syslog-ipcore-parsing-rules – cisco-iox-trap-ipcore-parsing-rules • product scheme: – cisco-iox-syslog-product-parsing-rules – cisco-iox-trap-ipcore-parsing-rules	Select based on whether you are supporting a syslog or a trap: • Syslog: – cisco-syslog-repository – cisco-router-iox-syslog-repository[2] • Trap: – cisco-trap-repository – mib2-trap-repository[1]
Cisco ASR 9000 Series Aggregation Services Routers	cisco-asr90xx-syslog-ipcore-parsing-rules	• cisco-syslog-repository • cisco-router-iox-syslog-repository[2]
	cisco-asr90xx-trap-ipcore-parsing-rules	• cisco-trap-repository • mib2-trap-repository[1]
Cisco Security Appliances	cisco-asa-syslog-product-parsing-rules	cisco-syslog-repository
	cisco-asa-trap-product-parsing-rules	• cisco-asa-trap-repository[3] • cisco-trap-repository
Cisco Service Control Engines	cisco-sce-syslog-product-parsing-rules	cisco-syslog-repository
	cisco-sce-trap-product-parsing-rules	cisco-sce-trap-repository
Generic UVNEs		
See U-VNE Templates, page 22-1	• genericuvne-syslog-parsing-rules • genericuvne-trap-parsing-rules	• genericuvne-syslog-repository • genericuvne-trap-repository

1. Select mib2-trap-repository when you are adding a standard MIB-II trap. Otherwise, select cisco-trap-repository.
2. Select cisco-router-iox-syslog-repository when you are adding a syslog that applies to IOX devices only. Otherwise, select cisco-syslog-repository.
3. Select cisco-asa-trap-repository only when you are adding a trap that applies to Cisco Security Appliances only. Otherwise, select cisco-trap-repository.

Configuring Event Patterns

An event pattern associates event parsing rules with a device family or a scheme. Use the VCB to view and add event pattern configurations, as described in the following sections:

Viewing Event Patterns

To view event patterns, enter one or more of the commands listed in Table 21-8.

Table 21-8 VCB Eventpatterns View Commands

To View...	Use This Command...
The list of eventpatterns in a parsing rules hive	**vcb eventpattern view -group** *parsingrules hive* **-rulename** *all* **-user** *username* **-password** *password* To list the events that are supported for a device type, provide the parsing rules hive for the device type that interests you; (see Table 21-7).
The pattern ID that was automatically generated for an eventpattern that you added	**vcb eventpattern view -group** *parsingrules hive* **-rulename** *rulename* **-user** *username* **-password** *password* **Note** You need the pattern ID to modify or delete an event pattern.
The complete configuration, including event, event parsing rules, and event pattern	**vcb eventpattern view -group** *parsingrules hive* **-rulename** {*rulename* | **all** } **-full -user** *username* **-password** *password* Use this command to statically test your event customization.
All event patterns for a particular technology	**vcb eventpattern view -group** *parsingrules hive* **-rulename** *bgp* **-substringmatch -user** *username* **-password** *password* Returns patterns for rules that include the substring bgp.

Adding Event Patterns

Use the **vcb eventpattern add** command to create a pointer from the parsing rules file to the parsing rules defined in the repository file.

Note For every device, 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.

Enter the following VCB commands on the Cisco ANA gateway:

vcb eventpattern add -group *parsing rules hive* **-repository** *parsing rules repository hive* **-rulename** *ruleName* **-user** *username* **-password** *password*

Where:

- parsing rules hive—Is a vendor-specific parsing rules file.

- parsing rules repository hive—Is a vendor-specific trap or syslog repository. It should match the repository filename used when creating the event parsing rules.
- ruleName—Is a string that is used in the key name for event rule definition. It should match the rulename used for defining event parsing rules.

Related Topics

- vcb eventpattern add, page 21-38

Testing and Certifying Event Customizations

Test and certify in your lab before moving the customizations to production:

- Preparing Your Test Environment, page 21-17
- Testing Events, page 21-18
- Moving a Tested and Certified Event Customization to Production, page 21-18

Preparing Your Test Environment

Do the following:

- Restart Cisco ANA or restart at least all VNEs/AVMs that need to support the new event.

  A restart is required because event patterns are loaded at VNE initialization only. Cisco ANA should be restarted (or at least all VNEs/AVMs that need to support the new event) to make sure that the customizations are available to the required VNEs.
- If using a simulator, enable Cisco ANA to process traps sent from a simulator by running this command and restarting avm100:

  ```
  runRegTool.sh -gs localhost set 0.0.0.0
  avm100/agents/trap/processors/snmp-processor/class
  com.sheer.metrocentral.framework.instrumentation.trap.processor.RawAgentIpSnmpEvent
  Processor
  ```
- To receive events from a device, you must configure the device with the details of the Cisco ANA server. See Configuring a Device to Send Events, page 21-17.

Configuring a Device to Send Events

As part of the testing process, you must verify that Cisco ANA receives traps and syslogs sent from the VNE or U-VNE. One easy way to begin this process is to generate a Link Down/Link Up trap on an interface.

Step 1 Connect to the device and enter the necessary commands for sending events. For example, use the following commands for devices running Cisco IOS or Catalyst OS software:

```
snmp-server host 172.20.2.160
logging trap informational
logging source-interface Loopback0
logging on
logging 172.20.2.160
```

A similar set of commands should be used for devices belonging to other manufacturers.

These commands enable the device to send traps to the specified gateway IP over port 162. Therefore, port 162 must be enabled to receive traps from the device. In addition, you must reserve port 1162 for the general trap processing by the Cisco ANA server. This task is handled by the AVM 100 process.

Step 2 Using Telnet, shut down an interface on the device (not the management interface). Assuming that the event is configured correctly (Step 1), the device should generate a link down event.

Step 3 Restart the interface. Assuming that the event is configured correctly (Step 1), the device should generate a link up event.

Testing Events

Perform testing as follows:

- Check whether the customization is OK by viewing the event pattern. To check the event pattern customization for the cevcEvcCreationNotification event for example, use this command:

  ```
  vcb eventpattern view -user root -password admin -rulename cevcEvcCreationNotification
  -group cisco-trap-ipcore-parsing-rules -full
  ```

- Send an event from an NE or from a simulator. Use Cisco ANA EventVision to verify that:
  - Cisco ANA recognizes and processes the event
  - Event parameters—type, subtype, severity, and so on—are as expected
  - Unique ID is appended to source ID (ManagedElement)

After you complete the tests, move the customization to production and test there as well; see Moving a Tested and Certified Event Customization to Production, page 21-18.

Moving a Tested and Certified Event Customization to Production

Tip Always perform customization during a scheduled maintenance window.

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

- Exporting VCB Registry Customizations, page 18-7
- Importing VCB Registry Customizations, page 18-7

After you import the changes to the production server:

1. Restart Cisco ANA or at a minimum, restart all VNEs and AVMs that need to support the new events.
2. Repeat testing and certification to ensure that the customizations are functioning as expected in your production environment. See Testing and Certifying Event Customizations, page 21-17

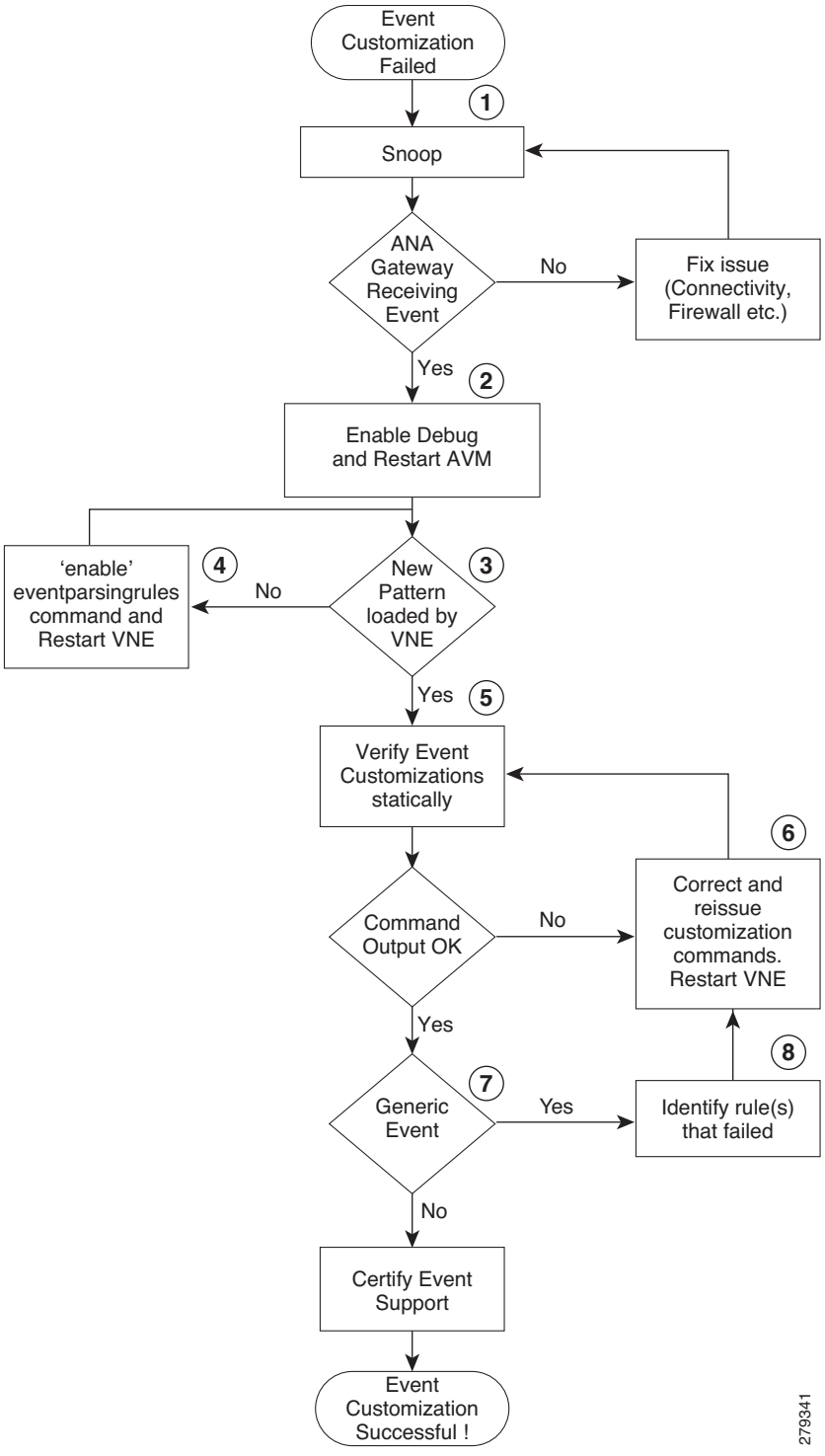
Troubleshooting Event Customization

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 18-9.

Errors that occur in the server are not as interactive and obvious. If a newly supported event does not appear in Event Vision or Network Vision, you need to perform some troubleshooting. The flowchart in Figure 21-3 depicts the event customization troubleshooting process.

Figure 21-3 Event Customization Troubleshooting Process



For more information about the steps depicted in Figure 21-3, see the following procedure.

Step 1 Use any tool to snoop and check whether the simulated network events that you are sending are actually arriving at the Cisco ANA Gateway. If not, fix the issue whether it is connectivity, firewall and so on. If OK, proceed to next step.

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 and restart the AVM (not just the VNE).

```
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 21-40). 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
p]

########## 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_0K: 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_0K: 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_0K: Fan 3 had earlier reported a rotation
error. It is ok now
  # syslog = 7.212.205 %FAN-3-FAN_0K: 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 Cisco ANA 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.

Step 9 Restart the AVM and repeat the above steps until all errors are resolved and the event is parsed correctly and the Cisco ANA event is generated as expected.

Event Customization Command Reference

This section includes the following commands:

vcb event

Use the following **vcb event** commands to create, view, modify, and delete events:

vcb event add

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

Note To create a script to add unsupported traps from a MIB, use the e **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 21-26.

Synopsis

```
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]

. . .
[-subtypen subtypenName] [-ticketablen][-severityn
critical|major|minor|warning|info|cleared ]
[ -shortdescn short description string] [-autoclearn false]

-user username -password password
```

Description

The **vcb event add** command creates an event definition in Cisco ANA 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
-alarmid 2002
-subtype1 "stack switch removed syslog"
-severity1 minor
-subtype2 "stack switch added syslog"
-severity2 cleared
-user root -password admin
```

Adds a syslog event definition with the name “stack switch status syslog” and alarm ID of 2002. 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 21-9 Options and Arguments—vcb event add

Option *Argument*	Description
eventtype *event type*	Type of event. Valid values are: • syslog • trap
eventname *event name*	Unique string that identifies the event within Cisco ANA.

Table 21-9 Options and Arguments—vcb event add (continued)

Option *Argument*	Description
alarmid *alarm ID*	(Optional) Unique integer identifier for the event. **Note** Recommendation—Do not provide this argument; the VCB automatically generates a unique number.
subtype$_n$ *subtypen name*	Unique string that identifies the subevent with the Cisco ANA event.
ticketable$_n$	(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$_n$ `false|true`	(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. **Note** Root cause events are not autocleared even when autoclear is set to false. For information about root cause events, see "Fault Management" and "Causality Correlation" in *Cisco Active Network Abstraction 3.7.1 Theory of Operations*. By default, autoclear is true for user-defined event definitions.
severity$_n$ *severity level*	Mandatory parameter for subtype. The severity of the subevent. Possible values are critical, major, minor, warning, info, and cleared.
shortdesc *short description*	Optional parameter for subtype. A short description of the subevent. This string is stored in the event database.

Note For the list of global options, see Global Command Options, page 18-9.

Error Codes

Table 21-10 Error Codes—vcb event add

Code	Description
201	Event name already exists in Cisco ANA.
202	Alarm ID already exists in Cisco ANA.

Note For the list of general VCB error codes, see General Error Codes, page 18-10.

vcb event view

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

Synopsis

vcb event view -eventname {*eventName* | *all*} [**-substringmatch**]} **-user** *username* **-password** *password*

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 Cisco ANA using the VCB.

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

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

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. (For sample output, see Using the VCB to Determine Whether any Traps in a MIB Are Unsupported, page 21-45).

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; see (Optional) Editing a Generated Script Before Running It, page 21-47. 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 Cisco ANA 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 21-11 Options and Arguments—vcb event view

Option *Argument*	Description
eventname *eventName*	Unique string that represents the event. **Tip** Enter **all** as the *eventName* to display information on all the event definitions in Cisco ANA. 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.
genericevents *generic event type*	Displays all events from the Cisco ANA database (in raw format). **Tip** Enter **all** as the *generic event type* to display information on all the events in Cisco ANA.
ipaddress *neip*	(Optional) Generic events filter. IP address for the NE for which you want to see generic events.
date *yyyy-mm-dd*	(Optional) Generic events filter. The date after which the events arrived.(Returns events that arrived after the given day.)
time *hh:mm:ss*	(Optional) Generic events filter. The time after which the events arrived. (Returns events that arrived after the given time.)
maxrecords *num*	(Optional) Generic events filter. The maximum number of events that you want to display. The default value is 100.
mibfile `complete-path-mibFilename`	Loads MIB modules and compares the traps defined in the MIB against the events that are supported in Cisco ANA. Displays lists of supported traps and unsupported traps. **Note** 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.
generatecli	(Optional) When provided, produces a script, *ANAHOME*/Main/VcbEventCommand.sh. The script contains commands to add basic event support for each unsupported trap that was identified through the **-mibfile** option.
repository *ParsingrulesHive*	Mandatory **-generatecli** option. Hive that includes event parsing rules for traps. For a list, see Selecting Parsing Rules Files and Repositories, page 21-14.
group *PatternsHive*	Mandatory **-generatecli** option. Hive that includes event patterns for traps. For a list, see Selecting Parsing Rules Files and Repositories, page 21-14.

Note For the list of global options, see Global Command Options, page 18-9.

Error Codes

Table 21-12 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 18-10.

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).

Synopsis

```
vcb event modify -eventname eventName [-alarmid alarmId] [-override]
{-subtype1 subtype1Name {[-ticketable1] [-autoclear1 false]-severity1
critical|major|minor|warning|info|cleared
-shortdesc1 short description string…
{-subtypen subtypenName [-ticketablen] [-autoclear1 false]
-severityn critical|major|minor|warning|info|cleared
[-shortdescn] short description string -user username -password password
```

Description

The **vcb event modify** command modifies an event definition in Cisco ANA based on the input provided by the user.

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 Example

```
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 21-24 for the **vcb event add**.

Options

Table 21-13 Options and Arguments—vcb event modify

Option *Argument*	Description
eventname *eventName*	Unique string identifies the event within Cisco ANA.
alarmid *alarmId*	(Optional) Unique integer identifier for the event. If not provided, the VCB automatically generates a unique number. **Note** We recommend that you do not provide an input alarm ID.
subtype*n* *subtypenName*	Unique string that identifies the subevent with the Cisco ANA 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*n*	(Optional) Parameter for subtype. Indicates whether a ticket should be generated for this subtype. If not specified, no ticket is generated. **Note** 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$_n$ `false|true`	(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. **Note** Root cause events are not autocleared even when autoclear is set to false. For information about root cause events, see "Fault Management" and "Causality Correlation" in *Cisco Active Network Abstraction 3.7.1 Theory of Operations*. By default, autoclear is true for user-defined event definitions.
severity*n* *value*	(Optional) Parameter for subtype. Specifies the severity of the subevent. Possible values are critical, major, minor, warning, info, and cleared.
shortdesc*n* `short description string`	(Optional) Parameter for subtype. A short description.
override	(Optional) Indicates that you expect to override attributes for a factory-defined event.

Note For the list of global options, see Global Command Options, page 18-9.

Error Codes

Table 21-14 Error Codes—vcb event modify

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

Note For the list of general VCB error codes, see General Error Codes, page 18-10.

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.

Synopsis

```
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 21-15 Options and Arguments—vcb event delete

Argument	Description
eventname *eventName*	Name of the event to be deleted

Note For the list of global options, see Global Command Options, page 18-9.

Error Codes

Table 21-16 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 18-10.

vcb eventparsingrules

Use the following **vcb eventparsingrules** commands to create, view, modify, and delete event parsing rules:

- vcb eventparsingrules add, page 21-31
- vcb eventparsingrules view, page 21-34
- vcb eventparsingrules modify, page 21-35
- vcb eventparsingrules delete, page 21-37

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.

Synopsis

```
vcb eventparsingrules add -templates templateName1, templateName2, …, templateNamen
-group repository -rulename rulename [-enable true|false]
{ [-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 Cisco ANA 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 Cisco ANA can differentiate customizations created using the VCB from changes supplied in VNE-driver registration files.

Usage Example

```
vcb eventparsingrules add
   -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_expression_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 Cisco ANA 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 Cisco ANA 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 21-17 Options and Arguments—vcb eventparsingrules add

Option *Argument*	Description
templates *template name1, ... template namen*	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 *repository*	Specifies the vendor-specific trap or syslog repository file under which the customizations should be made. For the repository file that you should enter, see Table 21-7.
rule *rulename*	String that is used as a key name for event rule definition.
enable *true*	(Optional) Indicates whether the rule is enabled or disabled. Only enabled rules are used to parse incoming traps and syslogs. The default value is true.
variable arguments	Arguments vary from one event template to another event template.
syslog_identification_ expression_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 18-9.

Error Codes

Table 21-18 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 18-10.

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).

Synopsis

vcb eventparsingrules view -group *repository name* **-rulename** { *rulename* | all } [**-detail**]

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 21-19 Options and Arguments—vcb eventparsingrules view

Option *Argument*	Description
group *repository name*	The trap or syslog repository filename; see Selecting Parsing Rules Files and Repositories, page 21-14.
rulename *ruleName*	The unique string that is used to represent the event parsing rules. **Tip** Enter **all** as the *ruleName* to display information on all the rules in Cisco ANA.
detail	(Optional) Lists the entire rule contents including the parsing rule entry details.
template *templateName*	The event template name. **Tip** Enter **all** as the *templateName* to display information on all event templates in Cisco ANA.
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 18-9.

Error Codes

Table 21-20 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.

Note For the list of general VCB error codes, see General Error Codes, page 18-10.

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.

Synopsis

```
vcb eventparsingrules modify -templateS templateName1, templateName2, …, templateNamen
-group repository -rulename rulename [-enable true|false] [-example syslogMessage]
{ [-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 21-21 Options and Arguments—vcb eventparsingrules modify

Option *Argument*	Description
templates *template name1, ... template namen*	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 *repository*	The hive under which the customizations should be made. The hive is the vendor-specific trap or syslog repository file. For a list, see Selecting Parsing Rules Files and Repositories, page 21-14.
rulename *ruleName*	String that is used as a key name for event rule definition.
enable *true*	(Optional) Indicates whether the rule should be enabled or disabled. Only enabled rules are used to parse incoming traps and syslogs. The default value is true.
example *syslogMessage*	(Optional) Valid for syslogs only. The VCB uses this example syslog message to check the correctness of the regular expression that is automatically created by the VCB based on the user input.
variable arguments	Each event template can require different input and a different number of input parameters from none to more than one. See Event Templates Input Summary—Required and Optional Input, page 22-30.

Note For the list of global options, see Global Command Options, page 18-9.

Error Codes

Table 21-22 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 18-10.

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.

Synopsis

```
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 21-23 Options and Arguments—vcb eventparsingrules delete

Option *Argument*	Description
group *repository hive*	The hive from which to remove the event parsing rule.
rulename *ruleName*	The rule to delete.

Note For the list of global options, see Global Command Options, page 18-9.

Error Codes

Table 21-24 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 18-10.

vcb eventpattern

Use the following **vcb eventpattern** commands to create, view, modify, and delete event patterns:

- vcb eventpattern add, page 21-38
- vcb eventpattern view, page 21-40
- vcb eventpattern modify, page 21-41
- vcb eventpattern delete, page 21-43

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, (for more information, see Table 21-7) 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.

Synopsis

```
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 21-25 Options and Arguments—vcb eventpattern add

Option *Argument*	Description
patternid *patternId*	(Optional) (Recommendation: do not provide.) Unique integer to identify the supported event to VNEs. If not provided, VCB generates this number automatically. **Note** 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 *parsing rules hive*	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; see Selecting Parsing Rules Files and Repositories, page 21-14.
repository *parsing rules repository hive*	The trap or syslog repository where the actual parsing rules are defined. (See Table 21-7.) **Note** Enter the same hive that was specified when creating parsing rules registrations using the **vcb eventparsingrules add** command.
rulename *rulename*	String that is used as a key name for event rule definition. **Note** Enter exactly the same string as the one that was specified when creating parsing rules registrations using the **vcb eventparsingrules add** command.

Note For the list of global options, see Global Command Options, page 18-9.

Error Codes

Table 21-26 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 18-10.

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.

Synopsis

```
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 21-27 Options and Arguments—vcb eventpattern view

Option *Argument*	Description
group *parsingrules hive*	The parsing-rules filename used by the NE type or scheme. See Selecting Parsing Rules Files and Repositories, page 21-14.
rulename *rule name*	Unique string that represents the event parsing rules. **Tip** Enter **all** as the *rule name* to list all event parsing rules defined in the repository.
substringmatch	(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.
full	(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 Cisco ANA. **Note** 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 18-9.

Error Codes

Table 21-28 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 18-10.

vcb eventpattern modify

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

Synopsis

```
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.

Options

Table 21-29 Options and Arguments—vcb eventpattern modify

Option *Argument*	Description
patternid *patternId*	Unique integer to identify the supported event to a VNE.
group *parsing rules hive*	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. For a list, see Selecting Parsing Rules Files and Repositories, page 21-14.
repository *parsing rules repository hive*	(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 **vcb eventparsingrules add** command.
rulename *ruleName*	String that is used as a key name for event rule definition. **Note** Enter exactly the same string as the one that was specified when creating parsing rules registrations using the **vcb eventparsingrules add** command.

Note For the list of global options, see Global Command Options, page 18-9.

Error Codes

Table 21-30 Error Codes—vcb eventpattern modify

Code	Description
221	Parsing rules hive not found.

Table 21-30 Error Codes—vcb eventpattern modify (continued)

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 18-10.

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.

Synopsis

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 21-31 Options and Arguments—vcb eventpattern delete

Argument	Description
patternid *pattern ID*	Unique integer to identify the supported event to a VNE.
group *parsing rules hive*	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 18-9.

Error Codes

Table 21-32 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 18-10.

vcb eventarg

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

Synopsis

```
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.

Event Customization—Additional Examples

See the following:

- Supporting Traps from a MIB—A Step-by-Step Example, page 21-44
- Adding a Trap—A Step-by-Step Example, page 21-48
- Adding Support for a Syslog with Event Subtypes—A Step-By-Step Example, page 21-53
- Adding Support for an Event to an NE Type—An Advanced Example, page 21-57

Supporting Traps from a MIB—A Step-by-Step Example

Use the **vcb event view -mibfile** *mibfilename* to view the list of traps in the MIB and see which are supported or unsupported by Cisco ANA. As a prerequisite, you must:

1. Obtain and copy MIB modules to a folder on the Cisco ANA server. You must include all dependent MIB modules.
2. Remove the .my file extensions from the MIB module filenames.

A partial listing of MIB modules might look like this:

```
ana371@vne-dev1 [/]% ls /mibs
ACCOUNTING-CONTROL-MIB              MALLOC-MIB
```

```
ADSL-LINE-EXT-MIB                MAU-MIB
ADSL-LINE-MIB                    MIOX25-MIB
ADSL-TC-MIB                      MIP-MIB
AGENTX-MIB                       MOBILEIPV6-MIB
AGGREGATE-MIB                    MPLS-FTN-STD-MIB
ALARM-MIB                        MPLS-L3VPN-STD-MIB
APM-MIB                          MPLS-LC-ATM-STD-MIB
```

Tip Many MIBs are available for download using the Cisco SNMP Object Browser at this URL: http://tools.cisco.com/Support/SNMP/do/BrowseMIB.do?local=en&step=2.

To add traps from a MIB, follow this process:

1. Check whether any traps in a MIB are unsupported; see Using the VCB to Determine Whether any Traps in a MIB Are Unsupported, page 21-45.
2. To add unsupported traps from a MIB, generate a script; see Generating a Script to Support Traps from a MIB, page 21-46.

   Note The script includes commands to add each unsupported trap as an event without subtype support and associates the event with Managed Element.

3. To omit certain traps or to customize events, for example by adding subtype support to the event, see (Optional) Editing a Generated Script Before Running It, page 21-47.
4. Run the script; see Running the VcbEventCommand Script to Add Event Support for Traps from a MIB, page 21-48.

Using the VCB to Determine Whether any Traps in a MIB Are Unsupported

This example shows how to check whether traps from the BGP4-MIB module are supported. The syntax to use for the **vcb event view** command is:

vcb event view -mibfile *complete-path-mibFilename* **-user** *username* **-password** *password*

For example:

```
ana371@vne-dev1 [/]%  vcb event view -mibfile /mibs/BGP4-MIB -user root -password admin
```

If any necessary MIB modules are missing or if you enter the wrong path or mistype the MIB module name, an error, such as this one, is displayed:

```
Missing MIB Modules:
---------------------
BGP4-MIB
Please make sure the all the MIB modules are available for MIB parsing.
```

If an error appears, correct the problem (by copying the missing module or correcting the input) and try the command again. If no error appears, a report is displayed:

```
SUPPORTED TRAPS
---------------
Oid : .1.3.6.1.2.1.15.7.1    Name:bgpEstablished
  Rule Name    : bgp-established-trap
```

```
  File          : nexus-trap-repository
  Varbind Index: 3
  Type          : bgp trap
  SubType       : 6-bgp established trap,^-bgp down trap

Oid : .1.3.6.1.2.1.15.7.2    Name:bgpBackwardTransition
  Rule Name     : bgp-backward-transition-trap
  File          : nexus-trap-repository
  Type          : bgp trap
  SubType       : bgp down trap

UNSUPPORTED TRAPS
-----------------
Name    : bgpBackwardTransNotification   Module: BGP4-MIB
Oid     : .1.3.6.1.2.1.15.0.2   Version: V2
  Varbind : bgpPeerRemoteAddr ==> .1.3.6.1.2.1.15.3.1.7
  Varbind : bgpPeerLastError ==> .1.3.6.1.2.1.15.3.1.14
  Varbind : bgpPeerState ==> .1.3.6.1.2.1.15.3.1.2

Name    : bgpEstablishedNotification   Module: BGP4-MIB
Oid     : .1.3.6.1.2.1.15.0.1   Version: V2
  Varbind : bgpPeerRemoteAddr ==> .1.3.6.1.2.1.15.3.1.7
  Varbind : bgpPeerLastError ==> .1.3.6.1.2.1.15.3.1.14
  Varbind : bgpPeerState ==> .1.3.6.1.2.1.15.3.1.2
```

We can see that there are four traps:

- Two traps that are supported—bgpEstablished and bgpBackwardTransition.
- Two traps that are unsupported—bgpBackwardTransNotification and bgpEstablishedNotification; the report provides the trap OIDs and varbinds, information that you could use to manually add the trap using the VCB CLI.

To support the unsupported traps, generate a script; see Generating a Script to Support Traps from a MIB, page 21-46.

Generating a Script to Support Traps from a MIB

To add the unsupported traps from a MIB as Cisco ANA events, use the VCB to generate a script, and then run it to add them. Doing so provides an alternative to typing all commands yourself. The output script includes **vcb** commands to add the event, event parsingrules, and event pattern.

The syntax for generating a script follows:

vcb event view -mibfile *complete-path-mibFilename* **-generatecli - group** *ParsingRulesFilename* **-repository** *TrapRepositoryFilename* **-user** *username* **-password** *password*

To continue the example of the BGP4-MIB module, and to generate a script to add the unsupported traps, enter this command:

vcb event view -mibfile */mibs/BGP4-MIB* **-generateeventcli -group** *cisco-trap-product-parsing-rules* **-repository** *cisco-trap-repository* **-user** *root* **-password** *admin*

The output script is created in *ANAHOME*/Main and is named VcbEventCommand.sh.

Note Each time you generate a script, any existing VcbEventCommand.sh file is overwritten. To keep a record of your changes, rename the VcbEventCommand.sh file as appropriate.

To change the script before you run it, see (Optional) Editing a Generated Script Before Running It, page 21-47. Otherwise, run the script; to do so, see Running the VcbEventCommand Script to Add Event Support for Traps from a MIB, page 21-48.

(Optional) Editing a Generated Script Before Running It

For each unsupported trap, a VcbEventCommand.sh script contains commands to add an event, event parsing rules, and an event pattern. The commands of interest in the script start with $VCBPATH. Edit the content that follows $VCBPATH; however, you should keep the end of the command as given (**-user $USER -password $PASS >> "$VCT_IMPORT_SCRIPT_LOG"**). Before you edit a script, achieve a better understanding of how to form **vcb** commands by reviewing the related command reference and other examples throughout this section.

This is an example command from a VcbEventCommand.sh script:

```
$VCBPATH event add -eventtype trap -eventname bgpBackwardTransNotification -subtype1 "bgp
Backward Trans Notification" -shortdesc1 "bgp Backward Trans Notification" -user $USER
-password $PASS >> "$VCT_IMPORT_SCRIPT_LOG"
```

The command adds the bgpBackwardTransNotification trap as an event.

Note

- The **vcb** command that usually starts each command is replaced by the $VCBPATH variable.
- Even with one only subtype, you must use a subscript of 1 in the -subtype*n* option. For more information, see vcb event add.
- By default, because no **-severity***n* option is provided, the event is an INFO event.
- By default, and as recommended, because no **-alarmid***n* option is provided, the VCB generates a unique alarm ID for the event.
- Because no **-ticketable***n* option is provided, the event is not ticketable and the **-autoclear** option therefore is not applicable.

Here is the command to add event parsing rules for the bgpBackwardTransNotification trap:

```
$VCBPATH eventparsingrules add  -rulename bgpBackwardTransNotification -templates
snmp-trap-identification,create-managedelement-key,create-ana-trap-event
-snmp_trap_identification_oid .1.3.6.1.2.1.15.0.2 -create_ana_trap_event_subtype "bgp
Backward Trans Notification" -create_ana_trap_event_type "bgpBackwardTransNotification"
-group cisco-trap-repository -enable -user $USER -password $PASS >>
"$VCT_IMPORT_SCRIPT_LOG"
```

Templates—listed after the **-templates** option—create rules to extract info from the trap and associate it with the managed element. Parsing rules are stored in the cisco-trap-repository.xml file in the Cisco ANA registry. For more information, see vcb eventparsingrules add, page 21-31. Here is the command to add an event pattern for the bgpBackwardTransNotification trap:

```
$VCBPATH eventpattern add  -rulename bgpBackwardTransNotification -group
cisco-trap-product-parsing-rules -repository cisco-trap-repository -user $USER -password
$PASS >> "$VCT_IMPORT_SCRIPT_LOG"
```

Note

- By default, and as recommended, the **-patternid** option is not given, enabling the VCB to generate a unique pattern ID.
- For more information, see vcb eventpattern add, page 21-38.

Running the VcbEventCommand Script to Add Event Support for Traps from a MIB

Note To create a VcbEventCommand script, see Generating a Script to Support Traps from a MIB, page 21-46.

When you run the VcbEventCommand, it adds each unsupported trap as an event without subtype support and associates the event with Managed Element. To change the script before you run it, see (Optional) Editing a Generated Script Before Running It, page 21-47.

To run the script, type this command:

```
VcbEventCommand.sh user password
```

For example:

```
VcbEventCommand.sh root xpwd150z
```

You should see an output confirmation message.

Adding a Trap—A Step-by-Step Example

The example Adding Unsupported Traps from a MIB to Cisco ANA as Events, page 21-1 shows how to use the VCB to obtain information about traps and enable Cisco ANA to recognize them by adding them as events using an automatically generated script. This example shows how to:

- Write your own commands in a text file, building a file of commands that include everything you need to add support for the cevcEvcCreationNotification trap using the VCB CLI directly.
- Investigate the trap using the Cisco SNMP Object Navigator.

Note In Cisco ANA 3.7.1, the cevcEvcCreationNotification trap is not supported. If the trap is supported in a later release, the principles for investigating and adding support for a trap remain the same regardless.

Add the Event for the cevcEvcCreationNotification Trap

Add the trap as follows:

```
vcb event add -user root -password admin -eventtype trap -eventname "EVC created trap"
-subtype1 "EVC created trap" -severity1 info -shortdesc1 "EVC created trap"
```

There is only one subtype. The convention in Cisco ANA is to use the same name for the event and the subtype; however, you are free to use different names. By default, event severity is info. Because you are adding support for a trap, the value for the **-eventtype** option must be trap. For more information, see vcb event add, page 21-23.

Before adding event parsing rules, you must investigate the trap.

Investigate the cevcEvcCreationNotification Trap

One way to get information about a trap to use the Cisco SNMP Object Navigator at this URL:

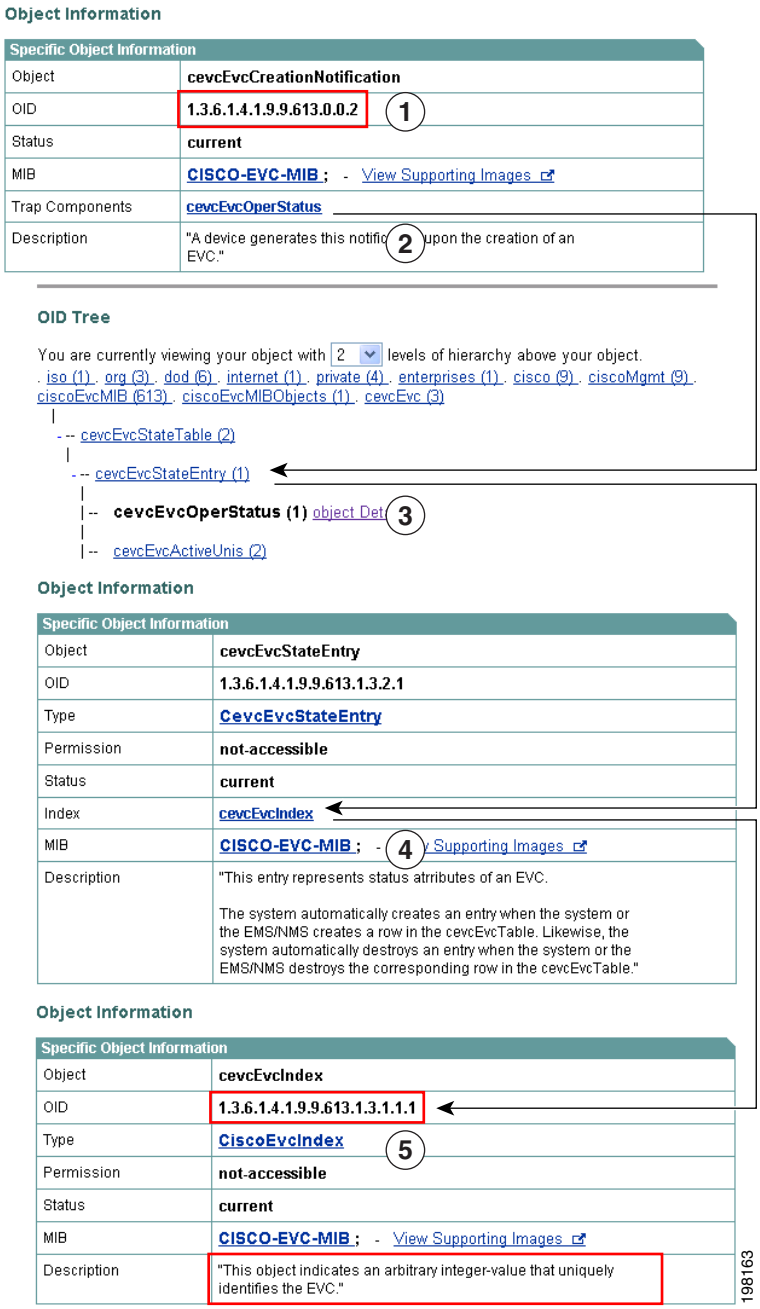
http://tools.cisco.com/Support/SNMP/do/BrowseOID.do?local=en

Note As an alternative, look at the MIB.

To get started, enter the object identifier, cevcEvcCreationNotification, and click **Translate**.

The trap is recognized and Object Information is displayed. Figure 21-4 shows the object information and successive screens of data that result from clicking the links to drill down.

Figure 21-4 Using the SNMP Object Navigator to Investigate a Trap



1	The trap OID, 1.3.6.1.2.99.613.0.0.2, uniquely identifies the event (and is useful input for an event identification template)	**4**	After clicking cevcEvcStateEntry, object information shows that cevcEvcStateEntry is indexed by cevcEvcIndex.
2	There is only one trap component, cevcEvcOperStatus.	**5**	The cevcEvcIndex OID is a unique identifier.
3	After clicking cevcEvcOperStatus and scrolling down, the OID tree shows that the cevcEvcOperStatus in defined in cevcEvcStateTable.		

Add Event Parsing Rules for the cevcEvcCreationNotification Trap

This example shows you how to build the **vcb eventparsingrules add** command to add support for the cevcEvcCreationNotification trap. This is the command syntax that you will use:

vcb eventparsingrules add [global args] [-enable] -rulename *name* **-group** *repository-hive* **-templates** *comma-separated-list-of-templates variable_args…*

One step at a time, build the necessary command as follows:

- Enter the Rulename and the Group, page 21-51
- Select and Enter the Appropriate Templates for the Trap, page 21-51
- Enter Arguments for the Templates for the Trap, page 21-52
- Putting It All Together and Running the Complete Command, page 21-52

Enter the Rulename and the Group

To start assembling the command, enter the following:

vcb eventparsingrules add -enable -rulename *cevcEvcCreationNotification* **-group** *cisco-trap-repository*

To supply values for:

- rulename—Match the name of the MIB object, cevcEvcCreationNotification.
- group—Enter cisco-trap-repository. For more information, see Selecting Parsing Rules Files and Repositories, page 21-14.

Tip Use a text editor to create the command. The advantages are that you keep a record of what you have done and can quickly correct any errors.

Select and Enter the Appropriate Templates for the Trap

Next, select the templates that are appropriate for adding a trap. The information in Table 21-5 can guide you in your selection. You must select the templates that are mandatory for a trap:

- snmp-trap-identification
- create-managedelement-key
- create-ana-trap-event

Because you found a unique ID (the cevcEvcIndex), select this optional template also:

- snmp-trap-identifier-from-oid

Enter the templates as a comma-separated list after the **-templates** option as follows:

-templates *snmp-trap-identification, create-managedelement-key, create-ana-trap-event, snmp-trap-identifier-from-oid*

Note The order in which you list the templates is not important.

Enter Arguments for the Templates for the Trap

You command so far includes rulename, group, and templates:

```
vcb eventparsingrules add -user root -password admin -enable
-rulename cevcEvcCreationNotification
-group cisco-trap-repository
-templates
snmp-trap-identification,snmp-trap-identifier-from-oid,create-managedelement-key,
create-ana-trap-event
```

Next, you must supply any mandatory input for the templates; you should also supply any optional input that you have. To see the arguments for the templates, do one of the following:

- List the arguments using one of these VCB commands:

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

  This command returns all event templates and lists the arguments for them.

  ```
  vcb eventparsingrules view -template template-name -inputparam username -password
  password
  ```

  This command returns the arguments for a specific event template.

- See Event Templates Input Summary—Required and Optional Input, page 22-30.

After you list the argument names, add them to your command and supply the values that you have acquired through your research into the event. Table 21-33 provides a summary.

Table 21-33 Template Arguments for Adding Event Parsing Rules for the cevcEvcNotification Trap

Template	Argument	Value	Command Input
snmp-trap-identification	oid	1.3.6.1.4.1.9.9.613.0.0.2	**-snmp_trap_identification_oid 1.3.6.1.4.1.9.9.613.0.0.2**
snmp-trap-identifier-from-oid	• inOID • index	1.3.6.1.4.1.9.9.613.1.3.2.1.1 1	**-snmp_trap_identifier_from_oid_inOID 1.3.6.1.4.1.9.9.613.1.3.2.1.1** **-snmp_trap_identifier_from_oid_index 1**
create-ana-trap-event	• type • subtype[1]	EVC created trap EVC created trap **Note** These values must match the input for the **vcb event add -eventname** and **-subtype** options.	**-create_ana_trap_event_type** *"EVC created trap"* **-create_ana_trap_event_subtype** *"EVC created trap"*

1. This argument is optional.

Putting It All Together and Running the Complete Command

The complete command follows:

```
vcb eventparsingrules add -user root -password admin -enable
-rulename cevcEvcCreationNotification
-group cisco-trap-repository
-templates
snmp-trap-identification,snmp-trap-identifier-from-oid,create-managedelement-key,create
-ana-trap-event
-snmp_trap_identification_oid .1.3.6.1.4.1.9.9.613.0.0.2
```

```
-snmp_trap_identifier_from_oid_inOID .1.3.6.1.4.1.9.9.613.1.3.2.1.1
-snmp_trap_identifier_from_oid_index 1
-create_ana_trap_event_type "EVC created trap"
-create_ana_trap_event_subtype "EVC created trap"
```

- When you run the command, confirmation messages should be displayed.

To complete the configuration, you must add an event pattern for the trap.

Add an Event Pattern for the cevcEvcCreationNotification Trap

Add the event pattern as follows:

```
vcb eventpattern add -user root -password admin -rulename cevcEvcCreationNotification
-repository cisco-trap-repository -group cisco-trap-ipcore-parsing-rules
```

Where:

- rulename—Must match the rulename provided in the **vcb eventparsingrules add** command.
- repository—Match match the argument for group provided in the **vcb eventparsingrules add** command.
- group—Must be the appropriate parsing rules file for the VNE type or scheme. For more information, see Selecting Parsing Rules Files and Repositories, page 21-14.

Note For changes to take effect, you must restart the VNEs that are affected.

Adding Support for a Syslog with Event Subtypes—A Step-By-Step Example

This example shows how to investigate a syslog, FWSM-5-111004, and add support for it using the VCB.

Note In Cisco ANA 3.7.1, the FWSM-5-111004 syslog is not supported. If the syslog is supported in a later release, the principles for investigating and adding support for a syslog remain the same regardless.

See the following sections:

Investigate the FWSM-5-111004 Syslog

In this example, we discuss searching for information about a syslog using a Cisco-provided tool and searching for the message directly on Cisco.com:

- Cisco provides an Error Message Decoder tool at this URL:

  http://www.cisco.com/cgi-bin/Support/Errordecoder/index.cgi

- Cisco.com search can yield links to system log message guides, such as *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module System Log Messages, 3.2*.

In this case, information that the tool provides also matches this excerpt from the messages guide:

Error Message `%FWSM-5-111004: IP_address end configuration: {FAILED|OK}`

Explanation This message is displayed when you enter the **config floppy/memory/ network** command or the **write floppy/memory/network/standby** command. The IP_address value indicates whether the login was made at the console port or through a Telnet connection.

Recommended Action None required if the message ends with OK. If the message indicates a failure, try to fix the problem. For example, if writing to a floppy disk, ensure that the floppy disk is not write protected; if writing to a TFTP server, ensure that the server is up.

The investigation provides us with information that we need to create the event and the event parsing rules:

- event ID—%FWSM-5-111004
- unique ID—IP Address
- subtypes—OK, FAILED

Add an Event for the FWSM-5-111004 Syslog

The investigation of the syslog showed that there are two subtypes for the %FWSM-6-111004 syslog: FAILED and OK. Prepare to add a unique event name and two subtypes, one that is ticketable and one that clears the event:

- Event name—config-write-status-syslog
- Subtypes:
  - "config write failed syslog"

    severity: minor

    ticketable

    short description: "config write failed syslog"
  - "config write ok syslog"

    severity: cleared

    short description: "config write ok syslog"

Enter the following command:

```
vcb event add -user root -password admin -eventtype syslog -eventname
config-write-status-syslog -subtype1 "config write failed syslog" -ticketable1 -severity1
minor -shortdesc1 "config write failed syslog" -subtype2 "config write ok syslog"
-severity2 cleared -shortdesc2 "config write ok syslog"
```

Note The value for the **-eventtype** option must be syslog because you are adding support for a syslog.

Add Event Parsing Rules for the FWSM-5-111004 Syslog

This example shows you how to build the **vcb eventparsingrules add** command to add support for the FWSM-5-111004 syslog. This is the command syntax that you will use:

```
vcb eventparsingrules add [global args] [-enable] -rulename name -group repository-hive
-templates comma-separated-list-of-templates variable_args…
```

One step at a time, build the necessary command as follows:

- Enter the Rulename and the Group, page 21-55
- Select and Enter the Appropriate Templates for the FWSM-5-111004 Syslog, page 21-55

Enter the Rulename and the Group

To start assembling the command, enter the following:

```
vcb eventparsingrules add -user username -password password -enable
-rulename FWSM-5-111004 -repository cisco-syslog-repository
```

To provide values for these options:

- rulename—Match the error but omit the percent sign %.
- repository—Enter cisco-syslog-repository. For more information, see Selecting Parsing Rules Files and Repositories, page 21-14.

Tip Use a text editor to create the command. The advantages are that you keep a record of what you have done and can quickly correct any errors.

Select and Enter the Appropriate Templates for the FWSM-5-111004 Syslog

Next, select the templates that are appropriate for adding a trap. The information in Table 21-5 can guide you in your selection. You must select the templates that are mandatory for a syslog:

- syslog-identification
- create-ana-syslog-event
- create-managedelement-key

From the optional templates, because there are subtypes for the syslog that you are supporting, select the following template:

- syslog-subtype-from-expression

After you have selected the templates, enter them as a comma-separated list after the **-templates** option as follows:

```
-templates syslog-identification, create-ana-syslog-event, create-managedelement-key,
syslog-subtype-from-expression, syslog-subtype-from-expression
```

Enter Arguments for the Templates for the Syslog

You command so far includes rulename, group, and templates:

```
vcb eventparsingrules add -user username -password password -enable
-rulename FWSM-5-111004 -repository cisco-syslog-repository
-templates syslog-identification, create-ana-syslog-event, create-managedelement-key,
syslog-subtype-from-expression, syslog-subtype-from-expression
```

Next, you must supply any mandatory input for the templates; you should also supply any optional input that you have. View the arguments using VCB commands or elsewhere in this document as explained in Enter Arguments for the Templates for the Syslog, page 21-55.

After you list the argument names, add them to your command and supply the values that you have acquired through your research into the event. Table 21-34 provides a summary.

Table 21-34 Template Arguments for Adding Event Parsing Rules for the FWSM-5-111004 Syslog

Template	Argument	Value	Command Input
syslog-identification	expression	%FWSM-5-111004: %%uniqueid%% end configuration: %%subtypekey%% **Note** The keywords, %%uniqueid%% and %%subtypekey%%, will be replaced, respectively, by the value of the IP address and the subtype (OK or FAILED).	**-syslog_identification_expression** *"%FWSM-5-111004: %%uniqueid%% end configuration: %%subtypekey%%"*
syslog-subtype-from-expression	replacing-rules	FAILED-config write failed syslog, OK-config write ok syslog **Note** Replace input subtypes with the subtypes that you defined with the **vcb event add** command.	**-syslog_subtype_from_expression_replacing_rules—***"FAILED-config write failed syslog,OK-config write ok syslog"*
create-ana-syslog-event-type	type	config-write-status-syslog	**-create_ana_syslog_event_type** *config-write-status-syslog*

Putting It All Together and Running the Complete Command

The complete command follows:

```
vcb eventparsingrules add -user root -password admin -enable
-rulename FWSM-5-111004
-templates
syslog-identification,create-managedelement-key,create-ana-syslog-event,syslog-subtype-
from-expression
-group cisco-syslog-repository
-syslog_identification_expression "%FWSM-5-111004: %%uniqueid%% end configuration:
%%subtypekey%%"
-create_ana_syslog_event_type config-write-status-syslog
-syslog_subtype_from_expression_replacing_rules "FAILED-config write failed
syslog,OK-config write ok syslog"
```

Add an Event Pattern for the FWSM-5-111004 Syslog

Add the event pattern as follows:

```
vcb eventpattern add -user root -password admin -rulename FWSM-5-111004
-repository cisco-syslog-repository -group cisco-syslog-ipcore-parsing-rules
```

Where:

- rulename—Must match the rulename provided in the **vcb eventparsingrules add** command.

- repository—Match match the argument for group provided in the **vcb eventparsingrules add** command.
- group—Must be the appropriate parsing rules file for the VNE type or scheme. For more information, see Selecting Parsing Rules Files and Repositories, page 21-14.

Note For changes to take effect, you must restart the VNEs that are affected.

Adding Support for an Event to an NE Type—An Advanced Example

The VCB does not support adding an event to a single VNE or to all VNEs that run a particular software version. However, the VCB does support adding an event at the group level. Cisco NE types are organized into a hierarchy; NE types belong to groups of devices. NEs types in the same group use the same parsing rules files. NE types that use the same parsing rules file support the same events.

To add event recognition to an NE type, you must identify the parsing rules file that the NE type uses and then update that group; for more information, see the following procedure.

Step 1 Identify the group to which an NE type belongs. To do so, see the tables in the chapter, Overview—Supported Network Elements in Cisco ANA Version 3.7.1, in *Cisco Active Network Abstraction 3.7.1 Reference Guide*.

For example the NE type, ciscoCrs8S, is a Cisco CRS-1 8-Slot Single-Shelf System; it is grouped under Cisco CRS-1 Carrier Routing Systems.

Step 2 Identify the parsing rules file used by the group; (see Table 21-7).

For example, Table 21-7 shows that NE types in Cisco CRS-1 Carrier Routing Systems use these scheme-specific parsing rules files:

- ipcore scheme:
  - cisco-iox-syslog-ipcore-parsing-rules
  - cisco-iox-trap-ipcore-parsing-rules
- product scheme:
  - cisco-iox-syslog-product-parsing-rules
  - cisco-iox-trap-ipcore-parsing-rules

Note For information about schemes, see *Cisco Active Network Abstraction 3.7.1 Administrator Guide*.

If the NE type is in the ipcore scheme and you are adding event recognition for a trap, use the cisco-iox-trap-ipcore-parsing-rules file. Use cisco-iox-trap-ipcore-parsing-rules as input to the **-group** option for the **vcb eventpattern** command.

Step 3 Identify the repository where the parsing rules are stored; (see Table 21-7).

For example, Table 21-7 shows that Cisco CRS-1 Carrier Routing Systems uses these repository files:

- cisco-trap-repository
- mib2-trap-repository

Depending on the trap that you want Cisco ANA to recognize, select the repository file. Enter the repository filename as input to the **vcb eventpattern add -repository** command option.