Embedded Event Manager 3.2

First Published: September 29, 2009
Last Updated: June 28, 2010

This module describes how to write Embedded Event Manager (EEM) 3.2 policies using Cisco IOS command-line interface (CLI) applets and EEM policies using Tool command language (Tcl) scripts to handle Cisco IOS software faults and events.

EEM is a distributed and customized approach to event detection and recovery offered directly in a Cisco IOS device. EEM offers the ability to monitor events and take informational, corrective, or any desired action when the monitored events occur or when a threshold is reached. The EEM policy engine receives notifications when faults and other events occur. EEM policies implement recovery on the basis of the current state of the system and the actions specified in the policy for a given event. Recovery actions are triggered when the policy is run.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for Embedded Event Manager 3.2" section on page 30.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for EEM 3.2

Cisco IOS Release 12.2(52)SE and later releases.

Information About EEM 3.2

EEM 3.2 is supported in Cisco IOS Release 12.2(52)SE and later releases, and introduced the following new event detectors:

- Neighbor Discovery—Neighbor Discovery event detector provides the ability to publish a policy to respond to automatic neighbor detection when:
  - a Cisco Discovery Protocol (CDP) cache entry is added, deleted or updated.
  - a Link Layer Discovery Protocol (LLDP) cache entry is added, deleted, or updated.
  - an interface link status changes.
  - an interface line status changes.
- Identity—Identity event detector generates an event when AAA authorization and authentication is successful, when failure occurs, or after normal user traffic on the port is allowed to flow.
- Mac-Address-Table—Mac-Address-Table event detector generates an event when a MAC address is learned in the MAC address table.

Note The Mac-Address-Table event detector is supported only on switch platforms and can be used only on Layer 2 interfaces where MAC addresses are learned. Layer 3 interfaces do not learn addresses and routers do not usually support the mac-address-table infrastructure needed to notify EEM of a learned MAC address.

EEM 3.2 also introduces new CLI commands to support the applets to work with the new event detectors.

Cisco IOS CLI Commands for EEM 3.2

debug event manager

To turn on the debugging output of Embedded Event Manager (EEM) processes, use the **debug event manager** command in privileged EXEC mode. To turn off debugging output, use the **no** form of this command or the **undebug** command.

debug event manager {**action cli** | **action cns** | **action mail** | **all** | **api calls** | **api errors** | **common** | **detector all** | **detector appl** | **detector cli** | **detector config** | **detector counter** | **detector env** | **detector gold** | **detector identity** | **detector interface** | **detector ioswdsysmon** | **detector ipsla** | **detector mat** | **detector neighbor-discovery** | **detector nf** | **detector none** | **detector oir** | **detector resource** | **detector rf** | **detector routing** | **detector rpc** | **detector snmp** | **detector snmp-notification** | **detector syslog** | **detector test** | **detector timer** | **detector track** | **metricdir** | **policydir** | **server ISSU** | **server events** | **server scheduling** | **snap calls** | **snap errors** | **tcl cli_library** | **tcl commands** | **tcl smtp_library** | **xml parser**}

no debug event manager {**action cli** | **action cns** | **action mail** | **all** | **api calls** | **api errors** | **common** | **detector all** | **detector appl** | **detector cli** | **detector config** | **detector counter** | **detector env** | **detector gold** | **detector identity** | **detector interface** | **detector ioswdsysmon** | **detector ipsla** | **detector mat** | **detector neighbor-discovery** | **detector nf** | **detector none** | **detector oir** | **detector resource** | **detector rf** | **detector routing** | **detector rpc** | **detector snmp** | **detector snmp-notification** | **detector syslog** | **detector test** | **detector timer** | **detector track** | **metricdir** | **policydir** | **server ISSU** | **server events** | **server scheduling** | **snap calls** | **snap errors** | **tcl cli_library** | **tcl commands** | **tcl smtp_library** | **xml parser**}

Syntax Description

action cli	Displays debugging messages about command-line interface (CLI) event messages.
action cns	Displays debugging messages about Cisco Networking Services (CNS) event messages.
action mail	Displays debugging messages about e-mail event messages.
all	Displays all debugging messages.
api calls	Displays debugging messages about EEM client application programming interface (API) calls.
api errors	Displays debugging messages about EEM client API errors.
common	Displays common library code debugging messages.
detector all	Displays all event detector debugging messages.
detector appl	Displays debugging messages about the application-specific event detector. **Note** In Cisco IOS Release 12.4(20)T and later releases, the **application** keyword was replaced with the **appl** keyword.
detector cli	Displays debugging messages about the CLI event detector.
detector config	Displays debugging messages about the config event detector.
detector counter	Displays debugging messages about the counter event detector.
detector env	Displays debugging messages about the environmental event detector.
detector gold	Displays debugging messages about the GOLD event detector.
detector identity	Displays debugging messages about the identity event detector.
detector interface	Displays debugging messages about the interface counter event detector.
detector ioswdsysmon	Displays debugging messages about the IOS watchdog event detector.

detector ipsla	Displays debugging messages about the IP SLA event detector.
detector mat	Displays debugging messages about the mac-address-table (MAT) event detector.
detector neighbor-discovery	Displays debugging messages about the Neighbor Discovery event detector.
detector nf	Displays debugging messages about the NetFlow event detector.
detector none	Displays debugging messages about the none event detector.
detector oir	Displays debugging messages about the OIR event detector.
detector resource	Displays debugging messages about the Embedded Resource Manager (ERM) event detector.
detector rf	Displays debugging messages about the redundancy-facility (RF) event detector.
detector routing	Displays debugging messages about the routing event detector.
detector rpc	Displays debugging messages about the remote procedure call (RPC) event detector.
detector snmp	Displays debugging messages about the Simple Network Management Protocol (SNMP) event detector.
detector snmp-notification	Displays debugging messages about the SNMP notification event detector.
detector syslog	Displays debugging messages about the syslog event detector.
detector test	Displays debugging messages about the test event detector.
detector timer	Displays debugging messages about the timer event detector.
detector track	Displays debugging messages about the Enhanced Object Tracking (EOT).
metricdir	Displays debugging messages about the EEM metric event detector.
policydir	Displays debugging messages about the EEM policy director.
server ISSU	Displays debugging messages about In-Service Software Upgrade (ISSU) server events.
server events	Displays debugging messages about the EEM server events.
server scheduling	Displays all debugging messages about the EEM server scheduling events.
snap calls	Displays debugging messages about EEM SNAP client application programming interface (API) calls.
snap errors	Displays debugging messages about EEM SNAP client API errors.
tcl cli_library	Displays all debugging messages about the Tool Command Language (Tcl) command-line interface (CLI) library.
tcl commands	Displays all debugging messages about the Tcl commands.
tcl smtp_library	Displays all debugging messages about the Tcl Simple Mail Transfer Protocol (SMTP) library.
xml parser	Displays debugging messages about the EEM XML parser.

Command Modes Privileged EXEC (#)

Command History

Release	Modification
12.0(26)S	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.3(2)XE	This command was integrated into Cisco IOS Release 12.3(2)XE.
12.2(25)S	The **detector application**, **detector counter**, **detector interface**, **detector ioswdsysmon**, and **detector timer** keywords were added and this command was integrated into Cisco IOS Release 12.2(25)S.
12.3(14)T	The **action cli**, **action mail**, **detector all**, **detector cli**, **detector none**, **detector oir**, and **metricdir** keywords were added.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.4(2)T	The **detector resource**, **detector rf**, and **detector track** keywords were added.
12.2(18)SXF4	The **detector gold** keyword was added and this command was integrated into Cisco IOS Release 12.2(18)SXF4 to support Software Modularity images only.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF5	This command was integrated into Cisco IOS Release 12.2(18)SXF5.
12.4(20)T	The **common**, **detector config**, **detector env**, **detector rf**, **detector snmp-notification**, **detector test**, **server ISSU**, and **xml parser** keywords were added and the **detector application** keyword was replaced with the **detector appl** keyword.
12.4(22)T	This command was modified. The **detector ipsla**, **detector nf**, and **detector routing** keywords were added.
12.2(52)SE	This command was modified. The **detector identity**, **detector mat** and **detector neighbor-discovery** keywords were added.
12.2(54)SG	This command was integrated into Cisco IOS Release 12.2(54)SG.

Usage Guidelines

Use the **debug event manager** command to troubleshoot EEM command operations.

Caution Use any debugging command with caution because the volume of generated output can slow or stop the router operations. We recommend that this command be used only under the supervision of a Cisco engineer.

Examples

The following example turns on debugging messages about EEM server events and then configures an applet to write a message—Test message—to syslog. The debug output that follows displays the various EEM operations that occur as the applet is processed.

```
Router# debug event manager server events

Debug Embedded Event Manager server events debugging is on
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# event manager applet timer-test
Router(config-applet)# event timer countdown time 20
Router(config-applet)# action label1 syslog msg "Test message"
Router(config-applet)# end
```

```
03:46:55: fh_server: fh_io_msg: received msg 6 from client jobid 11
03:46:55: fh_server: fh_io_msg: handling event register with esid = 23
03:46:55: fh_msg_send_to_fd: receive a reply msg, minor: 5
03:46:55: fh_server: fh_io_msg: received msg 26 from client jobid 11
03:46:55: fh_msg_send_to_fd: receive a reply msg, minor: 5
03:46:55: %SYS-5-CONFIG_I: Configured from console by console
03:47:15: fd_pulse_hndlr: received a pulse from /dev/fm/fd_timer
03:47:15: fh_msg_send_to_fd: receive a reply msg, minor: 5
03:47:15: fd_pulse_hndlr: received FH_MSG_EVENT_PUBLISH
03:47:15: fh_schedule_callback: fh_schedule_callback: cc=632C0B68 prev_epc=0; epc=63A41670
03:47:15: fh_io_msg: received FH_MSG_API_INIT; jobid=13, processid=82, client=3, job
name=EEM Callback Thread
03:47:15: fh_server: fh_io_msg: received msg 10 from client jobid 13
03:47:15: %HA_EM-6-LOG: timer-test: Test message
03:47:15: fh_server: fh_io_msg: received msg 62 from client jobid 13
03:47:15: fh_schedule_callback: fh_schedule_callback: cc=632C0B68 prev_epc=63A41670; epc=0
03:47:15: fh_server: fh_io_msg: received msg 1 from client jobid 13
03:47:15: fh_io_msg: received FH_MSG_API_CLOSE client=3
```

Table 1 describes the significant fields shown in the display.

Table 1 debug event manager Field Descriptions

Field	Description
Debug Embedded Event Manager server events debugging	Indicates the type of debugging output and whether the debugging is on or off.
fh_server	Indicates a server event message.
fh_io_msg	Indicates that a message has been sent to, or received from, a client process.
fh_msg_send_to_fd	Indicates that a message has been sent to the event detector.
fd_pulse_hndlr	Indicates that a message has been received by the event detector pulse handler.

event identity

To publish an event after authentication, authorization or normal traffic has begun to flow on the interface, use the **event identity** command in applet configuration mode. To disable the publishing of events, use the **no** form of this command.

event [**tag** *event-tag*] **identity interface** {*type number* | **regexp** *interface-name*} [**maxrun** *maxruntime-number*] [**aaa-attribute** *attribute-name*] [**authc** {**all** | **fail** | **success**}] [**authz** {**all** | **fail** | **success**}] [**authc-complete**] [**mac-address** *mac-address*]

no event identity

Syntax Description

tag	(Optional) Specifies a tag using the event-tag argument that can be used with the trigger command to support multiple event statements within an applet.
event-tag	(Optional) String that identifies the tag.
interface	Specifies the interface.
type number	Interface type and number.
regexp *interface-name*	Specifies a regular expression pattern to match against interface names.
maxrun	(Optional) Specifies the maximum runtime of the applet. If the maxrun keyword is specified, the maxruntime-number value must be specified. If the maxrun keyword is not specified, the default applet run time is 20 seconds.
maxruntime-number	(Optional) Number of seconds specified in ssssssss[.mmm] format, where ssssssss must be an integer representing seconds from 0 to 31536000, and where mmm must be an integer representing milliseconds between 0 and 999.
aaa-attribute	(Optional) Specifies the regular expression pattern for AAA attributes.
attribute-name	(Optional) AAA attribute name.
authc	(Optional) Triggers events on successful, failed or both successful and failed authentication. You must specify one of the following: • **all**—Triggers an event in all cases of authentication. • **fail**—Triggers an event if authentication fails. • **success**—Triggers an event if authentication is successful.
authz	(Optional) Trigger events on successful, failed or both successful and failed authorization. You must specify one of the following: • **all**—Triggers an event in all cases of authorization. • **fail**—Triggers an event if authorization fails. • **success**—Triggers an event if authorization is successful.
authz-complete	(Optional) Triggers events once the device connected to the interface is fully authenticated, authorized and normal traffic has begun to flow on that interface.
mac-address	(Optional) Specifies the MAC address.
mac-address	(Optional) The MAC address.

Command Default By default, no events are published.

Command Modes Applet configuration (config-applet)

Command History

Release	Modification
12.2(52)SE	This command was introduced.
12.2(54)SG	This command was integrated into Cisco IOS Release 12.2(54)SG.

Usage Guidelines You must specify an interface. You can specify any or all of the other keywords. The keywords can be used in any combination.

Examples The following example shows how to publish an event when authorization is successful or failure and when the device connected to the interface is fully authenticated, authorized and normal traffic has begun to flow on that interface:

```
Router(config)# event manager applet identity
Router(config-applet)# event identity interface fastethernet0 authz all athuz-complete
Router(config-applet)#
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the Embedded Event Manager and enters applet configuration mode.

event mat

To publish an event when a mac-address is learned in the mac-address-table, use the **event mat** command in applet configuration mode. To disable the publishing of events, use the **no** form of this command.

event [**tag** *event-tag*] **mat {interface {***type number* **| regexp** *interface-name***}** [**mac-address** *mac-address*] **| mac-address** *mac-address* [**interface {***type number* **| regexp** *interface-name***}]} [maxrun** *maxruntime-number***]** [**hold-down** *seconds*] [**type** {**add** | **delete**}]

no event mat

Syntax Description

tag	(Optional) Specifies a tag using the event-tag argument that can be used with the trigger command to support multiple event statements within an applet.
event-tag	(Optional) String that identifies the tag.
interface	Specifies the interface.
type number	Interface type and number.
regexp *interface-name*	Specifies a regular expression pattern to match against interface names.
mac-address	Specifies the MAC address.
mac-address	The MAC address.
maxrun	(Optional) Specifies the maximum runtime of the applet. If the maxrun keyword is specified, the maxruntime-number value must be specified. If the maxrun keyword is not specified, the default applet run time is 20 seconds.
maxruntime-number	(Optional) Number of seconds specified in ssssssss[.mmm] format, where ssssssss must be an integer representing seconds from 0 to 31536000, and where mmm must be an integer representing milliseconds between 0 and 999.
hold-down	(Optional) Specifies the time to delay the event processing.
seconds	(Optional) Number that represents seconds and optional milliseconds in the format sssssssssss[.mmm]. The range for seconds is from 1 to 4294967295. The range for milliseconds is from 0 to 999. If using milliseconds only, specify the milliseconds in the format 0.mmm.
type	(Optional) Monitors the MAC address table events. You must specify one of the following options: • **add**—Monitors only MAC address table add events. • **delete**—Monitor only MAC address table delete events.

Command Default

By default, no events are published.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.2(52)SE	This command was introduced.
12.2(54)SG	This command was integrated into Cisco IOS Release 12.2(54)SG.

Usage Guidelines

You must specify either interface or mac-address. If one of them is specified, the other one is optional. All the keywords can be used in any combination.

Examples

The following example shows how to publish an event when a mac-address is learned in the mac-address-table:

```
Router(config)# event manager applet mat
Router(config-applet)# event mat interface fastethernet0 hold-down 34 type delete
Router(config-applet)#
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the Embedded Event Manager and enters applet configuration mode.

event neighbor-discovery

To publish an event when a Cisco Discovery Protocol (CDP) or Link Layer Discovery Protocol (LLDP) cache entry changes or a interface link status changes in an Embedded Event Manager (EEM) applet, use the **event neighbor-discovery** command in applet configuration mode. To disable the action of publishing the event, use the **no** form of this command.

event [**tag** *event-tag*] **neighbor-discovery interface** {*type number* | **regexp** *interface-name*} [**maxrun** *maxruntime-number*] *event-to-monitor*

no event neighbor-discovery

Syntax Description

tag	(Optional) Specifies a tag using the event-tag argument that can be used with the trigger command to support multiple event statements within an applet.
event-tag	(Optional) String that identifies the tag.
interface	Specifies the interface.
type number	Interface type and number.
regexp *interface-name*	Specifies a regular expression pattern to match against interface names.
maxrun	(Optional) Specifies the maximum runtime of the applet. If the maxrun keyword is specified, the maxruntime-number value must be specified. If the maxrun keyword is not specified, the default applet run time is 20 seconds.

maxruntime-number	(Optional) Number of seconds specified in ssssssss[.mmm] format, where ssssssss must be an integer representing seconds from 0 to 31536000, and where mmm must be an integer representing milliseconds between 0 and 999.
event-to-monitor	Specifies the event to be monitored on the interface. You must specify one of the following values. You can specify more than one value. • **cdp**—Triggers an event when a matching cdp event occurs. You must specify one of the following options. – **add**—Triggers events only when a new cdp cache entry is created in the cdp table. – **all**—Triggers an event when a cdp cache entry is added or deleted from the cdp cache table and when a remote cdp device sends a keepalive to update the cdp cache entry. – **delete**—Triggers events only when a cdp cache entry is deleted from the cdp table. – **update**—Triggers an event when a cdp cache entry is added to the cdp table or when the remote cdp device sends a cdp keepalive to update the cdp cache entry. • **lldp**—Triggers an event when a matching lldp event occurs. You must specify one of the following options. – **add**—Triggers events only when a new cdp cache entry is created in the cdp table. – **all**—Triggers an event when a cdp cache entry is added or deleted from the cdp cache table and when a remote cdp device sends a keepalive to update the cdp cache entry. – **delete**—Triggers events only when a cdp cache entry is deleted from the cdp table. – **update**—Triggers an event when a cdp cache entry is added to the cdp table or when the remote cdp device sends a cdp keepalive to update the cdp cache entry. • **line-event**—Triggers an event when the interface line protocol status changes. • **link-event**—Triggers an event when the interface link status changes. You must specify one of the following options. – **admindown**—Monitors link admin-down events. – **all**—Monitors all link events. – **deleted**—Monitors link deleted events. – **down**—Monitors link down events. – **goingdown**—Monitors link going-down events. – **init**—Monitors link init events. – **reset**—Monitors link reset events. – **testing**—Monitors link testing events. – **up**—Monitors link up events.

Command Default By default, no events are published.

Command Modes Applet configuration (config-applet)

Command History

Release	Modification
12.2(52)SE	This command was introduced.
12.2(54)SG	This command was integrated into Cisco IOS Release 12.2(54)SG.

Usage Guidelines You must specify interface and at least one of cdp, lldp, link-event and line-event for the event specification to be accepted. You can use interface and maxrun keywords and the event-trigger-criteria argument in any order.

Examples The following example shows how to publish an event when CDP cache entry changes:

```
Router(config)# event manager applet discovery
Router(config-applet)# event neighbhor-discovery interface fastethernet0 cdp all
Router(config-applet)#
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the Embedded Event Manager and enters applet configuration mode.

show event manager detector

To display information about Embedded Event Manager (EEM) event detectors, use the **show event manager detector** command in privileged EXEC mode.

show event manager detector [**all** | *detector-name*] [**detailed**]

Syntax Description

all	(Optional) Displays information about all available event detectors.
detector-name	(Optional) Name of event detector. The following values are valid: • **application**—Application event detector. • **cli**—Command-line interface (CLI) event detector. • **config**—Config event detector. • **counter**—Counter event detector. • **env**—Environmental event detector. • **gold**—Generic Online Diagnostic (GOLD) event detector. • **identity**—Identity event detector. • **interface**—Interface event detector. • **ioswdsysmon**—Watchdog system monitor event detector. • **ipsla**—IPSLA event detector. • **mat**—mac-address-table (MAT) event detector. • **neighbor-discovery**—Neighbor discovery event detector • **nf**—NetFlow (NF) event detector. • **none**—No event detector. • **oir**—Online insertion and removal (OIR) event detector. • **resource**—Resource event detector. • **rf**—Redundancy Framework (RF) event detector. • **rpc**—Remote Procedure Call (RPC) event detector. • **snmp**—Simple Network Management Protocol (SNMP) event detector. • **snmp-notification**—SNMP notification event detector. • **syslog**—Syslog event detector. • **test**—Test event detector. • **timer**—Timer event detector. • **track**—Track event detector.
detailed	(Optional) Displays detailed information about a specified event detector.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(20)T	This command was introduced.
12.4(22)T	This command was modified. The **ipsla**, **nf** and **routing** detectors were added.
12.2(52)SE	This command was modified. The **identity**, **mat** and **neighbor-discovery** detectors were added.
12.2(54)SG	This command was integrated into Cisco IOS Release 12.2(54)SG.

Usage Guidelines

Use the **show event manager detector** command to display information about EEM event detectors. The **all** keyword displays information about all event detectors. The **detailed** keyword displays detailed information, including:

- The event registration syntax for the Tool Command Language (Tcl) policies.
- The available array variables for the Tcl policies after event_reqinfo() is called.
- The event registration syntax for applet policies.
- The built-in variables available when an applet policy is triggered by this event detector.

Examples

The following is sample output from the **show event manager detector** command specifying the counter value:

```
Router# show event manager detector counter
No.  Name                Version   Node        Type
1    counter             01.00     node5/1     RP

Router# show event manager detector counter detailed
No.  Name                Version   Node        Type
1    counter             01.00     node5/1     RP

        Tcl Configuration Syntax:
        ::cisco::eem::event_register_counter
                 [tag <tag-val>]
                 name <counter-name>
                 entry_val <entry-val>
                 entry_op {gt | ge | eq | ne | lt | le}
                 exit_val <exit-val>
                 exit_op {gt | ge | eq | ne | lt | le}
                 [queue_priority {normal | low | high | last}]
                 [maxrun <sec.msec>] [nice {0 | 1}]

        Tcl event_reqinfo Array Names:
        event_id
        event_type
        event_type_string
        event_pub_time
        event_pub_sec
        event_pub_msec
        event_severity
        name
        value

        Applet Configuration Syntax:
        [ no ] event [tag <tag-val>] counter
                 name <counter-name>
                 entry-val <entry-val>
```

```
                entry-op {gt | ge | eq | ne | lt | le}
                exit-val <exit-val>
                exit-op {gt | ge | eq | ne | lt | le}
                [maxrun <sec.msec>]

Applet Built-in Environment Variables:
        $_event_id
        $_event_type
        $_event_type_string
        $_event_pub_time
        $_event_pub_sec
        $_event_pub_msec
        $_event_severity
        $_counter_name
        $_counter_value
```

Table 2 describes the significant fields shown in the display.

Table 2 show event manager detector Field Descriptions

Field	Description
No.	The number assigned to the event detector.
Name	Name of the event detector.
Version	Version number.
Node	Node name.
Type	Where the event detector resides.

Configuration Examples for Embedded Event Manager Applet

Example: Identity Event Detector

The following example shows how a policy named “EventIdentity” is triggered every time the authentication on the Fast Ethernet interface 0 is success.

```
event manager applet EventIdentity
 event identity interface FastEthernet0 authc success
action 1.0 syslog msg "Applet EventIdentity"
```

Example: MAT Event Detector

The following example shows how a policy named “EventMat” is triggered every time a mac-address is learned in the mac-address-table.

```
event manager applet EventMat
 event mat interface FastEthernet0
action 1.0 syslog msg "Applet EventMat"
```

Example: Neighbor-Discovery Event Detector

The following example shows how a policy named “EventNeighbor” is triggered when a Cisco Discovery Protocol (CDP) cache entry changes.

```
event manager applet EventNeighbor
 event neighbor-discovery interface FastEthernet0 cdp all
action 1.0 syslog msg "Applet EventNeighbor"
```

Event Registration Tcl Command Extensions for EEM 3.2

event_register_identity

Registers for an identity event. Use this Tcl command extension to generate an event when AAA authentication or authorization is successful or failure or after normal user traffic on the port is allowed to flow.

Syntax

```
event_register_identity [tag ?] interface ?
[aaa-attribute ?]
[authc {all | fail | success}]
[authz {all | fail | success}]
[authz-complete]
[mac-address ?]
[queue_priority {normal | low | high | last}]
[maxrun ?] [nice {0 | 1}]
```

Arguments

tag	(Optional) String identifying a tag that can be used with the trigger Tcl command extension to support multiple event statements within a Tcl script.
interface	A regular expression pattern to match against interface names.
aaa-attribute	(Optional) A regular expression that can be used to filter events by specific AAA attributes.
authc	(Optional) Triggers events on successful, failed or both successful and failed authentication.
authz	(Optional) Triggers events on successful, failed or both successful and failed authorization.
authz-complete	(Optional) Triggers events once the device connected to the interface is fully authenticated, authorized and normal traffic has begun to flow on that interface.
mac-address	(Optional) A regular expression pattern that can be used to filter events by mac addresses of the remote device.
maxrun	(Optional) Maximum run time of the script (specified in SSSSSSSSSS[.MMM] format, where SSSSSSSSSS must be an integer representing seconds between 0 and 31536000, inclusive, and where MMM must be an integer representing milliseconds between 0 and 999). If this argument is not specified, the default 20-second run-time limit is used.

queue_priority	(Optional) Priority level at which the script will be queued: • queue_priority low—Specifies that the script is to be queued at the lowest of the three priority levels. • queue_priority normal—Specifies that the script is to be queued at a priority level greater than low priority but less than high priority. • queue_priority high—Specifies that the script is to be queued at the highest of the three priority levels. • queue_priority last—Specifies that the script is to be queued at the lowest priority level. If more than one script is registered with the "queue_priority_last" argument set, these scripts will execute in the order in which the events are published. The queue_priority argument specifies the queuing priority, but not the execution priority, of the script being registered. If this argument is not specified, the default queuing priority is normal.
nice	(Optional) Policy run-time priority setting. When the nice argument is set to 1, the policy is run at a run-time priority that is less than the default priority. The default value is 0.

Result String

None

Set _cerrno

No

Event_reqinfo For EEM_EVENT_IDENTITY

```
"event_id %u event_type %u event_type_string {%s} event_pub_sec %u event_pub_msec %u
event_severity %u identity_stage %u identity_status %u interface %u identity_mac %u
identity_<attribute> {%s}"
```

Event Type	Description
event_id	Unique number that indicates the ID for this published event. Multiple policies may be run for the same event, and each policy will have the same event_id.
event_type	Type of event.
event_type_string	An ASCII string that represents the name of the event for this event type.
event_pub_sec **event_pub_msec**	The time, in seconds and milliseconds, at which the event was published to the EEM.
event_severity	The severity of the event.
identity_stage	One among authentication, autherization or autherization-complete stages.
identity_status	Success or one of these failure types: fail_authc, fail_aaa_server, fail_no_response, fail_timeout, fail_authz. For autherization-complete it is always success.

interface	The interface for the event.
identity_mac	The MAC address of the remote device for the event.
identity_<attribute>	For each AAA attribute, a set a dynamic variable to the value corresponding to that AAA attribute in the attribute or value list.

event_register_mat

Registers for a MAT event. Use this Tcl command extension to generate an event when a mac-address is learned in the mac-address-table.

Syntax

```
event_register_identity [tag ?] interface ?
[mac-address ?]
[type {add | delete}]
[hold-down ?]
[maxrun ?]
```

Arguments

tag	(Optional) String identifying a tag that can be used with the trigger Tcl command extension to support multiple event statements within a Tcl script.
interface	A regular expression pattern to match against interface names.
mac-address	Mandatory if the interface parameter is not specified. A regular expression pattern that can be used to filter events by mac addresses of the remote device.
type	(Optional) Filter based on a mac-address-table event type of add or delete. If not specified, the event type is not used in determining whether the event should be triggered.
hold-down	(Optional) When a mac-address-table event comes in, the hold-down timer can be set to make the event to wait between 1 and 4294967295 seconds before processing the policy. If not set then the policy is not delayed in being processed.
maxrun	(Optional) Maximum run time of the script (specified in SSSSSSSSSS[.MMM] format, where SSSSSSSSSS must be an integer representing seconds between 0 and 31536000, inclusive, and where MMM must be an integer representing milliseconds between 0 and 999). If this argument is not specified, the default 20-second run-time limit is used.

Result String

None

Set _cerrno

No

Event_reqinfo For EEM_EVENT_MAT

```
"event_id %u event_type %u event_type_string {%s} event_pub_sec %u event_pub_msec %u
event_severity %u notification %u intf_name %u mac_address {%s}"
```

Event Type	Description
event_id	Unique number that indicates the ID for this published event. Multiple policies may be run for the same event, and each policy will have the same event_id.
event_type	Type of event.
event_type_string	An ASCII string that represents the name of the event for this event type.

event_pub_sec **event_pub_msec**	The time, in seconds and milliseconds, at which the event was published to the EEM.
event_severity	The severity of the event.
notification	Notification type—add or delete.
intf_name	The interface name for the address table entry.
mac_address	The mac-address for the address table entry.

event_register_neighbhor_discovery

Registers for a neighbor discover event. Use this Tcl command extension to generate an event when a Cisco Discovery Protocol (CDP) or Link Layer Discovery Protocol (LLDP) cache entry or a interface link status changes.

Syntax

```
event_register_neighbhor_discovery [tag ?] interface ?
[cdp {add | update | delete | all}]
[lldp {add | update | delete | all}]
[link-event]
[line-event]
[queue_priority {normal | low | high | last}]
[maxrun ?] [nice {0 | 1}]
```

Arguments

tag	(Optional) String identifying a tag that can be used with the trigger Tcl command extension to support multiple event statements within a Tcl script.
interface	A regular expression pattern to match against interface names.
cdp	Trigger an event when a matching CDP event occurs. One of the following options should be specified. • add—Trigger events only when a new CDP cache entry is created in the CDP table. • all—Trigger an event when a CDP cache entry is added or deleted from the CDP cache table and when a remote CDP device sends a keepalive to update the CDP cache entry. • delete—trigger events only when a CDP cache entry is deleted from the CDP table. • update—trigger an event when a CDP cache entry is added to the CDP table or when the remote CDP device sends a CDP keepalive to update the CDP cache entry.
lldp	Trigger an event when a matching lldp event occurs. One of the following options should be specified. • add—Trigger events only when a new cdp cache entry is created in the cdp table. • all—Trigger an event when a cdp cache entry is added or deleted from the cdp cache table and when a remote cdp device sends a keepalive to update the cdp cache entry. • delete—trigger events only when a cdp cache entry is deleted from the cdp table. • update—trigger an event when a cdp cache entry is added to the cdp table or when the remote cdp device sends a cdp keepalive to update the cdp cache entry.
line-event	Trigger an event when the interface line protocol status changes.
link-event	Trigger an event when the interface link status changes.

queue_priority	(Optional) Priority level at which the script will be queued: • queue_priority low—Specifies that the script is to be queued at the lowest of the three priority levels. • queue_priority normal—Specifies that the script is to be queued at a priority level greater than low priority but less than high priority. • queue_priority high—Specifies that the script is to be queued at the highest of the three priority levels. • queue_priority last—Specifies that the script is to be queued at the lowest priority level. If more than one script is registered with the "queue_priority_last" argument set, these scripts will execute in the order in which the events are published. The queue_priority argument specifies the queuing priority, but not the execution priority, of the script being registered. If this argument is not specified, the default queuing priority is normal.
maxrun	(Optional) Maximum run time of the script (specified in SSSSSSSSSS[.MMM] format, where SSSSSSSSSS must be an integer representing seconds between 0 and 31536000, inclusive, and where MMM must be an integer representing milliseconds between 0 and 999). If this argument is not specified, the default 20-second run-time limit is used.
nice	(Optional) Policy run-time priority setting. When the nice argument is set to 1, the policy is run at a run-time priority that is less than the default priority. The default value is 0.

Result String

None

Set _cerrno

No

Event_reqinfo For EEM_EVENT_NEIGHBOR_DISCOVERY

```
"event_id %u event_type %u event_type_string {%s} event_pub_sec %u event_pub_msec %u
event_severity %u nd_notification {%s}"
```

Event Type	Description
event_id	Unique number that indicates the ID for this published event. Multiple policies may be run for the same event, and each policy will have the same event_id.
event_type	Type of event.
event_type_string	An ASCII string that represents the name of the event for this event type.
event_pub_sec **event_pub_msec**	The time, in seconds and milliseconds, at which the event was published to the EEM.
event_severity	The severity of the event.
Common Event_Reqinfo	
nd_notification	The type of notification—cdp-add, cdp-update, cdp-delete, lldp-add, lldp-update, lldp-delete, link, line.

nd_intf_linkstatus	The current interface link status, up or down.
nd_intf_linestatus	The current interface line status, down, goingdown, init, testing, up, reset, admindown, deleted.
nd_local_intf_name	The local interface name for the event.
nd_short_local_intf_name	The short name of the local interface for the event.
nd_port_id	The port id as identified by either the cdp or lldp protocol. This is not set for link or line protocol events.
CDP-specific Event_reqinfo	
nd_protocol	Identifies which protocol triggered the event, for CDP it will always be set to cdp.
nd_proto_notif	Identifies which type of protocol event triggered the event, add, update or delete.
nd_proto_new_entry	If set to 1, the event was triggered because the cache entry is new, otherwise it will be set to 0.
nd_cdp_entry_name	The name of the cdp cache entry in the cdp table.
nd_cdp_hold_time	The time remaining until the cdp cache entry expires and is deleted from the cdp table. This time will be reset to some maximum by an update from the cdp neighbor. It is usually set to 0 for new entries.
nd_cdp_mgmt_domain	The CDP VTP management domain.
nd_cdp_platform	The platform name reported by the remote device.
nd_cdp_version	The version of code running on the remote device.
nd_cdp_capabilities_string	The contents of the CDP capabilities field in a string format: Router, Trans-Bridge, Source-Route-Bridge, Switch, Host, IGMP, Repeater, Phone, Remotely-Managed device, CVTA phone port, Two-port Mac Relay or any combination of these separated by commas.
nd_cdp_capabilities_bits	The CDP capabilities bits in a hexadecimal number preceded with 0x.
nd_cdp_capabilities_bit_[0-31]	A series of values that will be set to YES if that bit in the capabilities field is set or NO if it is not set.
LLDP-specific Event_reqinfo	
nd_protocol	Identifies which protocol triggered the event, for LLDP it will always be set to lldp.
nd_proto_notif	Identifies which type of protocol event triggered the event, add, update or delete.
nd_proto_new_entry	If set to 1, the event was triggered because the cache entry is new, otherwise it will be set to 0.
nd_lldp_chassis_id	The chassis id field from the LLDP cache entry.
nd_lldp_system_name	The system name from the LLDP cache entry.
nd_lldp_system_description	The system description field from the LLDP cache entry.
nd_lldp_ttl	The LLDP time to live field from the LLDP cache entry.
nd_lldp_port_description	The port description field from the LLDP cache entry.

nd_lldp_system_capabilities_string	The LLDP system capabilities field from the LLDP cache entry. Provided as a string that can contain O, P, B, W, R, T, C, S or any combination of these separated by commas.
nd_lldp_enabled_capabilities_string	The LLDP enabled system capabilities field from the LLDP cache entry. Provided as a string that can contain O, P, B, W, R, T, C, S or any combination of these separated by commas.
nd_lldp_system_capabilities_bits	The LLDP system capabilities bits field from the LLDP cache entry. Provided as a hexadecimal number preceded by 0x.
nd_lldp_enabled_capabilities_bits	The LLDP enabled capabilities bits field from the LLDP cache entry. Provided as a hexadecimal number preceded by 0x.
nd_lldp_capabilities_bits	The LLDP capabilities bits field from the LLDP cache entry. Provided as a hexadecimal number preceded by 0x.
nd_lldp_capabilities_bit_[0-31]	A series of values that will be set to YES if that bit in the capabilities field is set or NO if it is not set.

Where to Go Next

- If you want to get information about EEM overview, see the "Embedded Event Manager Overview" module.
- If you want to write EEM policies using the Cisco IOS CLI, see the "Writing Embedded Event Manager Policies Using the Cisco IOS CLI" module.
- If you want to write EEM policies using Tcl, see the "Writing Embedded Event Manager Policies Using Tcl" module.

Additional References

Related Documents

Related Topic	Document Title
EEM Overview	*Embedded Event Manager Overview*
Writing EEM policies using IOS CLI	*Writing Embedded Event Manager Policies Using the Cisco IOS CLI*
Writing EEM policies using Tcl	*Writing Embedded Event Manager Policies Using Tcl*
Network Management commands (including EEM commands): complete command syntax, defaults, command mode, command history, usage guidelines, and examples	*Cisco IOS Network Management Command Reference*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Embedded Event Manager 3.2

Table 3 lists the release history for this feature.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 3 lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 3 Feature Information for EEM 3.2

Feature Name	Releases	Feature Information
Embedded Event Manager 3.2	12.2(52)SE	EEM is a distributed and customized approach to event detection and recovery offered directly in a Cisco IOS device. The following sections provide information about this feature: • Information About EEM 3.2, page 2 • Cisco IOS CLI Commands for EEM 3.2, page 3 • Configuration Examples for Embedded Event Manager Applet, page 17 • Event Registration Tcl Command Extensions for EEM 3.2, page 19 The following commands were introduced or modified: **debug event manager**, **event identity**, **event mat**, **event neighbor-discovery**, **show event manager detector**.