# event mat through R Commands


- event mat through R Commands, page 1


# event mat through R Commands

# 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 sssssssss[.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:<br>• **add** --Monitors only MAC address table add events.<br>• **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.<br>• **deleted**--Monitors link deleted events.<br>• **down**--Monitors link down events.<br>• **goingdown**--Monitors link going-down events.<br>• **init**--Monitors link init events.<br>• **reset**--Monitors link reset events.<br>• **testing**--Monitors link testing events.<br>• **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. |

# event nf

To publish an event when a NetFlow operation is triggered in an Embedded Event Manager (EEM) applet, use the **event nf** command in applet configuration mode. To disable the action of publishing an event when NetFlow operations are triggered, use the **no** form of this command.

**event** [**tag** *event-tag*] **nf monitor-name** *name* **event-type** {**create**| **delete**| **update**} **[exit-event-type]** {**create**| **delete**| **update**} *subevent* **field** *field-type* **entry-value** *value-string* [**exit-value** *value-string*] **entry-op** *operator-value* [**exit-op** *operator-value*] [**rate-interval** *seconds*] [**exit-rate-interval** *seconds*] [**maxrun** *maxruntime-number*]

**no event** [**tag** *event-tag*] **nf**

## 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. |
| **monitor-name** *name* | Specifies the name of the NetFlow monitor. |
| **event-type** | Specifies the type of event to monitor, cache or field. |
| **create** | Creates a NetFlow event. |
| **delete** | Deletes a NetFlow event. |
| **update** | Updates a NetFlow event. |
| **exit-event-type** | The event-type (create, delete, update) at which the event will be rearmed to be monitored again. |
| *subevent* | Specifies the event and its attributes to monitor. Valid values are **event1**, **event2**, **event3**, **event4**.<br>**Note** The subevent keywords can be used alone, together, or in any combination with each other, but each keyword can be used only once. |

| | |
|---|---|
| **field** *field-type* | Specifies the cache or field attribute to be monitored. One of the following attributes can be specified:<br>• **counter** {**bytes** \| **packets**}--Specifies the counter fields.<br>• **datalink** {**dot1q** \| **mac**}--Specifies the datalink (layer2) fields.<br>• **flow** {**direction** \| **sampler**}--Specifies the flow identifying fields.<br>• **interface** {**input** \| **output**}--Specifies the interface fields.<br>• **ipv4** *field-type*-- Specifies the IPv4 fields.<br>• **ipv6** *field-type*-- IPv6 fields<br>• **routing** *routing-attrribute* **--** Specifies the routing attributes.<br>• **timestamp sysuptime** {**first \| last**}**--**Specifies the timestamp fields.<br>• **transport** *field-type*-- Specifies the Transport layer fields.<br>For more information, use the question mark (?) online help function. |
| **entry-value** *value-string* | Specifies the entry value to be compared. |
| **exit-value** *string* | (Optional) Specifies the value at which the event is set to be monitored again. |
| **rate-interval** *sec* | Specifies the rate interval value in seconds. The valid range is from 1 to 4294967295. |
| **exit-rate-interval** *sec* | (Optional) Specifies the interval value for cache rate and cache entry. The valid range is from 0 to 4294967295. |
| **entry-op** | Specifies the operator used to compare the collected usage sample with the specified value. The valid values are: |

| | |
|---|---|
| *operator-value* | The comparison operator. Valid values are:<br>• **eq** - Equal to<br>• **ge** - Greater than or equal to<br>• **gt** - Greater than<br>• **le** - Less than or equal to<br>• **lt** - Less than<br>• **wc** - Wildcard |
| **exit-op** | (Optional) The operator used to compare the current event attribute value with the exit value. |
| **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. |

## Command Default

By default, no events are published when NetFlow operations are triggered.

## Command Modes

Applet configuration (config-applet)

## Command History

| Release | Modification |
|---|---|
| 12.4(22)T | This command was introduced. |
| 12.2(33)SRE | This command was integrated into Cisco IOS Release 12.2(33)SRE. |

## Usage Guidelines

You can use the **event nf**command to monitor the NetFlow events. Multiple events can be specified together for additional filtering on more than one event.

## Examples

The following example how to configure an applet to monitor NetFlow events:

```
Router(config)# event manager applet EventNF
```

```
Router(config-applet)# event nf event-type create monitor-name mon1 event1 entry-op eq
entry-val val1 field counter bytes long rate-interval 12
Router(config-applet)#
```

## Related Commands

| Command | Description |
|---|---|
| **event manager applet** | Registers an event applet with the EEM and enters applet configuration mode. |

# event none

To specify that an Embedded Event Manager (EEM) policy is to be registered with the EEM and can be run manually, use the **event none** command in applet configuration mode. To remove the **event none** command from the configuration file, use the **no** form of this command.

**event** [**tag** *event-tag*] **none** [**sync** {**yes**| **no**}] **[default]** [**maxrun** *maxruntime-number*]

**no event none**

## 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. |
| **synch** | Indicates whether the policy should be executed synchronously before the CLI command executes.<br>• If the yes keyword is specified, the policy will run synchronously with the CLI command.<br>• If the no keyword is specified, the policy will run asynchronously with the CLI command. |
| **default** | (Optional) The time period during which the CLI event detector waits for the policy to exit (specified in sssssssss[.mmm] format, where sssssssss must be an integer representing seconds from 0 to 4294967295, and where mmm must be an integer representing milliseconds from 0 to 999). If the default time period expires before the policy exits, the default action will be executed. The default action is to run the command. If this argument is not specified, the default time period is set to 30 seconds. |
| **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 between 0 and 31536000, inclusive, and where mmm must be an integer representing milliseconds between 0 and 999). |

## Command Default

No EEM events are triggered on the basis of Cisco IOS system monitor counters.

## Command Modes

Applet configuration (config-applet).

## Command History

| Release | Modification |
|---|---|
| 12.3(14)T | This command was introduced. |
| 12.2(28)SB | This command was integrated into Cisco IOS Release 12.2(28)SB. |
| 12.2(18)SXF4 | 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. |
| 12.2SX | This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware. |
| 12.4(20)T | The tag and maxrun keywords were added to support multiple event statements within an applet. |

## Usage Guidelines

EEM usually schedules and runs policies on the basis of an event specification that is contained within the policy itself. The **event none** command allows EEM to identify an EEM policy that can either be run manually or be run when an EEM applet is triggered. To run the policy, use either the **action policy** command in applet configuration mode or the **event manager run** command in global configuration mode.

## Examples

The following example shows how to register a policy named manual-policy to be run manually and then how to execute the policy:

```
Router(config)# event manager applet manual-policy
Router(config-applet)# event none
Router(config-applet)# exit
Router(config)# event manager run manual-policy
```

## Related Commands

| Command | Description |
|---|---|
| **action policy** | Registers an EEM policy with EEM. |
| **event manager applet** | Registers an EEM applet with EEM and enters applet configuration mode. |

| Command | Description |
|---|---|
| **event manager run** | Manually runs a registered EEM policy. |
| **show event manager policy registered** | Displays registered EEM policies. |

# event routing

To publish an event when route entries change in Routing Information Base (RIB) infrastructure, use the **event routing** command in applet configuration mode. To stop publishing events when route entries change in RIB, use the **no** form of this command.

**event**[**tag** *event-tag*]**routing network** *ip-address*/*length*[**ge** *ge-length*][**le** *le-length*][**protocol** *protocol-value*][**type**{**add**| **all**| **modify**| **remove**}][**maxrun** *maxruntime-number*]

**no event** [**tag** *event-tag*] **routing**

## 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. |
| **network** | Specifies the network ip address and length, whose route is to be monitored. |
| *ip-address* / *length* | The ip address and length of the network to be monitored. For example, 192.0.2.4/8. |
| **ge** *ge-length* | (Optional) Specifies the minimum prefix length to be matched. |
| **le** *le-length* | (Optional) Specifies the maximum prefix length to be matched. |
| **ne** *ne-length* | (Optional) Specifies the prefix length not to be matched. |
| **protocol** | (Optional) Specifies the protocol value for the network being monitored. |
| *protocol-value* | The network protocol value. One of the following protocols can be used: **all**, **bgp**, **connected**, **eigrp**, **isis**, **iso-igrp**, **mobile**, **odr**, **ospf**, **rip**, and **static**. The default is **all**. |
| **type** | (Optional) Specifies the desired policy trigger. The default is **all**. |
| **add** | Specifies that an entry is added to the routing table. |
| **all** | Specifies that a routing table entry is added, removed, or modified. |

| | |
|---|---|
| **modify** | Specifies that an entry in the routing table is modified. |
| **remove** | Specifies that an entry is removed from the routing table |
| **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, inclusive, and where mmm must be an integer representing milliseconds between 0 and 999. |

## Command Default

By default, no events are published when route entries change in RIB infrastructure.

## Command Modes

Applet configuration (config-applet)

## Command History

| Release | Modification |
|---|---|
| 12.4(22)T | This command was introduced. |
| 12.2(33)SRE | This command was integrated into Cisco IOS Release 12.2(33)SRE. |

## Usage Guidelines

An EEM event is published when route-entry changes are detected in a RIB infrastructure. The network IP address for the route to be monitored must be specified. Network prefixes to be matched, protocol values, and type are optional parameters.

**Note** Modification of an existing static route may result in a remove event followed by an add event for the old API (v1.0) or a modify event for the new API (v2.0) depending on the Cisco IOS release.

## Examples

The following example shows how a specific route entries change when many parameters is monitored:

```
Router(config
)# event manager applet EventRouting
Router(config-applet)# event routing 192.0.2.4/8 protocol static type add ge 5 maxrun 56
Router(config-applet)#
```

The following example shows the output for the Cisco IOS version that uses the old routing API (v1.0):

```
Router# show event manager detector routing
No. Name     Version     Node        Type
1  routing     01.00     node0/0     RP
```

The following example shows the output for the Cisco IOS version that uses the new routing API (v2.0):

```
Router# show event manager detector routing
No. Name     Version     Node        Type
1  routing     02.00     node0/0     RP
```

## Related Commands

| Command | Description |
|---|---|
| **event manager applet** | Registers an event applet with the EEM and enters applet configuration mode. |

# event snmp

To specify the event criteria for an Embedded Event Manager (EEM) applet that is run by sampling Simple Network Management Protocol (SNMP) object identifier values, use the **event snmp**command in applet configuration mode. To remove the SNMP event criteria, use the **no** form of this command.

**event** [**tag** *event-tag*] **snmp oid** *oid-value* **get-type** {**exact**| **next**} **entry-op** *operator* **entry-val** *entry-value* **entry-type** {**value**| **increment**| **rate**} [**exit-comb** {**or**| **and**}] [**exit-op** *operator*] [**exit-val** *exit-value*] [**exit-type** {**value**| **increment**| **rate**}] [**exit-time** *exit-time-value*] [**exit-event** {**true**| **false**}] [**average-factor** *average-factor-value*] **poll-interval** *poll-int-value* [**maxrun** *maxruntime-number*]

**no event** [**tag** *event-tag*] **snmp oid** *oid-value* **get-type** {**exact**| **next**} **entry-op** *operator* **entry-val** *entry-value* **entry-type** {**value**| **increment**| **rate**} [**exit-comb** {**or**| **and**}] [**exit-op** *operator*] [**exit-val** *exit-value*] [**exit-type** {**value**| **increment**| **rate**}] [**exit-time** *exit-time-value*] [**exit-event** {**true**| **false**}] [**average-factor** *average-factor-value*] **poll-interval** *poll-int-value* [**maxrun** *maxruntime-number*]

## 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. |
| **oid** | Specifies the SNMP object identifier (object ID) value in the *oid-value* argument as the event criteria. |
| *oid-value* | Object ID value of the data element, in SNMP dotted notation. An OID is defined as a type in the associated MIB, CISCO-EMBEDDED-EVENT-MGR-MIB, and each type has an object value. Monitoring of some OID types is supported. When the **oid** keyword is used, an error message is returned if the OID is not one of the following:<br>• INTEGER_TYPE<br>• COUNTER_TYPE<br>• GAUGE_TYPE<br>• TIME_TICKS_TYPE<br>• COUNTER_64_TYPE<br>• OCTET_PRIM_TYPE<br>• OPAQUE_PRIM_TYPE |
| **get-type** | Specifies the type of SNMP get operation to be applied to the object ID specified by the *oid-value* argument. |

| | |
|---|---|
| **exact** | Retrieves the object ID specified by the *oid-value* argument. |
| **next** | Retrieves the object ID that is the alphanumeric successor to the object ID specified by the *oid-value* argument. |
| **entry-op** | Compares the contents of the current object ID with the entry value using the specified operator. If there is a match, an event is triggered and event monitoring is disabled until the exit criteria are met. |
| *operator* | Two-character string. The *operator*argument takes one of the following values:<br>• **gt** --Greater than.<br>• **ge** --Greater than or equal to.<br>• **eq** --Equal to.<br>• **ne** --Not equal to.<br>• **lt** --Less than.<br>• **le** --Less than or equal to. |
| **entry-val** | Specifies the value with which the contents of the current object ID are compared to decide if an SNMP event should be raised. |
| *entry-value* | Entry object ID value of the data element. |
| **entry-type** | Specifies a type of operation to be applied to the object ID specified by the *entry-value* argument. |
| **value** | Value is defined as the actual value of the *entry-value* or *exit-value*argument. |
| **increment** | Increment uses the *entry-value* or *exit-value*field as an incremental difference and the *entry-value*or *exit-value*is compared with the difference between the current counter value and the value when the event was last triggered (or the first polled sample if this is a new event). A negative value checks the incremental difference for a counter that is decreasing. |

| | |
|---|---|
| **rate** | Rate is defined as the average rate of change over a period of time. The time period is the *average-factor-value* multiplied by the *poll-int-value*. At each poll interval the difference between the current sample and the previous sample is taken and recorded as an absolute value. An average of the previous *average-factor-value* samples is taken to be the rate of change. |
| **exit-comb** | (Optional) Indicates the combination of exit conditions that must be met before event monitoring is reenabled. |
| **or** | (Optional) Specifies that an exit comparison operator and an exit object ID value or an exit time value must exist. |
| **and** | (Optional) Specifies that an exit comparison operator, an exit object ID value, and an exit time value must exist. |
| **exit-op** | (Optional) Compares the contents of the current object ID with the exit value using the specified operator. If there is a match, an event is triggered and event monitoring is reenabled. |
| **exit-val** | (Optional) Specifies the value with which the contents of the current object ID are compared to decide whether the exit criteria are met. |
| *exit-value* | (Optional) Exit object ID value of the data element. |
| **exit-type** | (Optional) Specifies a type of operation to be applied to the object ID specified by the *exit-value* argument. If not specified, the value is assumed. |
| **exit-time** | (Optional) Specifies the time period after which the event monitoring is reenabled. The timing starts after the event is triggered. |
| *exit-time-value* | (Optional) Number that represents seconds and optional milliseconds in the format ssssss[.mmm]. The range for seconds is from 0 to 4294967295. The range for milliseconds is from 0 to 999. If only milliseconds are used, the format is 0.mmm. |
| **exit-event** | (Optional) Indicates whether a separate exit event is to be triggered when event monitoring is enabled after an initial event is triggered. |

| | |
|---|---|
| **true** | (Optional) Specifies that a separate exit event is triggered. |
| **false** | (Optional) Specifies that a separate exit event is not triggered. This is the default. |
| **average-factor** | (Optional) Specifies a number used to calculate the period used for rate-based calculations. The *average-factor-value* is multiplied by the *poll-int-value* to derive the period in milliseconds. |
| *average-factor-value* | (Optional) Number in the range from 1 to 64. The minimum average factor value is 1. |
| **poll-interval** | Specifies the time interval between consecutive polls. |
| *poll-int-value* | Number that represents seconds and optional milliseconds in the format ssssss[.mmm]. The range for seconds is from 1 to 4294967295. The range for milliseconds is from 0 to 999. The minimum polling interval is 1 second. |
| **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 SSSSSSSSSS[.MMM] format, where SSSSSSSSSS must be an integer representing seconds between 0 and 4294967295, inclusive, and where MMM must be an integer representing milliseconds between 0 and 999. |

**Command Default**

No EEM events are triggered on the basis of SNMP object identifier values.

**Command Modes**

Applet configuration (config-applet)

**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. |

| Release | Modification |
|---|---|
| 12.2(25)S | This command was integrated into Cisco IOS Release 12.2(25)S. |
| 12.3(14)T | Optional keywords to support SNMP rate-based events were added. |
| 12.2(28)SB | This command was integrated into Cisco IOS Release 12.2(28)SB. |
| 12.2(18)SXF4 | 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. |
| 12.2SX | This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware. |
| 12.4(20)T | The **tag** and **maxrun**keywords and associated arguments were added. |

## Usage Guidelines

An EEM event is triggered when one of the fields specified by an SNMP object ID crosses a defined threshold. If multiple conditions exist, the SNMP event will be triggered when all the conditions are met.

Exit criteria are optional. If exit criteria are not specified, event monitoring will be reenabled immediately. If exit criteria are specified--on the basis of values or time periods--event monitoring is not reenabled until the criteria are met.

When the **entry-op** keyword is used and there is a match, an event is triggered and event monitoring is disabled until the exit criteria are met.

When the **exit-op** keyword is used and there is a match, an event is triggered and event monitoring is reenabled.

The **entry-type**keyword triggers one of the following actions:

- If the **value** keyword is specified, the *entry-value* is an actual value and an SNMP event is raised whenever the absolute value occurs.
- If the **increment** keyword is specified, the *entry-value* is an increment and an SNMP event is raised whenever the incremental value is reached.
- If the **rate** keyword is specified, the *entry-value* is a rate of change and an SNMP event is raised whenever the rate of change value is reached.

When the optional **exit-type**keyword is used, the following occurs:

- If the **value** keyword is specified, the *exit-value* is an actual value and the event monitoring is reenabled whenever the absolute value occurs. This is the default.
- If the **increment** keyword is specified, the *exit-value* is an increment and the event monitoring is reenabled whenever the incremental value is reached.
- If the **rate** keyword is specified, the *exit-value* is a rate of change and the event monitoring is reenabled whenever the rate of change value is reached.

The increment and rate types are supported only for the following OID types: INTEGER_TYPE, COUNTER_TYPE, and COUNTER_64_TYPE.

## Examples

The following example shows how an EEM applet called memory-fail will run when there is an exact match on the value of a specified SNMP object ID that represents the amount of current process memory. A message saying that process memory is exhausted and noting the current available memory will be sent to syslog.

```
Router(config)# event manager applet memory-fail
Router(config-applet)# event snmp oid 1.3.6.1.4.1.9.9.48.1.1.1.6.1 get-type exact entry-op
 lt entry-val 5120000 poll-interval 10
Router(config-applet)# action 1.0 syslog msg "Memory exhausted; current available memory
is $_snmp_oid_val bytes"
```

The following example shows an EEM applet called IPSLAping1 being registered to run when there is an exact match on the value of a specified SNMP object ID that represents a successful IP SLA ICMP echo operation (this is equivalent to a **ping** command). Four actions are triggered when the echo operation fails, and event monitoring is disabled until after the second failure.

A message saying that the ICMP echo operation to a server failed is sent to syslog, an SNMP trap is generated, EEM publishes an application-specific event, and a counter called IPSLA1F is incremented by a value of one.

```
Router(config)# event manager applet IPSLAping1
Router(config-applet)# event snmp oid 1.3.6.1.4.1.9.9.42.1.2.9.1.6.4 get-type exact
entry-op eq entry-val 1 exit-op eq exit-val 2 poll-interval 5
Router(config-applet)# action 1.0 syslog priority critical msg "Server IP echo failed:
OID=$_snmp_oid_val"
Router(config-applet)# action 1.1 snmp-trap strdata "EEM detected server reachability
failure to 10.1.88.9"
Router(config-applet)# action 1.2 publish-event sub-system 88000101 type 1 arg1 10.1.88.9
arg2 IPSLAEcho arg3 fail
```

## Related Commands

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

# event snmp-notification

To register the event criteria for an Embedded Event Manager (EEM) applet that is run by sampling Simple Network Management Protocol (SNMP) notification, use the **event snmp-notification**command in applet configuration mode. To remove the SNMP notification event criteria, use the **no** form of this command.

**event** [**tag** *event-tag*] **snmp-notification oid** *oid-string* **oid-val** *comparison-value* **op** *operator* [**maxrun** *maxruntime-number*] [**src-ip-address** *ip-address*] [**dest-ip-address** *ip-address*] [**default** *seconds*] [**direction** {**incoming**| **outgoing**}] [**msg-op** {**drop**| **send**}]

**no event** [**tag** *event-tag*] **snmp-notification**

## 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. |
| **oid** | Specifies the SNMP object identifier (object ID) values in the *oid-val* argument as the event criteria. |
| *oid-string* | Object ID value of the data element, in SNMP dotted notation. An OID is defined as a type in the associated MIB, CISCO-EMBEDDED-EVENT-MGR-MIB, and each type has an object value. Monitoring of some OID types is supported. When the **oid** keyword is used, an error message is returned if the OID is not one of the following:<br>• COUNTER_TYPE<br>• COUNTER_64_TYPE<br>• GAUGE_TYPE<br>• INTEGER_TYPE<br>• OCTET_PRIM_TYPE<br>• OPAQUE_PRIM_TYPE<br>• TIME_TICKS_TYPE |
| **oid-val** *comparison-value* | Specifies the OID comparison value. |
| **op** | Compares the contents of the current object ID with the SNMP Protocol Data Unit (PDU) entry value using the specified operator. If there is a match, an event is triggered and event monitoring is disabled until the exit criteria are met. |

| | |
|---|---|
| *operator* | Two-character string. The *operator*argument takes one of the following values:<br>• **gt** --Greater than.<br>• **ge** --Greater than or equal to.<br>• **eq** --Equal to.<br>• **ne** --Not equal to.<br>• **lt** --Less than.<br>• **le** --Less than or equal to. |
| **maxrun** | (Optional) Specifies the maximum runtime of the applet. 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 between 0 and 31536000, inclusive, and where mmm must be an integer representing milliseconds between 0 and 999. The default value is 20 seconds. |
| **src-ip-address** | (Optional) Specifies the source IP address where the SNMP notification trap originates. The default is all; it is set to receive SNMP notification traps from all IP addresses. |
| *ip-address* | (Optional) The source IP address. |
| **dest-ip-address** | (Optional) Specifies the destination IP address where the SNMP notifications trap is sent. The default is all; it is set to receive SNMP traps from all destination IP addresses. |
| *dest-ip-address* | (Optional) The destination IP address. |
| **default** *seconds* | (Optional) Specifies the time period during which the snmp notification event detector waits for the policy to exit. The time period is specified in sssssssss[.mmm] format, where sssssssss must be an integer representing seconds between 0 and 4294967295 and mmm must be an integer representing milliseconds between 0 and 999. |

| | |
|---|---|
| **direction** | (Optional) Determines the direction of the SNMP trap or inform PDU to filter. The default is incoming.<br>**incoming** --Specifies the incoming direction of the SNMP trap or inform PDU to filter.<br>**outgoing** --Specifies the outgoing direction of the SNMP trap or inform PDU to filter. |
| **msg-op** | (Optional) Indicates the action to be taken on the SNMP PDU, drop it or send it once the event is triggered.<br>**drop** --Specifies to drop the messages.<br>**send** --Specifies to send the messages. |

**Command Default**

No EEM events are triggered on the basis of SNMP notification object identifier values.

**Command Modes**

Applet configuration (config-applet)

**Command History**

| Release | Modification |
|---|---|
| 12.4(20)T | This command was introduced. |
| 15.0(1)M | This command was modified. The following keywords and arguments were added: **default**, *seconds*, **direction**, **incoming**, **outgoing**, **msg-op**, **drop**, and **send**. |

**Usage Guidelines**

The SNMP notification event detector provides the ability to intercept SNMP trap and inform messages coming into the router. An SNMP notification event is generated when an incoming SNMP trap or inform message matches specified values or crosses specified thresholds.

The SNMP and the SNMP server manager must be configured and enabled prior to the use of the snmp-notification event detector.

An EEM event is triggered when one of the fields specified by an SNMP notification object ID crosses a defined threshold. If multiple conditions exist, the SNMP notification event is triggered when all the conditions are met.

An OID is defined as a type in the associated MIB, CISCO-EMBEDDED-EVENT-MGR-MIB, and each type has an object value. Monitoring of some OID types is supported. When the **oid** keyword is used, an error message is returned if the OID is not one of the following:

- INTEGER_TYPE
- COUNTER_TYPE
- GAUGE_TYPE

- TIME_TICKS_TYPE
- COUNTER_64_TYPE
- OCTET_PRIM_TYPE
- OPAQUE_PRIM_TYPE

When the **op** keyword is used and there is a match, an event is triggered and event monitoring is disabled until the exit criteria are met.

The *operator*argument takes one of the following values:

- **gt** --Greater than.
- **ge** --Greater than or equal to.
- **eq** --Equal to.
- **ne** --Not equal to.
- **lt** --Less than.
- **le** --Less than or equal to.

## Examples

The following example shows how to configure the **snmp-server community** public RW and **snmp-server manager** commands before **event snmp-notification** is configured.

```
Router(config)# snmp-server community public RW
Router(config)# snmp-server manager
```

The following example shows how an EEM applet called SNMP_Notification is being registered to run an EEM script when the router receives an SNMP notification on destination IP address 192.168.1.1 for object OID 1 whose value equals 10.

```
Router(config)# event manager applet SNMP_Notification
Router(config-applet)# event snmp-notification dest-ip-address 192.168.1.1 oid 1 op eq
oid-val 10
Router(config-applet)# action 1 policy eem_script
```

The following example shows how to intercept an outgoing SNMP trap with the OID 1.3.6.1.4.1.318.2.3.3 and OID value of "UPS: Returned from battery backup power", drop the message and send out a different one.

```
Router(config)# event manager applet SNMP_Notification
Router(config-applet)# event snmp-notification dest_ip_address 192.168.1.1 oid
1.3.6.1.4.1.318.2.3.3 op eq oid-value "UPS: Returned from battery backup power" direction
outgoing msg-op drop
```

## Related Commands

| Command | Description |
|---|---|
| **event manager applet** | Registers an event applet with the EEM and enters applet configuration mode. |

# event snmp-object

To register the Simple Network Management Protocol (SNMP) object event for an Embedded Event Manager (EEM) applet that is run by sampling the SNMP object, use the **event snmp-object** command in applet configuration mode. To remove the SNMP object event criteria, use the **no** form of this command.

**event snmp-object oid** *oid-value* **type** *value* **sync** {**yes**| **no**} **skip** {**yes**| **no**} **istable** {**yes**| **no**} [**default** *seconds*] [**maxrun** *maxruntime-number*]

**no event snmp-object**

## Syntax Description

| | |
|---|---|
| **oid** | Specifies the SNMP object identifier (object ID). |
| *oid-value* | Object ID value of the data element in SNMP dotted notation. An OID is defined as a type in the associated MIB, CISCO-EMBEDDED-EVENT-MGR-MIB, and each type has an object value. |
| **type** *value* | Specifies the type of object. The following values are valid:<br>• **counter** --A 32-bit number with a minimum value of 0. When the maximum value is reached, the counter resets to 0.<br>• **counter64** --A 64-bit number with a minimum value of 0. When the maximum value is reached, the counter resets to 0.<br>• **gauge** --A 32-bit number with a minimum value of 0. For example, the interface speed on a router is measured using a gauge object type.<br>• **int** --A 32-bit number used to specify a numbered type within the context of a managed object. For example, to set the operational status of a router interface, 1 represents up and 2 represents down.<br>• **ipv4** --IP version 4 address.<br>• **octet** --An octet string in hex notation used to represent physical addresses.<br>• **oid** --Object identifier value.<br>• **string** --An octet string in text notation used to represent text strings.<br>• **uint** --A 32-bit number used to represent decimal value. |

| | |
|---|---|
| **sync** | Specifies the SNMP and EEM policy execution.<br>• **no** --Policy and SNMP will run asynchronously.<br>• **yes** --Run policy and the result determines whether to run SNMP request. |
| **skip** | Mandatory if **sync** is set to **no** and should not be used if **sync** is **yes**. Specifies whether to skip CLI command execution.<br>• **no** --CLI command should be executed.<br>• **yes** --CLI command should not be executed. |
| **istable** | (Optional) Specifies whether the OID is a SNMP table.<br>• **yes** --OID is an SNMP table.<br>• **no** --IOD is not an SNMP table. |
| **default** | (Optional) The time period during which the SNMP Object event detector waits for the policy to exit. |
| *seconds* | (Optional) Number that represents seconds and optional milliseconds in the format sssssssss[.mmm]. The range for seconds is from 0 to 4294967295. The range for milliseconds is from 0 to 999. If using milliseconds only, specify the milliseconds in the format 0.mmm. |
| **maxrun** | (Optional) Specifies the maximum runtime of the applet. |
| *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. The default value is 20 seconds. |

## Command Modes

Applet configuration (config-applet)

## Command History

| Release | Modification |
|---|---|
| 15.0(1)M | This command was introduced. |

| Release | Modification |
|---|---|
| 15.0(1)M1 | This command was modified. The **counter64** and **oid** values were added to the **type** keyword. |

## Usage Guidelines

Use the **event snmp-object** command to register the SNMP object event for an EEM applet that is run by sampling SNMP object.

## Examples

The following example shows how to use the **event snmp-object** command:

```
Router(config)# event manager applet test
Router(config-applet)# event snmp-object
```

## Related Commands

| Command | Description |
|---|---|
| **action syslog** | Specifies the action of writing a message to syslog when an EEM applet is triggered. |
| **event manager applet** | Registers an event applet with the EEM and enters applet configuration mode. |

# event track

To specify the event criteria for an Embedded Event Manager (EEM) applet that is run on the basis of a Cisco IOS Object Tracking subsystem report for the specified object number, use the **event track** command in applet configuration mode. To remove the report event criteria, use the **no** form of this command.

**event** [ *label* ] [**tag** *event-tag*] **track** *object-number* [**state** {**up**| **down**| **any**}] [**maxrun** *maxruntime-number*]

**no event** [ *label* ] [**tag** *event-tag*] **track** *object-number* [**state** {**up**| **down**| **any**}] [**maxrun** *maxruntime-number*]

## 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. |
| *label* | (Optional) Unique identifier that can be any string. If the string contains embedded blanks, enclose it in double quotation marks. |
| *object-number* | Tracked object number in the range from 1 to 500, inclusive. The number is defined using the **track stub** command. |
| **state** | (Optional) Specifies that the tracked object transition will cause an event to be raised. |
| **up** | (Optional) Specifies that an event will be raised when the tracked object transitions from a down state to an up state. |
| **down** | (Optional) Specifies that an event will be raised when the tracked object transitions from an up state to a down state. |
| **any** | (Optional) Specifies that an event will be raised when the tracked object transitions to or from any state. This is the default. |
| **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 between 0 and 31536000, inclusive, and where mmm must be an integer representing milliseconds between 0 and 999). |

**Command Default** No EEM event criteria are specified.

**Command Modes** Applet configuration (config-applet)

**Command History**

| Release | Modification |
|---|---|
| 12.4(2)T | This command was introduced. |
| 12.2(31)SB3 | This command was integrated into Cisco IOS Release 12.2(31)SB3. |
| 12.2(33)SRB | This command was integrated into Cisco IOS Release 12.2(33)SRB. |
| 12.4(20)T | The **tag** and **maxrun**keywords were added to support multiple event statements within an applet. |
| 12.2(33)SXI | This command was integrated into Cisco IOS Release 12.2(33)SXI. |

**Usage Guidelines** There are two entry variables associated with this command:

- _track_number--Number of the tracked object that caused the event to be triggered.
- _track_state--State of the tracked object when the event was triggered; valid states are “up” or “down.”

This command is used to help track objects using EEM. Each tracked object is identified by a unique number that is specified on the tracking command-line interface (CLI). Client processes such as EEM use this number to track a specific object. The tracking process periodically polls the tracked objects and notes any change of value. The changes in the tracked object are communicated to interested client processes, either immediately or after a specified delay. The object values are reported as either up or down.

**Examples** The following example shows how to specify event criteria based on a tracked object:

```
event manager applet track-ten
 event track 10 state any
 action 1.0 track set 10 state up
 action 2.0 track read 10
```

## Related Commands

| Command | Description |
|---|---|
| **action track read** | Specifies the action of reading the state of a tracked object when an EEM applet is triggered. |
| **action track set** | Specifies the action of setting the state of a tracked object when an EEM applet is triggered. |
| **event manager applet** | Registers an event applet with the Embedded Event Manager and enters applet configuration mode. |
| **show track** | Displays tracking information. |
| **track stub** | Creates a stub object to be tracked. |