



Event MIB and Expression MIB Enhancements

This document provides information about the several existing Simple Network Management Protocol (SNMP) MIBs that are enhanced and new SNMP MIBs that are added.

- [Finding Feature Information, page 1](#)
- [Information about Event MIB and Expression MIB, page 1](#)
- [How to Configure Event MIB and Expression MIB, page 3](#)
- [Configuration Examples for Event MIB and Expression MIB, page 25](#)
- [Additional References, page 26](#)
- [Feature Information for Event MIB and Expression MIB Enhancements, page 29](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and 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 table.

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

Information about Event MIB and Expression MIB

Event MIB

The Event MIB provides the ability to monitor MIB objects on a local or remote system using the SNMP, and initiates simple actions whenever a trigger condition is met. For example, an SNMP trap can be generated when an object is modified. When the notifications are triggered through events, the NMS does not need to constantly poll managed devices to track changes.

By allowing the SNMP notifications to take place only when a specified condition is met, Event MIB reduces the load on affected devices and improves the scalability of network management solutions.

The Event MIB operates based on event, object lists configured for the event, event action, trigger, and trigger test.

Events

The event table defines the activities to be performed when an event is triggered. These activities include sending a notification and setting a MIB object. The event table has supplementary tables for additional objects that are configured according to event action. If the event action is set to notification, notifications are sent out whenever the object configured for that event is modified.

Object List

The object table lists objects that can be added to notifications based on trigger, trigger test type, or the event that sends a notification. The Event MIB allows wildcarding, which enables you to monitor multiple instances of an object. To specify a group of object identifiers, you can use the wildcard option.

Trigger

The trigger table defines conditions to trigger events. The trigger table lists the objects to be monitored and associates each trigger with an event. An event occurs when a trigger is activated. To create a trigger, you should configure a trigger entry in the `mteTriggerTable` of the Event MIB. This trigger entry specifies the object identifier of the object to be monitored. Each trigger is configured to monitor a single object or a group of objects specified by a wildcard (*). The Event MIB process checks the state of the monitored object at specified intervals.

Trigger Test

The trigger table has supplementary tables for additional objects that are configured based on the type of test performed for a trigger. For each trigger entry type such as existence, threshold, or Boolean, the corresponding tables (existence, threshold, and Boolean tables) are populated with the information required to perform the test. The Event MIB allows you to set event triggers based on existence, threshold, and Boolean trigger types. When the specified test on an object returns a value of *true*, the trigger is activated. You can configure the Event MIB to send out notifications to the interested host when a trigger is activated.

Expression MIB

The Expression MIB allows you to create expressions based on a combination of objects. The expressions are evaluated according to the sampling method. The Expression MIB supports the following types of object sampling:

- Absolute
- Delta
- Changed

If there are no delta or change values in an expression, the expression is evaluated when a requester attempts to read the value of the expression. In this case, all requesters get a newly calculated value.

For expressions with delta or change values, an evaluation is performed for every sampling. In this case, requesters get the value as of the last sample period.

Absolute Sampling

Absolute sampling uses the value of the MIB object during sampling.

Delta Sampling

Delta sampling is used for expressions with counters that are identified based on delta (difference) from one sample to the next. Delta sampling requires the application to do continuous sampling, because it uses the value of the last sample.

Changed Sampling

Changed sampling uses the changed value of the object since the last sample.

How to Configure Event MIB and Expression MIB

Configuring Event MIB Using SNMP

The Event MIB can be configured using SNMP directly. In this procedure, the Event MIB is configured to monitor the delta values of ifInOctets for all interfaces once per minute. If any of the samples exceed the specified threshold, a trap notification will be sent.

There are no Cisco software configuration tasks associated with the Event MIB. All configuration of Event MIB functionality must be performed through applications using SNMP. This section provides a sample configuration session using a network management application on an external device. See the “Additional References” section for information about configuring SNMP on your Cisco routing device.

All configuration of Event MIB functionality must be performed through applications using SNMP. The following section provides a step-by-step Event MIB configuration using SNMP research tools available for Sun workstations. The **setany** commands given below are executed using the SNMP application.



Note

These are not Cisco command line interface commands. It is assumed that SNMP has been configured on your routing device.

In this configuration, the objective is to monitor ifInOctets for all interfaces. The Event MIB is configured to monitor the delta values of ifInOctets for all interfaces once per minute. If any of the samples exceed the specified threshold of 30, a Trap notification will be sent.

There are five parts to the following example:

Setting the Trigger in the Trigger Table

Perform this task to set the trigger in the trigger table.

SUMMARY STEPS

1. `setany -v2c $ADDRESS private mteTriggerEntryStatus.4.106.111.104.110.1 -i 5`
2. `setany -v2c $ADDRESS private mteTriggerValueID.4.106.111.104.110.1 -d 1.3.6.1.2.1.2.2.1.10`
3. `setany -v2c $ADDRESS private mteTriggerValueIDWildcard.4.106.111.104.110.1 -i 1`
4. `setany -v2c $ADDRESS private mteTriggerTest.4.106.111.104.110.1 -o '20'`
5. `setany -v2c $ADDRESS private mteTriggerFrequency.4.106.111.104.110.1 -g 60`
6. `setany -v2c $ADDRESS private mteTriggerSampleType.4.106.111.104.110.1 -i 2`
7. `setany -v2c $ADDRESS private mteTriggerEnabled.4.106.111.104.110.1 -i 1`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>setany -v2c \$ADDRESS private mteTriggerEntryStatus.4.106.111.104.110.1 -i 5</code>	Creates a trigger row in the table with john as the mteOwner and 1 as the trigger name. <ul style="list-style-type: none"> • The index is given in decimal representation of the ASCII value of john.1.
Step 2	<code>setany -v2c \$ADDRESS private mteTriggerValueID.4.106.111.104.110.1 -d 1.3.6.1.2.1.2.2.1.10</code>	Sets the mteTriggerValueID to the OID to be watched. <ul style="list-style-type: none"> • In this example, the OID to be monitored is ifInOctets.
Step 3	<code>setany -v2c \$ADDRESS private mteTriggerValueIDWildcard.4.106.111.104.110.1 -i 1</code>	Sets the mteTriggerValueIDWildcard to TRUE to denote a object referenced through wildcarding.
Step 4	<code>setany -v2c \$ADDRESS private mteTriggerTest.4.106.111.104.110.1 -o '20'</code>	Sets the mteTriggerTest to Threshold.
Step 5	<code>setany -v2c \$ADDRESS private mteTriggerFrequency.4.106.111.104.110.1 -g 60</code>	Sets the mteTriggerFrequency to 60. This means that ifInOctets are monitored once every 60 seconds.
Step 6	<code>setany -v2c \$ADDRESS private mteTriggerSampleType.4.106.111.104.110.1 -i 2</code>	Sets the sample type to Delta.
Step 7	<code>setany -v2c \$ADDRESS private mteTriggerEnabled.4.106.111.104.110.1 -i 1</code>	Enables the trigger.

Creating an Event in the Event Table

Perform this task to create an event in the event table.

SUMMARY STEPS

1. `setany -v2c $ADDRESS private mteEventEntryStatus.4.106.111.104.110.101.118.101.110. 116 -i 5`
2. `setany -v2c $ADDRESS private mteEventEnabled.4.106.111.104.110.101.118.101.110.116 -i 1`
3. `setany -v2c $ADDRESS private mteEventEntryStatus.4.106.111.104.110.101.118.101.110. 116 -i 1`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>setany -v2c \$ADDRESS private mteEventEntryStatus.4.106.111.104.110.101.118.101.110. 116 -i 5</code>	Creates a row in the Event Table. <ul style="list-style-type: none"> • The mteOwner here is again john, and the event is mteEventName. • The default action is to send out a notification.
Step 2	<code>setany -v2c \$ADDRESS private mteEventEnabled.4.106.111.104.110.101.118.101.110.116 -i 1</code>	Enables the Event.
Step 3	<code>setany -v2c \$ADDRESS private mteEventEntryStatus.4.106.111.104.110.101.118.101.110. 116 -i 1</code>	Makes the EventRow active.

Setting and Activating the Trigger Threshold in the Trigger Table

Perform this task to set the trigger threshold in the trigger table.

SUMMARY STEPS

1. `setany -v2c $ADDRESS private mteTriggerThresholdRising.4.106.111.104.110.1 -i 30`
2. `setany -v2c $ADDRESS private mteTriggerThresholdRisingEventOwner.4.106.111.104.110.1 -D "owner"`
3. `setany -v2c $ADDRESS private mteTriggerEntryStatus.4.106.111.104.110.1 -i 1`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>setany -v2c \$ADDRESS private mteTriggerThresholdRising.4.106.111.104.110.1 -i 30</code>	Sets the Rising Threshold value to 30. Note that a row would already exist for john.1 in the Trigger Threshold Table.
Step 2	<code>setany -v2c \$ADDRESS private mteTriggerThresholdRisingEventOwner.4.106.111.104.110.1 -D "owner"</code>	Points to the entry in the Event Table that specifies the action to be performed.

	Command or Action	Purpose
	Example: <pre>setany -v2c \$ADDRESS private mteTriggerThresholdRisingEvent.4.106.111.104.110.1 -D "event"</pre>	
Step 3	<pre>setany -v2c \$ADDRESS private mteTriggerEntryStatus.4.106.111.104.110.1 -i 1</pre>	Makes the trigger active.

What to Do Next

To confirm that the above configuration is working, ensure that at least one of the interfaces gets more than 30 packets in a minute. This should cause a trap to be sent out after one minute.

Monitoring and Maintaining Event MIB

Use the following commands to monitor Event MIB activity from the Cisco command line interface:

Command	Purpose
debug management event mib	Prints messages to the screen whenever the Event MIB evaluates a specified trigger. These messages are given in realtime and are intended to be used by technical support engineers for troubleshooting purposes.
show management event	Displays the SNMP Event values that have been configured on your routing device through the use of the Event MIB.

Configuring Event MIB Using Command Line Interface

The Event MIB can be configured using SNMP directly. In this procedure, the Event MIB is configured to monitor delta values of ifInOctets for all interfaces once per minute. If any of the samples exceed the specified threshold, a trap notification will be sent.

Depending on your release, note that the Event MIB feature is enhanced to add command line interface commands to configure the events, event action, and trigger.

This section contains the following tasks to configure the Event MIB:

Configuring Scalar Variables

Perform this task to configure scalar variables for Event MIB.

Before You Begin

To configure the scalar variables for Event MIB, you should be familiar with the Event MIB scalar variables.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp mib event sample minimum *value***
4. **snmp mib event sample instance maximum *value***
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	snmp mib event sample minimum <i>value</i> Example: Device(config)# snmp mib event sample minimum 10	Sets the minimum value for object sampling.
Step 4	snmp mib event sample instance maximum <i>value</i> Example: Device(config)# snmp mib event sample instance maximum 50	Sets the maximum value for object instance sampling.
Step 5	end Example: Device(config)# end	Exits global configuration mode and enters privileged EXEC mode.

Configuring Event MIB Object List

To configure the Event MIB, you need to set up a list of objects that can be added to notifications according to the trigger, trigger test, or event.

Before You Begin

To configure the Event MIB object list, you should be familiar with the Event MIB objects and object identifiers, which can be added to notifications according to the event, trigger, or trigger test.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp mib event object list owner** *object-list-owner* **name** *object-list-name* *object-number*
4. **object id** *object-identifier*
5. **wildcard**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	snmp mib event object list owner <i>object-list-owner</i> name <i>object-list-name</i> <i>object-number</i> Example: Device(config)# snmp mib event object list owner owner1 name objectA 10	Configures the Event MIB object list.
Step 4	object id <i>object-identifier</i> Example: Device(config-event-objlist)# object id ifInOctets	Specifies the object identifier for the object configured for the event.

	Command or Action	Purpose
Step 5	wildcard Example: Device(config-event-objlist)# wildcard	(Optional) Starts a wildcard search for object identifiers. By specifying a partial object identifier, you can obtain a list of object identifiers.
Step 6	end Example: Device(config-event-objlist)# end	Exits object list configuration mode.

Configuring Event

Perform this task to configure a management event.

Before You Begin

To configure a management event, you should be familiar with SNMP MIB events and object identifiers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp mib event owner** *event-owner* **name** *event-name*
4. **description** *event-description*
5. **enable**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	snmp mib event owner <i>event-owner</i> name <i>event-name</i> Example: Device(config)# snmp mib event owner owner1 event EventA	Enters event configuration mode.
Step 4	description <i>event-description</i> Example: Device(config-event)# description "EventA is an RMON event"	Describes the function and use of the event.
Step 5	enable Example: Device(config-event)# enable	Enables the event. Note The event can be executed during an event trigger only if it is enabled.
Step 6	end Example: Device(config-event)# end	Exits event configuration mode.

Configuring Event Action

By configuring an event action, you can define the actions that an application can perform during an event trigger. The actions for an event include sending a notification, setting a MIB object and so on. You can set the event action information to either **set** or **notification**. The actions for the event can be configured only in event configuration mode.

The following sections contain the tasks to configure an event action:

Configuring Action Notification

Perform this task to set the notification action for the event.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp mib event owner** *event-owner* **name** *event-name*
4. **action notification**
5. **object id** *object-id*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	snmp mib event owner <i>event-owner name</i> <i>event-name</i> Example: Device(config)# snmp mib event owner owner1 name test	Enters event configuration mode.
Step 4	action notification Example: Device(config-event)# action notification	Sets the notification action for an event and enters action notification configuration mode. Note If the event action is set to notification, a notification is generated whenever an object associated with an event is modified.
Step 5	object id <i>object-id</i> Example: Device(config-event-action-notification)# object id ifInOctets	Configures an object for action notification. When the object specified is modified, a notification will be sent to the host system.
Step 6	end Example: Device(config-event-action-notification)# end	Exits action notification configuration mode.

Configuring Action Set

Perform this task to set actions for an event.

SUMMARY STEPS

1. **action set**
2. **object id *object-id***
3. **value *integer-value***
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	action set Example: Device(config-event)# action set	Enters action set configuration mode.
Step 2	object id <i>object-id</i> Example: Device(config-event-action-set)# object id ifInOctets	Configures an object for action set. • When the object specified is modified, a specified action will be performed.
Step 3	value <i>integer-value</i> Example: Device(config-event-action-set)# value 10	Sets a value for the object.
Step 4	end Example: Device(config-event-action-set)# end	Exits action set configuration mode.

Configuring Event Trigger

By configuring an event trigger, you can list the objects to monitor, and associate each trigger to an event. Perform this task to configure an event trigger.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp mib event trigger owner** *trigger-owner* **name** *trigger-name*
4. **description** *trigger-description*
5. **frequency** *seconds*
6. **object list owner** *object-list-owner* **name** *object-list-name*
7. **object id** *object-identifier*
8. **enable**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	snmp mib event trigger owner <i>trigger-owner</i> name <i>trigger-name</i> Example: Device(config)# snmp mib event trigger owner owner1 name EventTriggerA	Enables event trigger configuration mode for the specified event trigger.
Step 4	description <i>trigger-description</i> Example: Device(config-event-trigger)# description "EventTriggerA is an RMON alarm."	Describes the function and use of the event trigger.
Step 5	frequency <i>seconds</i> Example: Device(config-event-trigger)# frequency 120	Configures the waiting time (number of seconds) between trigger samples.
Step 6	object list owner <i>object-list-owner</i> name <i>object-list-name</i> Example: Device(config-event-trigger)# object list owner owner1 name ObjectListA	Specifies the list of objects that can be added to notifications.
Step 7	object id <i>object-identifier</i> Example: Device(config-event-trigger)# object id ifInOctets	Configures object identifiers for an event trigger.
Step 8	enable Example: Device(config-event-trigger)# enable	Enables the event trigger.

	Command or Action	Purpose
Step 9	end Example: Device(config-event-trigger)# end	Exits event trigger configuration mode.

Configuring Existence Trigger Test

You should configure this trigger type in event trigger configuration mode.

Perform this task to configure trigger parameters for the test existence trigger type.

SUMMARY STEPS

1. **test existence**
2. **event owner** *event-owner* **name** *event-name*
3. **object list owner** *object-list-owner* **name** *object-list-name*
4. **type** {present | absent | changed}
5. **startup** {present | absent}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	test existence Example: Device(config-event-trigger)# test existence	Enables test existence configuration mode.
Step 2	event owner <i>event-owner</i> name <i>event-name</i> Example: Device(config-event-trigger-existence)# event owner owner1 name EventA	Configures the event for the existence trigger test.
Step 3	object list owner <i>object-list-owner</i> name <i>object-list-name</i> Example: Device(config-event-trigger-existence)# object list owner owner1 name ObjectListA	Configures the list of objects for the existence trigger test.
Step 4	type {present absent changed}	Performs the specified type of existence test.

	Command or Action	Purpose
	<p>Example: Device(config-event-trigger-existence)# type present</p>	<p>Existence tests are of the following three types:</p> <ul style="list-style-type: none"> • Present—Setting type to present tests if the objects that appear during the event trigger exist. • Absent—Setting type to absent tests if the objects that disappear during the event trigger exist. • Changed—Setting type to changed tests if the objects that changed during the event trigger exist.
Step 5	<p>startup {present absent}</p> <p>Example: Device(config-event-trigger-existence)# startup present</p>	Triggers an event if the test is performed successfully.
Step 6	<p>end</p> <p>Example: Device(config-event-trigger-existence)# end</p>	Exits existence trigger test configuration mode.

Configuring Boolean Trigger Test

You should configure this trigger test in event trigger configuration mode.

Perform this task to configure trigger parameters for the Boolean trigger type.

SUMMARY STEPS

1. **test boolean**
2. **comparison** {unequal | equal | less | lessOrEqual | greater | greaterOrEqual}
3. **value** *integer-value*
4. **object list owner** *object-list-owner* **name** *object-list-name*
5. **event owner** *event-owner* **name** *event-name*
6. **startup**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	test boolean Example: Device(config-event-trigger)# test boolean	Enables Boolean trigger test configuration mode.
Step 2	comparison {unequal equal less lessOrEqual greater greaterOrEqual} Example: Device(config-event-trigger-boolean)# comparison unequal	Performs the specified Boolean comparison test. • The value for the Boolean comparison test can be set to unequal, equal, less, lessOrEqual, greater, or greaterOrEqual.
Step 3	value integer-value Example: Device(config-event-trigger-boolean)# value 10	Sets a value for the Boolean trigger test.
Step 4	object list owner object-list-owner name object-list-name Example: Device(config-event-trigger-boolean)# object list owner owner1 name ObjectListA	Configures the list of objects for the Boolean trigger test.
Step 5	event owner event-owner name event-name Example: Device(config-event-trigger-boolean)# event owner owner1 name EventA	Configures the event for the Boolean trigger type.
Step 6	startup Example: Device(config-event-trigger-boolean)# startup	Triggers an event if the test is performed successfully.
Step 7	end Example: Device(config-event-trigger-boolean)# end	Exits Boolean trigger test configuration mode.

Configuring Threshold Trigger Test

You should configure this trigger test in event trigger configuration mode.

Perform this task to configure trigger parameters for the threshold trigger test.

SUMMARY STEPS

1. **test threshold**
2. **object list owner** *object-list-owner* **name** *object-list-name*
3. **rising** *integer-value*
4. **rising event owner** *event-owner* **name** *event-name*
5. **falling** *integer-value*
6. **falling event owner** *event-owner* **name** *event-name*
7. **delta rising** *integer-value*
8. **delta rising event owner** *event-owner* **name** *event-name*
9. **delta falling** *integer-value*
10. **delta falling event owner** *event-owner* **name** *event-name*
11. **startup** {**rising** | **falling** | **rising-or-falling**}
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	test threshold Example: Device(config-event-trigger)# test threshold	Enables threshold trigger test configuration mode.
Step 2	object list owner <i>object-list-owner</i> name <i>object-list-name</i> Example: Device(config-event-trigger-threshold)# object list owner owner1 name ObjectListA	Configures the list of objects for the threshold trigger test.
Step 3	rising <i>integer-value</i> Example: Device(config-event-trigger-threshold)# rising 100	Sets the rising threshold to the specified value.
Step 4	rising event owner <i>event-owner</i> name <i>event-name</i> Example: Device(config-event-trigger-threshold)# rising event owner owner1 name EventA	Configures an event for the threshold trigger test for the rising threshold.
Step 5	falling <i>integer-value</i> Example: Device(config-event-trigger-threshold)# falling 50	Sets the falling threshold to the specified value.

	Command or Action	Purpose
Step 6	falling event owner <i>event-owner</i> name <i>event-name</i> Example: Device(config-event-trigger-threshold)# falling event owner owner1 name EventB	Configures an event for the threshold trigger test for the falling threshold.
Step 7	delta rising <i>integer-value</i> Example: Device(config-event-trigger-threshold)# delta rising 30	Sets the delta rising threshold to the specified value when the sampling method specified for the event trigger is delta.
Step 8	delta rising event owner <i>event-owner</i> name <i>event-name</i> Example: Device(config-event-trigger-threshold)# delta rising event owner owner1 name EventC	Configures an event for the threshold trigger test for the delta rising threshold.
Step 9	delta falling <i>integer-value</i> Example: Device(config-event-trigger-threshold)# delta falling 10	Sets the delta falling threshold to the specified value when the sampling method specified for the event trigger is delta.
Step 10	delta falling event owner <i>event-owner</i> name <i>event-name</i> Example: Device(config-event-trigger-threshold)# delta falling event owner owner1 name EventAA	Configures an event for the threshold target test for the delta falling threshold.
Step 11	startup { rising falling rising-or-falling } Example: Device(config-event-trigger-threshold)# startup rising	Triggers an event when the threshold trigger test conditions are met.
Step 12	end Example: Device(config-event-trigger-threshold)# end	Exits threshold trigger test configuration mode.

Configuring Expression MIB Using SNMP

Expression MIB can be configured using SNMP directly.

There are no Cisco software configuration tasks associated with Expression MIB. All configurations of the Expression MIB functionality must be performed through applications using SNMP. This section provides a

sample configuration session using a network management application on an external device. See the Additional References section for information about configuring SNMP on your Cisco routing device.

The following section provides a step-by-step Expression MIB configuration using SNMP research tools available for Sun workstations. The **setany** commands given below are executed using the SNMP application. Note that these commands are not Cisco command line interface commands. It is assumed that SNMP has been configured on your routing device.

In the following configuration, a wildcarded expression involving the addition of the counters ifInOctets and ifOutOctets are evaluated.

SUMMARY STEPS

1. **setany -v2c \$SNMP_HOST private expResourceDeltaMinimum.0 -i 60**
2. **setany -v2c \$SNMP_HOST private expExpressionIndex.116.101.115.116 -g 9**
3. **setany -v2c \$SNMP_HOST private expNameStatus.116.101.115.116 -i 5**
4. **setany -v2c \$SNMP_HOST private expExpressionComment.9 -D "test expression"**
5. **setany -v2c \$SNMP_HOST private expExpression.9 -D '\$1 + \$2'**
6. **setany -v2c \$SNMP_HOST private expObjectID.9.1 -d ifInOctets**
7. **setany -v2c \$SNMP_HOST private expObjectSampleType.9.1 -i 2**
8. **setany -v2c \$SNMP_HOST private expObjectIDWildcard.9.1 -i 1**
9. **setany -v2c \$SNMP_HOST private expObjectStatus.9.1 -i 1**
10. **setany -v2c \$SNMP_HOST private expNameStatus.116.101.115.116 -i 1**

DETAILED STEPS

	Command or Action	Purpose
Step 1	setany -v2c \$SNMP_HOST private expResourceDeltaMinimum.0 -i 60	Sets the minimum delta interval that the system will accept.
Step 2	setany -v2c \$SNMP_HOST private expExpressionIndex.116.101.115.116 -g 9	Sets the identification number used for identifying the expression. <ul style="list-style-type: none"> • For example, expName can be 'test', which is ASCII 116.101.115.116.
Step 3	setany -v2c \$SNMP_HOST private expNameStatus.116.101.115.116 -i 5	Creates an entry in the expNameStatusTable. <p>Note When an entry is created in the expNameTable, it automatically creates an entry in the expExpressionTable.</p>
Step 4	setany -v2c \$SNMP_HOST private expExpressionComment.9 -D "test expression"	Sets the object to a comment to explain the use or meaning of the expression. <ul style="list-style-type: none"> • Here, the comment is "test expression".
Step 5	setany -v2c \$SNMP_HOST private expExpression.9 -D '\$1 + \$2'	Sets the object expExpression to an expression that needs to be evaluated. <ul style="list-style-type: none"> • In this expression, "\$1" corresponds to "ifInOctets", "\$2" corresponds to "ifOutOctets", and the expression signifies the addition of the two counter objects.

	Command or Action	Purpose
Step 6	<pre>setany -v2c \$SNMP_HOST private expObjectID.9.1 -d ifInOctets</pre> <p>Example:</p> <pre>setany -v2c \$SNMP_HOST private expObjectID.9.2 -d ifOutOctets</pre>	<p>Specifies the object identifiers used in the expression mentioned in the above set for calculation.</p> <ul style="list-style-type: none"> Here, the number "9", suffixed to the object expObjectID, corresponds to the unique identifier used for identifying the expression, and the number "1" following "9" is another unique identifier used for identifying an object within the expression. Set the expObjectID to the two objects used in forming the expression.
Step 7	<pre>setany -v2c \$SNMP_HOST private expObjectSampleType.9.1 -i 2</pre> <p>Example:</p> <pre>setany -v2c \$SNMP_HOST private expObjectSampleType.9.2 -i 2</pre>	<p>Sets the type of sampling to be done for objects in the expression.</p> <ul style="list-style-type: none"> There are two types of sampling: a) Absolute b) Delta. Here, the sample type has been set to "Delta".
Step 8	<pre>setany -v2c \$SNMP_HOST private expObjectIDWildcard.9.1 -i 1</pre> <p>Example:</p> <pre>setany -v2c \$SNMP_HOST private expObjectIDWildcard.9.2 -i 1</pre>	<p>Specifies whether the expObjectID is wildcarded or not. In this case, both the expObjectID are wildcarded.</p>
Step 9	<pre>setany -v2c \$SNMP_HOST private expObjectStatus.9.1 -i 1</pre> <p>Example:</p> <pre>setany -v2c \$SNMP_HOST private expObjectStatus.9.2 -i 1</pre>	<p>Sets the rows in the expObjectTable to active.</p>
Step 10	<pre>setany -v2c \$SNMP_HOST private expNameStatus.116.101.115.116 -i 1</pre>	<p>Sets the rows in the expNameTable to active so that the value of the expression can be evaluated.</p> <ul style="list-style-type: none"> The value of the expression can now be obtained from the expValueTable.

Configuring Expression MIB Using Command Line Interface

Expression MIB can be configured using SNMP directly. Depending on your release, you can find that the Expression MIB feature is enhanced to add command line interface commands to configure expressions. You should be familiar with expressions, object identifiers, and sampling methods before configuring Expression MIB.

The following sections contain the tasks to configure Expression MIB:

Configuring Expression MIB Scalar Objects

Expression MIB has the following scalar objects:

- expResourceDeltaMinimum
- expResourceDeltaWildcardInstanceMaximum

Perform this task to configure Expression MIB scalar objects.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp mib expression delta minimum** *seconds*
4. **snmp mib expression delta wildcard maximum** *number-of-instances*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	snmp mib expression delta minimum <i>seconds</i> Example: Device(config)# snmp mib expression delta minimum 20	(Optional) Sets the minimum delta interval in seconds. Note Application may use larger values for this minimum delta interval to lower the impact of constantly computing deltas. For larger delta sampling intervals, the application samples less often and has less overhead. By using this command, you can enforce a lower overhead for all expressions created after the delta interval is set.
Step 4	snmp mib expression delta wildcard maximum <i>number-of-instances</i> Example: Device(config)# snmp mib expression delta maximum 120	(Optional) Limits the maximum number of dynamic instance entries for wildcard delta objects in expressions. <ul style="list-style-type: none"> • For a given delta expression, the number of dynamic instances is the number of values that meet all criteria to exist, times the number of delta values in the expression. • There is no preset limit for the instance entries and it is dynamic based on a system's resources.

	Command or Action	Purpose
Step 5	end Example: Device (config) # end	Exits global configuration mode.

Configuring Expressions

Perform this task to configure an expression.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp mib expression owner** *expression-owner* **name** *expression-name*
4. **description** *expression-description*
5. **expression** *expression*
6. **delta interval** *seconds*
7. **value type** {counter32 | unsigned32 | timeticks | integer32 | ipaddress | octetstring | objectid | counter64}
8. **enable**
9. **object** *object-number*
10. **id** *object-identifier*
11. **wildcard**
12. **discontinuity object** *discontinuity-object-id* [wildcard] [type {timeticks | timestamp | date-and-time}]
13. **conditional object** *conditional-object-id* [wildcard]
14. **sample** {absolute | delta | changed}
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	<p>snmp mib expression owner <i>expression-owner</i> name <i>expression-name</i></p> <p>Example: Device(config-expression)# snmp mib expression owner owner1 name ExpA</p>	Enables the expression to be configured.
Step 4	<p>description <i>expression-description</i></p> <p>Example: Device(config-expression)# description this expression is created for the sysLocation MIB object</p>	Configures a description for the expression.
Step 5	<p>expression <i>expression</i></p> <p>Example: Device(config-expression)# expression (\$1+\$2)*800/\$3</p>	<p>Configures the expression to be evaluated.</p> <p>Note The expressions are in ANSI C syntax. However, the variables in an expression are defined as a combination of the dollar sign (\$) and an integer that corresponds to the object number of the object used in evaluating the expression.</p>
Step 6	<p>delta interval <i>seconds</i></p> <p>Example: Device(config-expression)# delta interval 180</p>	Configures the sampling interval for objects in the expression if the sampling method is delta.
Step 7	<p>value type {counter32 unsigned32 timeticks integer32 ipaddress octetstring objectid counter64}</p> <p>Example: Device(config-expression)# value type counter32</p>	Sets the specified value type for the expression.
Step 8	<p>enable</p> <p>Example: Device(config-expression)# enable</p>	Enables an expression for evaluation.
Step 9	<p>object <i>object-number</i></p> <p>Example: Device(config-expression)# object 2</p>	<p>Configures the objects that are used for evaluating an expression.</p> <ul style="list-style-type: none"> The object number is used to associate the object with the variables in the expression. The variable corresponding to the object is \$ and object number. Thus, the variable in the example used here corresponds to \$10.
Step 10	<p>id <i>object-identifier</i></p> <p>Example: Device(config-expression-object)# id ifInOctets</p>	Configures the object identifier.

	Command or Action	Purpose
Step 11	<p>wildcard</p> <p>Example: Device(config-expression-object)# wildcard</p>	(Optional) Enables a wildcarded search for objects used in evaluating an expression.
Step 12	<p>discontinuity object <i>discontinuity-object-id</i> [wildcard] [type {timeticks timestamp date-and-time}]</p> <p>Example: Device(config-expression-object)# discontinuity object sysUpTime</p>	<p>(Optional) Configures the discontinuity properties for the object if the object sampling type is set to delta or changed. The discontinuity object ID supports normal checking for a discontinuity in a counter.</p> <ul style="list-style-type: none"> Using the wildcard keyword, you can enable wildcarded search for objects with discontinuity properties. Using the type keyword, you can set value for objects with discontinuity properties.
Step 13	<p>conditional object <i>conditional-object-id</i> [wildcard]</p> <p>Example: Device(config-expression-object)# conditional object mib-2.90.1.3.1.1.2.3.112.99.110.4.101.120.112.53</p>	<p>(Optional) Configures the conditional object identifier.</p> <ul style="list-style-type: none"> Using the wildcard keyword, you can enable a wildcarded search for conditional objects with discontinuity properties.
Step 14	<p>sample {absolute delta changed}</p> <p>Example: Device(config-expression-object)# sample delta</p>	<p>Enables the specified sampling method for the object. This example uses the delta sampling method.</p> <p>You can set any of the three sampling methods: absolute, delta, and changed.</p> <ul style="list-style-type: none"> Absolute sampling—Uses the value of the MIB object during sampling. Delta sampling—Uses the last sampling value maintained in the application. This method requires applications to do continuous sampling. Changed sampling—Uses the changed value of the object since the last sample.
Step 15	<p>end</p> <p>Example: Device(config-expression-object)# end</p>	Exits expression object configuration mode.

Configuration Examples for Event MIB and Expression MIB

Example: Configuring Event MIB from SNMP

The following example shows how to configure scalar variables for an event:

```
Device# configure terminal
Device(config)# snmp mib event sample minimum 10
Device(config)# snmp mib event sample instance maximum 50
Device(config)# end
```

The following example shows how to configure object list for an event:

```
Device# configure terminal
Device(config)# snmp mib event object list owner owner1 name objectA number 1
Device(config-event-objlist)# object id ifInOctets
Device(config-event-objlist)# wildcard
Device(config-event-objlist)# end
```

The following example shows how to configure an event:

```
Device# configure terminal
Device(config)# snmp mib event owner owner1 event EventA
Device(config-event)# description "eventA is an RMON event."
Device(config-event)# enable
Device(config-event)# end
```

The following example shows how to set the notification action for an event:

```
Device(config-event)# action notification
Device(config-event-action-notification)# object id ifInOctets
Device(config-event-action-notification)# end
```

The following example shows how to set actions for an event:

```
Device(config-event)# action set
Device(config-event-action-set)# object id ifInOctets
Device(config-event-action-set)# value 10
Device(config-event-action-set)# end
```

The following example shows how to configure trigger for an event:

```
Device# configure terminal
Device(config)# snmp mib event trigger owner owner1 name EventTriggerA
Device(config-event-trigger)# description EventTriggerA is an RMON alarm.
Device(config-event-trigger)# frequency 120
Device(config-event-trigger)# object list owner owner1 name ObjectListA
Device(config-event-trigger)# object id ifInOctets
Device(config-event-trigger)# enable
Device(config-event-trigger)# end
```

The following example shows how to configure existence trigger test:

```
Device(config-event-trigger)# test existence
Device(config-event-trigger-existence)# event owner owner1 name EventA
Device(config-event-trigger-existence)# object list owner owner1 name ObjectListA
Device(config-event-trigger-existence)# type present
Device(config-event-trigger-existence)# startup present
Device(config-event-trigger-existence)# end
```

The following example shows how to configure Boolean trigger test:

```
Device(config-event-trigger)# test boolean
Device(config-event-trigger-boolean)# comparison unequal
```

Example: Configuring Expression MIB from SNMP

```

Device(config-event-trigger-boolean)# value 10
Device(config-event-trigger-boolean)# object list owner owner1 name ObjectListA
Device(config-event-trigger-boolean)# event owner owner1 name EventA
Device(config-event-trigger-boolean)# startup
Device(config-event-trigger-boolean)# end

```

The following example shows how to configure threshold trigger test:

```

Device(config-event-trigger)# test threshold
Device(config-event-trigger-threshold)# object list owner owner1 name ObjectListA
Device(config-event-trigger-threshold)# rising 100
Device(config-event-trigger-threshold)# rising event owner owner1 name EventA
Device(config-event-trigger-threshold)# falling 50
Device(config-event-trigger-threshold)# falling event owner owner1 name EventA
Device(config-event-trigger-threshold)# delta rising 30
Device(config-event-trigger-threshold)# delta rising event owner owner1 name EventA
Device(config-event-trigger-threshold)# delta falling 10
Device(config-event-trigger-threshold)# delta falling event owner owner1 name EventA
Device(config-event-trigger-threshold)# startup rising
Device(config-event-trigger-threshold)# end

```

Example: Configuring Expression MIB from SNMP

The following example shows how to configure the Expression MIB by using the `snmp mib expression` command in global configuration mode:

```

Device(config)# snmp mib expression owner pcn name exp6
Device(config-expression)# description this expression is created for the sysLocation MIB
object
Device(config-expression)# expression ($1+$2)*800/$3
Device(config-expression)# delta interval 120
Device(config-expression)# value type counter32
Device(config-expression)# enable
Device(config-expression)# object 2
Device(config-expression-object)# id ifInOctets
Device(config-expression-object)# wildcard
Device(config-expression-object)# discontinuity object sysUpTime
Device(config-expression-object)# conditional object
mib-2.90.1.3.1.1.2.3.112.99.110.4.101.120.112.53 wildcard
Device(config-expression-object)# sample delta
Device(config-expression-object)# end

```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
SNMP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS SNMP Command Reference
Cisco implementation of RFC 1724, RIP Version 2 MIB Extensions	RIPv2 Monitoring with SNMP Using the RFC 1724 MIB Extensions feature module

Related Topic	Document Title
DSP Operational State Notifications for notifications to be generated when a digital signaling processor (DSP) is used	DSP Operational State Notifications feature module

Standards and RFCs

Standard/RFC	Title
CBC-DES (DES-56) standard	<i>Symmetric Encryption Protocol</i>
STD: 58	<i>Structure of Management Information Version 2 (SMIv2)</i>
RFC 1067	<i>A Simple Network Management Protocol</i>
RFC 1091	<i>Telnet terminal-type option</i>
RFC 1098	<i>Simple Network Management Protocol (SNMP)</i>
RFC 1157	<i>Simple Network Management Protocol (SNMP)</i>
RFC 1213	<i>Management Information Base for Network Management of TCP/IP-based internets: MIB-II</i>
RFC 1215	<i>Convention for defining traps for use with the SNMP</i>
RFC 1901	<i>Introduction to Community-based SNMPv2</i>
RFC 1905	<i>Common Management Information Services and Protocol over TCP/IP (CMOT)</i>
RFC 1906	<i>Telnet X Display Location Option</i>
RFC 1908	<i>Simple Network Management Protocol (SNMP)</i>
RFC 2104	<i>HMAC: Keyed-Hashing for Message Authentication</i>
RFC 2206	<i>RSVP Management Information Base using SMIv2</i>
RFC 2213	<i>Integrated Services Management Information Base using SMIv2</i>
RFC 2214	<i>Integrated Services Management Information Base Guaranteed Service Extensions using SMIv2</i>
RFC 2271	<i>An Architecture for Describing SNMP Management Frameworks</i>
RFC 2570	<i>Introduction to Version 3 of the Internet-standard Network Management Framework</i>

Standard/RFC	Title
RFC 2578	<i>Structure of Management Information Version 2 (SMIv2)</i>
RFC 2579	<i>Textual Conventions for SMIv2</i>
RFC 2580	<i>Conformance Statements for SMIv2</i>
RFC 2981	<i>Event MIB</i>
RFC 2982	<i>Distributed Management Expression MIB</i>
RFC 3413	<i>SNMPv3 Applications</i>
RFC 3415	<i>View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)</i>
RFC 3418	<i>Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)</i>

MIBs

MIB	MIBs Link
<ul style="list-style-type: none"> • Circuit Interface Identification MIB • Cisco SNMPv2 • Ethernet-like Interfaces MIB • Event MIB • Expression MIB Support for Delta, Wildcarding, and Aggregation • Interfaces Group MIB (IF-MIB) • Interfaces Group MIB Enhancements • MIB Enhancements for Universal Gateways and Access Servers • MSDP MIB • NTP MIB • Response Time Monitor MIB • Virtual Switch MIB 	<p>To locate and download MIBs for selected platforms, releases, and feature sets, use Cisco MIB Locator found at the following URL:</p> <p>http://www.cisco.com/go/mibs</p>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Event MIB and Expression MIB Enhancements

The following table provides release information about the feature or features described in this module. This table 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.

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/cisco/featurenavigator](#). An account on Cisco.com is not required.

Table 1: Feature Information for Event MIB and Expression MIB Enhancements

Feature Name	Releases	Feature Information
Event MIB and Expression MIB Enhancements	12.2(33)SRE 12.2(50)SY 12.4(20)T 15.0(1)S	The Event MIB and Expression MIB feature introduces command line interface commands to configure the Event MIB and Expression MIB. The following commands were introduced or modified: action (event) , comparison , conditional object , delta (test threshold) , delta interval , description (event) , description (expression) , description (trigger) , discontinuity object (expression) , enable (event) , enable (expression) , event owner , expression , falling (test threshold) , frequency (event trigger) , object (expression) , object id , object list , rising (test threshold) , sample (expression) , snmp mib event object list , snmp mib event owner , snmp mib event trigger owner , snmp mib expression delta , snmp mib expression owner , startup (test boolean) , startup (test existence) , startup (test threshold) , test (event trigger) , type (test existence) , value (test boolean) , value type , and wildcard (expression) .

