

# Chapter 11

## Monitor RMON Alarms and Events

The remote monitoring (RMON) alarms and events feature can be used to monitor integer-valued MIB objects, standard or enterprise-specific, on a Juniper Networks router. Configuration and operational information are in the MIB objects defined in alarmTable, eventTable, and logTable in RFC 2819. Additional information is defined by the Juniper Networks enterprise-specific extension to alarmTable defined in jnxRmonMIB (jnx-rmon-mib.txt).

This chapter covers the following main topics:

RMON Alarms on page 125

RMON Events on page 131

### RMON Alarms

An RMON alarm identifies:

A specific MIB object that is monitored.

The frequency at which it is sampled.

The method of sampling.

The thresholds against which the monitored values are compared.

An RMON alarm can also identify a specific eventTable entry to be triggered when a threshold is crossed.

Configuration and operational values are defined in alarmTable in RFC 2819. Additional operational values are defined in Juniper Networks enterprise-specific extensions to alarmTable (jnxRmonAlarmTable).

This section covers the following topics:

alarmTable on page 126

jnxRmonAlarmTable on page 127

Use alarmTable to Monitor MIB Objects on page 127

## **alarmTable**

alarmTable in the RMON MIB allows you to monitor and poll the following:

alarmIndex—The index value for alarmTable that identifies a specific entry.

alarmInterval—The interval in seconds over which data is sampled and compared with the rising and falling thresholds.

alarmVariable—The MIB variable that is monitored by the alarm entry.

alarmSampleType—The method of sampling the selected variable and calculating the value to be compared against the thresholds.

alarmValue—The value of the variable during the last sampling period. This is the value that is compared with the rising and falling thresholds.

alarmStartupAlarm—The alarm that can be sent when the entry is first activated.

alarmRisingThreshold—The upper threshold for the sampled variable.

alarmFallingThreshold—The lower threshold for the sampled variable.

alarmRisingEventIndex—The eventTable entry that is used when a rising threshold is crossed.

alarmFallingEventIndex—The eventTable entry that is used when a falling threshold is crossed.

alarmStatus—Add and remove entries from the table. It can also be used to change the state of an entry to allow modifications.



**Note**

If this object is not set to valid, no action will be taken by the associated event alarm.

## ***jnxRmonAlarmTable***

The `jnxRmonAlarmTable` is a Juniper Networks enterprise-specific extension to `alarmTable`. It provides additional operational information and includes the following objects:

`jnxRmonAlarmGetFailCnt`—The number of times the internal Get request for the variable monitored by this entry has failed.

`jnxRmonAlarmGetFailTime`—The value of `sysUpTime` when an internal Get request for the variable monitored by this entry last failed.

`jnxRmonAlarmGetFailReason`—The reason an internal Get request for the variable monitored by this entry last failed.

`jnxRmonAlarmGetOKTime`—The value of `sysUpTime` when an internal Get request for the variable monitored by this entry succeeded and the entry left the `getFailure` state.

`jnxRmonAlarmState`—The current state of this RMON alarm entry.

To view the Juniper Networks enterprise-specific extensions to the RMON alarm and event MIB, see [www.juniper.net/techpubs/software/junos56/swconfig56-net-mgmt/html/mib-jnx-rmon.txt](http://www.juniper.net/techpubs/software/junos56/swconfig56-net-mgmt/html/mib-jnx-rmon.txt). For more information on the Juniper Networks enterprise-specific extensions to the RMON events and alarms MIB, see “Interpret the Enterprise-Specific RMON Events and Alarms MIB” on page 223.

## ***Use alarmTable to Monitor MIB Objects***

To use `alarmTable` to monitor a MIB object, perform the following tasks:

Create an Alarm Entry on page 127

Configure the Alarm MIB Objects on page 128

Activate a New Row in `alarmTable` on page 130

Modify an Active Row in `alarmTable` on page 131

Deactivate a Row in `alarmTable` on page 131

## ***Create an Alarm Entry***

To create an alarm entry, first create a new row in `alarmTable` using the `alarmStatus` object. For example, create alarm #1 using the UCD command-line utilities:

```
snmpset -Os -v2c router community alarmStatus.1 i createRequest
```

## Configure the Alarm MIB Objects

Once you have created the new row in `alarmTable`, configure the following alarm MIB objects:

`alarmInterval` on page 128

`alarmVariable` on page 128

`alarmSampleType` on page 129

`alarmValue` on page 129

`alarmStartupAlarm` on page 129

`alarmRisingThreshold` on page 129

`alarmFallingThreshold` on page 130

`alarmOwner` on page 130

`alarmRisingEventIndex` on page 130

`alarmFallingEventIndex` on page 130



**Note**

Other than `alarmStatus`, none of the objects in the entry can be modified if the associated `alarmStatus` object is set to valid.

### *alarmInterval*

The interval in seconds over which data is sampled and compared with the rising and falling thresholds. For example, to set `alarmInterval` for alarm #1 to 30 seconds, use the following SNMP Set request:

```
snmpset -Os -v2c router community alarmInterval.1 i 30
```

### *alarmVariable*

The object identifier of the variable to be sampled. During a Set request, if the supplied variable name is not available in the selected MIB view, a `badValue` error is returned. If at any time the variable name of an established `alarmEntry` is no longer available in the selected MIB view, the probe changes the status of `alarmVariable` to invalid. For example, to identify `ifInOctets.61` as the variable to be monitored, use the following SNMP Set request:

```
snmpset -Os -v2c router community alarmVariable.1 o .1.3.6.1.2.1.2.2.1.10.61
```

*alarmSampleType*

The method of sampling the selected variable and calculating the value to be compared against the thresholds. If the value of this object is `absoluteValue`, the value of the selected variable is compared directly with the thresholds at the end of the sampling interval. If the value of this object is `deltaValue`, the value of the selected variable at the last sample is subtracted from the current value, and the difference is compared with the thresholds. For example, to set `alarmSampleType` for alarm #1 to `deltaValue`, use the following SNMP Set request:

```
snmpset -Os -v2c router community alarmSampleType.1 i deltaValue
```

*alarmValue*

The value of the variable during the last sampling period. This is the value that is compared with the rising and falling thresholds. If the sample type is `deltaValue`, this value equals the difference between the samples at the beginning and end of the period. If the sample type is `absoluteValue`, this value equals the sampled value at the end of the period.

*alarmStartupAlarm*

An alarm that is sent when this entry is first set to valid. If the first sample after this entry becomes valid is greater than or equal to `risingThreshold`, and `alarmStartupAlarm` is equal to `risingAlarm` or `risingOrFallingAlarm`, then a single rising alarm is generated. If the first sample after this entry becomes valid is less than or equal to `fallingThreshold` and `alarmStartupAlarm` is equal to `fallingAlarm` or `risingOrFallingAlarm`, then a single falling alarm is generated. For example, to set `alarmStartupAlarm` for alarm #1 to `risingOrFallingAlarm`, use the following SNMP Set request:

```
snmpset -Os -v2c router community alarmStartupAlarm.1 i risingOrFallingAlarm
```

*alarmRisingThreshold*

A threshold for the sampled variable. When the current sampled value is greater than or equal to this threshold, and the value at the last sampling interval is less than this threshold, a single event is generated. A single event is also generated if the first sample after this entry becomes valid is greater than or equal to this threshold, and the associated `alarmStartupAlarm` is equal to `risingAlarm` or `risingOrFallingAlarm`. After a rising event is generated, another rising event cannot be generated until the sampled value falls below this threshold and reaches `alarmFallingThreshold`. For example, to set `alarmRisingThreshold` for alarm #1 to 100000, use the following SNMP Set request:

```
snmpset -Os -v2c router community alarmRisingThreshold.1 i 100000
```

*alarmFallingThreshold*

A threshold for the sampled variable. When the current sampled value is less than or equal to this threshold, and the value at the last sampling interval is greater than this threshold, a single event is generated. A single event is also generated if the first sample after this entry becomes valid is less than or equal to this threshold, and the associated alarmStartupAlarm is equal to fallingAlarm or risingOrFallingAlarm. After a falling event is generated, another falling event cannot be generated until the sampled value rises above this threshold and reaches alarmRisingThreshold. For example, to set alarmFallingThreshold for alarm #1 to 10000, use the following SNMP Set request:

```
snmpset -Os -v2c router community alarmFallingThreshold.1 i 10000
```

*alarmOwner*

Any text string specified by the creating management application or the CLI. Typically, it is used to identify a network manager (or application) and can be used for fine access control between participating management applications.

*alarmRisingEventIndex*

The index of the eventEntry object that is used when a rising threshold is crossed. If there is no corresponding entry in eventTable, then no association exists. If this value is zero, no associated event is generated because zero is not a valid event index. For example, to set alarmRisingEventIndex for alarm #1 to 10, use the following SNMP Set request:

```
snmpset -Os -v2c router community alarmRisingEventIndex.1 i 10
```

*alarmFallingEventIndex*

The index of the eventEntry object that is used when a falling threshold is crossed. If there is no corresponding entry in eventTable, then no association exists. If this value is zero, no associated event is generated because zero is not a valid event index. For example, to set alarmFallingEventIndex for alarm #1 to 10, use the following SNMP Set request:

```
snmpset -Os -v2c router community alarmFallingEventIndex.1 i 10
```

**Activate a New Row in alarmTable**

To activate a new row in alarmTable, set alarmStatus to valid using an SNMP Set request:

```
snmpset -Os -v2c router community alarmStatus.1 i valid
```

**Modify an Active Row in alarmTable**

To modify an active row, first set alarmStatus object to underCreation using an SNMP Set request:

```
snmpset -Os -v2c router community alarmStatus.1 i underCreation
```

Then change the row contents using an SNMP Set request:

```
snmpset -Os -v2c router community alarmFallingThreshold.1 i 1000
```

Finally, activate the row by setting alarmStatus object to valid using an SNMP Set request:

```
snmpset -Os -v2c router community alarmStatus.1 i valid
```

**Deactivate a Row in alarmTable**

To deactivate a row in alarmTable, set alarmStatus to invalid using an SNMP Set request:

```
snmpset -Os -v2c router community alarmStatus.1 i invalid
```

**RMON Events**

An RMON event allows you to log the crossing of thresholds of other MIB objects. It is defined in eventTable for the RMON MIB.

This section covers the following topics:

eventTable on page 131

Use eventTable to Log Alarms on page 132

**eventTable**

eventTable contains the following objects:

eventIndex—An index that uniquely identifies an entry in eventTable. Each entry defines one event that is to be generated when the appropriate conditions occur.

eventDescription—A comment describing the event entry.

eventType—Type of notification that the probe makes about this event.

eventCommunity—Trap group used if an SNMP trap is to be sent. If eventCommunity is not configured, a trap is sent to each trap group configured with the rmon-alarm category.

eventLastTimeSent—Value of sysUpTime when this event entry last generated an event.

eventOwner—Any text string specified by the creating management application or the CLI. Typically, it is used to identify a network manager (or application) and can be used for fine access control between participating management applications.

eventStatus—Status of this event entry.



**Note**

If this object is not set to valid, no action is taken by the associated event entry. When this object is set to valid, all previous log entries associated with this entry (if any) will be deleted.

## ***Use eventTable to Log Alarms***

To use eventTable to log alarms, perform the following tasks:

Create an Event Entry on page 132

Configure the MIB Objects on page 132

Activate the New Row in eventTable on page 134

Deactivate a Row in eventTable on page 134

## ***Create an Event Entry***

The RMON eventTable controls the generation of notifications from the router. Notifications can be logs (entries to logTable and syslogs) or SNMP traps. Each event entry can be configured to generate any combination of these notifications (or no notification). When an event specifies that an SNMP trap is to be generated, the trap group that is used when sending the trap is specified by the value of the associated eventCommunity object. Consequently, the community in the trap message will match the value specified by eventCommunity. If nothing is configured for eventCommunity, a trap is sent using each trap group that has the rmon-alarm category configured.

## ***Configure the MIB Objects***

Once you have created the new row in eventTable, set the following objects:

eventType on page 133

eventCommunity on page 133

eventOwner on page 133

eventDescription on page 134

The eventType object is required. All other objects are optional.

*eventType*

The type of notification that the router generates when the event is triggered.

This object can be set to the following values:

log—Adds the event entry to logTable.

log-and-trap—Sends an SNMP trap and creates a log entry.

none—Sends no notification.

snmptrap—Sends an SNMP trap.


For example, to set eventType for event #1 to log-and-trap, use the following SNMP Set request:

```
snmpset -Os -v2c router community eventType.1 i log-and-trap
```

*eventCommunity*

The trap group that is used when generating a trap (if eventType is configured to send traps). If that trap group has the rmon-alarm trap category configured, a trap is sent to all the targets configured for that trap group. The community string in the trap matches the name of the trap group (and hence, the value of eventCommunity). If nothing is configured, traps are sent to each group with the rmon-alarm category set. For example, to set eventCommunity for event #1 to boy-elroy, use the following SNMP Set request:

```
snmpset -Os -v2c router community eventCommunity.1 s "boy-elroy"
```



The eventCommunity object is optional. If you do not set this object, then the field is left blank.


**Note**

*eventOwner*

Any text string specified by the creating management application or the CLI. Typically, it is used to identify a network manager (or application) and can be used for fine access control between participating management applications.

For example, to set eventOwner for event #1 to george jetson, use the following SNMP Set request:

```
snmpset -Os -v2c router community eventOwner.1 s "george jetson"
```



eventOwner object is optional. If you do not set this object, then the field is left blank.

**Note**

- *eventDescription*

- Any text string specified by the creating management application or the CLI. The use of this string is application dependent.

- For example, to set eventDescription for event #1 to spacelys sprockets, use the following SNMP Set request:

- ```
snmpset -Os -v2c router community eventDescription.1 s "spacelys sprockets"
```



**Note**

The eventDescription object is optional. If you do not set this object, then the field is left blank.

- **Activate the New Row in eventTable**

- To activate the new row in eventTable, set eventStatus to valid using an SNMP Set request such as:

- ```
snmpset -Os -v2c router community eventStatus.1 i valid
```

- **Deactivate a Row in eventTable**

- To deactivate a row in eventTable, set eventStatus to invalid using an SNMP Set request such as:

- ```
snmpset -Os -v2c router community eventStatus.1 i invalid
```