How To Configure RMON Alarm and Event Settings Using SNMP Commands

Help us help you.

☐
Please rate this document.

◯ Excellent

◯ Good

◯ Average

◯ Fair

◯ Poor

☐
This document solved my problem.

◯ Yes

◯ No

◯ Just browsing

☐
Suggestions for improvement:

(256 character limit)

☐
[Send]

Contents

Introduction

This document provides a sample configuration for remote monitoring (RMON) alarm and event settings using SNMP commands.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

To follow the procedures in this document, your device needs to support the RMON-MIB. You can check this at Cisco IOS MIB Tools (registered customers only) .

The information in this document was created from the devices in a specific lab environment. All of the

devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

For more information on document conventions, refer to Cisco Technical Tips Conventions.

Background Information

The purpose of remote monitoring (RMON) alarm and event is to monitor a certain MIB object on the device, and warn the system administrator if one of those values is going out of the defined range.

The alarm monitors a specific object in the MIB and triggers an event when the condition (falling or rising threshold) is reached.

The event is the trap or log generated when the alarm triggers it. An example of a rising and falling threshold is:

```
n=value monitored by the alarm. The falling threshold is "5" and the rising thr
```

The alarm triggers an event when one of the two values is reached.
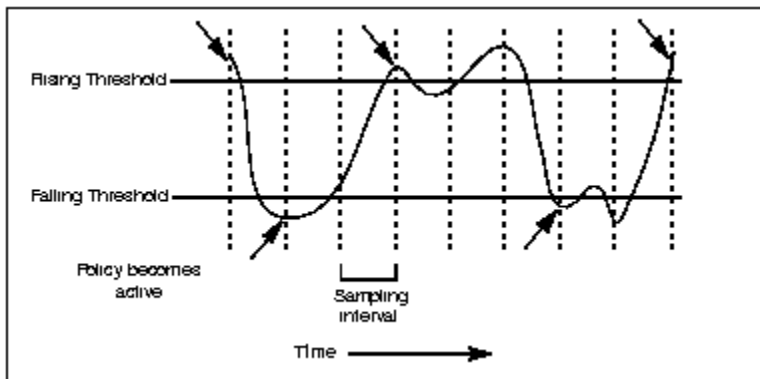
Value	Trap	Comment
n1=16	rising	reached the rising value : 15
n2=4	falling	reached the falling value : 5
n3=6	none	between 5 and 15
n4=6	rising	reached the rising value : 15
n5=13	none	good condition
n6=20	none	above 15 but did not go under 5 since the last event
n7=4	falling	reached the falling value : 5
n8=20	none	under 5 but did not go above 15 since the last event

| n9=16 | rising | reached the rising value : 15 |

You can configure RMON alarm and event on routers using the command line interface (CLI) (refer to Configuring RMON Alarm and Event Settings from the Command Line Interface), and on routers and switches using Simple Network Management Protocol (SNMP) commands. The parameters to modify are part of the RMON-MIB.

Step-by-Step Procedure

Creating An Event

Use this command to create an event:

```
# snmpset -c <read_write_community> <device_name> .1.3.6.1.2.1.16.9.1.1.x.y <va
```

First choose the event ID (variable *y*).

Follow this procedure to create an event. For each step, there is a description of the step, the name of the MIB object to modify, the object ID (OID), the *<variable type>*, and the *<value>* from the generic command.

1. Erase an eventual old event that would have used ID="y" (make sure you first check that you don't need it anymore. Otherwise, use another ID).

    ```
    * eventStatus / .1.3.6.1.2.1.16.9.1.1.7.y
    * variable type=integer
    * value=4
    ```

    Note: Use the same command to erase the event when needed.

2. Enter event creation mode:

    ```
    eventStatus / .1.3.6.1.2.1.16.9.1.1.7.y
    * variable type=integer
    * value v=2
    ```

3. Specify the event description:

    ```
    * eventDescription / .1.3.6.1.2.1.16.9.1.1.2.y
    * variable type=string (for Net-snmp) or octetsting (for Openview)
    * value = a description of the event
    ```

4. Specify the type of event you want:

    ```
    * eventType / .1.3.6.1.2.1.16.9.1.1.3.y
    * variable type=integer
    * value =
    "1" => none
    "2" => log
    "3" => snmp-trap
    "4" => log-and-trap
    ```

5. Specify the community string for the trap:

```
 * eventCommunity / .1.3.6.1.2.1.16.9.1.1.4.y
 * variable type=string (for Net-snmp) or octetsting (for Openview)
 * value="<trap_community_string>"
```

6. Specify the owner of the event:

```
 * eventOwner / .1.3.6.1.2.1.16.9.1.1.6.y
 * variable type=string (for Net-snmp) or octetsting (for Openview)
 * value="<event_owner>"
```

7. Activate the event:

```
 * eventStatus / .1.3.6.1.2.1.16.9.1.1.7.y
 * variable type=integer
 * value=1
```

Creating An Alarm

Use this command to create an alarm:

```
# snmpset -c .1.3.6.1.2.1.16.3.1.1.x.y <read_write_community> <device_name> <va
```

1. Erase an eventual old alarm that would have used ID=**y** (first check if you don't need it anymore. Otherwise, use another ID):

```
 * alarmStatus / .1.3.6.1.2.1.16.3.1.1.12.y
 * variable type=integer
 * value=4
```

2. Enter alarm creation mode:

```
 * alarmStatus / .1.3.6.1.2.1.16.3.1.1.12.y
 * variable type=integer
 * value=2
```

3. Set the interval (in seconds) over which the data is sampled and compared with the rising and falling thresholds:

```
 * alarmInterval / .1.3.6.1.2.1.16.3.1.1.2.y
 * variable type=integer
 * value=<n_seconds>
```

4. Specify the OID that you want to monitor:

```
 * alarmVariable / .1.3.6.1.2.1.16.3.1.1.3.y
 * variable type=objid (for Net-snmp) or objectidentifier (for Openview)
 * value=<oid_to_check>
```

5. Define the type of sample you want:

```
                   * alarmSampleType / .1.3.6.1.2.1.16.3.1.1.4.y
                   * variable type=integer
                   * value=<rising_threshold>
                   "1" => absoluteValue
                   "2" => deltaValue
```

6. Specify what triggers an alarm:

```
                   * alarmStartupAlarm / .1.3.6.1.2.1.16.3.1.1.6.y
                   * variable type=integer
                   * value=
                   "1" => risingAlarm
                   "2" => fallingAlarm
                   "3" => risingOrFallingAlarm
```

7. Define the rising threshold:

```
                   * alarmRisingThreshold / .1.3.6.1.2.1.16.3.1.1.7.y
                   * variable type=integer
                   * value=<rising_threshold>
```

8. Define the falling threshold:

```
                   * alarmFallingThreshold / .1.3.6.1.2.1.16.3.1.1.8.y
                   * variable type=integer
                   * value=<falling_threshold>
```

9. Specify the event ID you want to trigger when the rising threshold is crossed:

```
                   * alarmRisingEventIndex / .1.3.6.1.2.1.16.3.1.1.9.y
                   * variable type=integer
                   * value=<event_ID>
```

10. Specify the event ID when the falling threshold is crossed:

```
                   * alarmFallingEventIndex / .1.3.6.1.2.1.16.3.1.1.9.y
                   * variable type=integer
                   * value=<event_ID>
```

11. Specify the owner of the alarm:

```
                   * alarmOwner / .1.3.6.1.2.1.16.3.1.1.11.y
                   * variable type=string (for Net-snmp) or octetsting (for Openview)
                   * value=<owner>
```

12. Activate the alarm:

```
                   * alarmStatus / .1.3.6.1.2.1.16.3.1.1.12.y
                   * variable type=integer
                   * value=1
```

Example

In this example, **safari** is used to send a trap when the number of bytes going into interface 12 during

the last two minutes is above 140000000 or below 10.

Safari is Cisco IOS 2500 Software (C2500-JS-L), Version 12.1(9), RELEASE SOFTWARE (fc1).

This example was also tried successfully on WS-C6506 Software, Version NmpSW: 6.1(1b).

Note: In the Catalyst, there is no CLI command to check the configuration, but it can be done with the **snmpwalk** command on the server.

On the router and the switch, this configuration survives a reload.

```
safari# show rmon events
Event table is empty

# snmpset -c private safari .1.3.6.1.2.1.16.9.1.1.7.123 integer 4
16.9.1.1.7.123 = 4
# snmpset -c private safari .1.3.6.1.2.1.16.9.1.1.7.123 integer 2
16.9.1.1.7.123 = 2

safari#show rmon events
Event 123 is under creation, owned by
Description is
Event firing causes nothing, last fired 00:00:00

# snmpset -c private safari .1.3.6.1.2.1.16.9.1.1.2.123 string "test_event"
16.9.1.1.2.123 = "test_event"
# snmpset -c private safari .1.3.6.1.2.1.16.9.1.1.3.123 integer 4
16.9.1.1.3.123 = 4
# snmpset -c private safari .1.3.6.1.2.1.16.9.1.1.4.123 string "public"
16.9.1.1.4.123 = "public"
# snmpset -c private safari .1.3.6.1.2.1.16.9.1.1.6.123  string "event_owner"
16.9.1.1.6.123 = "event_owner"
# snmpset -c private safari .1.3.6.1.2.1.16.9.1.1.7.123 integer 1
16.9.1.1.7.123 = 1

safari# show rmon events
Event 123 is active, owned by event_owner
Description is test_event
Event firing  causes log and trap to community public, last fired 00:00:00

safari# show rmon alarm
Alarm table is empty

# snmpset -c private safari .1.3.6.1.2.1.16.3.1.1.12.321 integer 2
16.3.1.1.12.321 = 2

safari# show rmon alarm
Alarm 321 is under creation, owned by
Monitors ccitt.0 every 10 second(s)
Taking absolute samples, last value was 0
Rising threshold is 0, assigned to event 0
Falling threshold is 0, assigned to event 0
On startup enable rising or falling alarm

# snmpset -c private safari .1.3.6.1.2.1.16.3.1.1.2.321 integer 120
16.3.1.1.2.321 = 120
# snmpset -c private safari .1.3.6.1.2.1.16.3.1.1.3.321 objid .1.3.6.1.2.1.2.2.
16.3.1.1.3.321 = OID: interfaces.ifTable.ifEntry.ifInOctets.12
```

```
# snmpset -c private safari .1.3.6.1.2.1.16.3.1.1.4.321 integer 2
16.3.1.1.4.321 = 2
# snmpset -c private  safari .1.3.6.1.2.1.16.3.1.1.6.321 integer 3
16.3.1.1.6.321 = 3
# snmpset -c  private safari .1.3.6.1.2.1.16.3.1.1.7.321 integer 140000000
16.3.1.1.7.321 = 140000000
# snmpset -c private safari .1.3.6.1.2.1.16.3.1.1.8.321 integer 10
16.3.1.1.8.321 = 10
# snmpset -c private safari .1.3.6.1.2.1.16.3.1.1.9.321  integer 123
16.3.1.1.9.321 = 123
# snmpset -c private safari .1.3.6.1.2.1.16.3.1.1.10.321  integer 123
16.3.1.1.10.321 = 123
# snmpset -c private safari .1.3.6.1.2.1.16.3.1.1.11.321  string "alarm_owner"
16.3.1.1.11.321 = "alarm_owner"
# snmpset -c private safari  .1.3.6.1.2.1.16.3.1.1.12.321 integer 1
16.3.1.1.12.321 = 1

safari# show rmon alarm
Alarm 321 is active, owned by alarm_owner
Monitors ifEntry.10.1 every 120 second(s)
Taking delta samples, last value was 130244
Rising threshold is 140000000, assigned to event 123
Falling threshold is 10, assigned to event 123
On startup enable rising or falling alarm
```

Verify

There is currently no verification procedure available for this configuration.

Troubleshoot

There is currently no specific troubleshooting information available for this configuration.

Cisco Support Community - Featured Conversations

Cisco Support Community is a forum for you to ask and answer questions, share suggestions, and collaborate with your peers. Below are just some of the most recent and relevant conversations happening right now.

Want to see more? Join us by clicking **here**

Rmon Event/alarm problem **g.chambers** **1 Reply** 20-ene-2005 12:36

NAM - how can i configure alarm by... **johnyoon75** **1 Reply** 11-oct-2004 12:49

WLC complete event / trap dictionary?
https://supportforums.cisco.com/people/desmckee%40hotmail.com **1 Reply** 18-jun-2010 11:40

information about interface (can i use... **ogadri** **1 Reply** 20-jun-2007 6:22

CW2000-how to enhance existing package?... **nicolaswatson** **4 Replies** 27-may-2003 20:03

snmp vs rmon **branfarm1** **1 Reply** 29-mar-2007 9:28

Linksys SRW2048 RMON Event **koleckarr** **1 Reply** 17-mar-2009 5:31

How to configure RMON traps on Cisco... **raheel.hasan** **3 Replies** 27-oct-2008 23:23

how to make ssh login **mitang.prajapati** **1 Reply** 10-feb-2010 4:11

enabling RMON alarm traps for a...
https://supportforums.cisco.com/people/fselvado%40hns.com **1 Reply** 02-ago-2004 6:50

Start A New Discussion   **Subscribe**

Related Information

- **Configuring RMON Alarm and Event Settings from the Command Line Interface**
- **Event MIB Support**
- **RFC 1757**
- **Technical Support - Cisco Systems**

| Home | How to Buy | Login | Profile | Feedback | Site Map | Help |