RMON (Remote Network Monitoring)

Teldat-Dm 796-I

Legal Notice

Warranty

This publication is subject to change.

Teldat offers no warranty whatsoever for information contained in this manual.

Teldat is not liable for any direct, indirect, collateral, consequential or any other damage connected to the delivery, supply or use of this manual.

Table of Contents

I Related Documents

Teldat-Dm 712-I SNMP Agent

Chapter 1 MIB RMON

1.1 Introduction

Remote network monitoring devices, often called probes, help to manage a network. Generally speaking, these devices devote a significant amount of their internal resources to managing a network. An organization must have, at least, one device per network segment to manage its intranet.

The objects defined in the RMON MIB (Remote Network Monitoring) shall act as an interface between an RMON agent and an RMON management application.

The RMON MIB goals are:

Offline Operation: The RMON MIB is designed to obtain statistics and execute tests in the network without management station intervention. This means that information is accumulated in the probe until it can be sent to the management station.

Proactive Monitoring: Monitoring can be continuous and save logs and historic data on network performance. If an error is spotted, the probe notifies this to the management station so that it reviews the data history and tries to diagnose the problem.

Problem Detection and Reporting: The probe can be configured to recognize situations, normally errors and problems. It can also be configured to log this information and report to the management station.

Data with Added Value: Given that the monitoring probe is located on the monitored part of the network, the management station can receive first-hand information about the managed network. This adds significant value when resolving any problem that arises on the monitored network.

Multiple Managers: This gives multiple management stations simultaneous access to the resources and the information from the monitoring probe.

1.2 RMON MIB: Description

The MIB objects are arranged into the following groups:

- Ethernet Statistics Group
- History Control Group
- Ethernet History Group
- Alarm Group
- Host Group
- TopN Host Group
- Matrix Group
- Filter Group
- Packet Capture Group
- Event Group

Note

In our router's current implementation, only the Alarm and Event groups are supported. This includes the alarms table, the events table and the logs table.

1.2.1 Alarm Group

The alarm group periodically takes statistical samples from variables and compares them to the previously configured thresholds. If the monitored variable value crosses the threshold, an event is generated. A hysteresis mechanism limits the generation of alarms, preventing new alarms from being created until the opposite threshold has been reached. This group is made up of *alarmTable* entries and requires the Event group to be implemented.

The alarm table stores the configuration defining the monitored variable, the sampling period, the thresholds, the type of sampling and the events associated to each threshold. The type of sampling can be absolute or delta (i.e. the value used to compare against the thresholds is the absolute value of the variable or the difference between the latest values obtained).

An entry on the alarms table is as follows:

```
AlarmEntry ::= SEQUENCE {
    alarmIndex                  INTEGER (1..65535),
    alarmInterval               INTEGER,
    alarmVariable               OBJECT IDENTIFIER,
    alarmSampleType             INTEGER,
    alarmValue                  INTEGER,
    alarmStartupAlarm           INTEGER,
    alarmRisingThreshold        INTEGER,
    alarmFallingThreshold       INTEGER,
    alarmRisingEventIndex       INTEGER (0..65535),
    alarmFallingEventIndex      INTEGER (0..65535),
    alarmOwner                  OwnerString,
    alarmStatus                 EntryStatus
}
```

alarmIndex: An index that clearly identifies an entry in the table.

alarmInterval: The interval in seconds over which the variable value is sampled and compared against the rising and falling thresholds. When using the delta sampling, care should be taking when setting this interval so that the variable does not increase by more than 2^31 – 1 during the sampling interval.

alarmVariable: The object identifier for the particular variable to be sampled. Only variables that represent an *ASN.1 INTEGER* value are sampled.

alarmSampleType: Method used to sample the selected variable and calculate the value that is going to be compared against the thresholds. If the sample type is delta, its value is the difference between the current value and the value of the variable in the previous sampling period.

alarmValue: The value of the variable in the last sampling period. If the sample type is delta, its value is the difference between the current value and the value of the variable in the previous sampling period.

alarmStartupAlarm: indicates the type of alarm first generated when the entry enters into a *valid* state. The alarm can be configured as rising, falling or both.

alarmRisingThreshold: This is the rising threshold for the sampled value. If the sampled value is greater than (or equal to) this threshold, and the value at the last sampling interval was lower than this threshold, a single rising event is generated. Another such event will not be generated until the sampled value reaches the falling threshold.

alarmFallingThreshold: This is the falling threshold for the sampled value. If the sampled value is lower than (or equal to) this threshold, and the value at the last sampling interval was greater than this threshold, a single falling event is generated. Another such event will not be generated until the sampled value reaches the rising threshold.

alarmRisingEventIndex: This is the index for the event table associated to the rising event alarm. If there is no event in the events table (or if the value is zero) no association exists.

alarmFallingEventIndex: This is the index of the event table associated to the falling event alarm. If there is no event in the events table (or the value for this is zero) no association exists.

alarmOwner: This is the name of the entity that configured this entry and uses it. If the device's SNMP agent is the entry owner, its value begins with the word *'monitor'*.

alarmStatus: This is the status of the alarm entry. This can take the following values: *valid*(1), *createRequest*(2), *underCreation*(3), *invalid*(4). If you set the *invalid* value, the entry is deleted from the table. When creating a new entry, this enters a *createRequest* state before passing to *underCreation* for entry value configuration. Once this is completed, it becomes *valid* and is ready for activation.

1.2.2 Event Group

This group controls event generation and reporting. The group implements the *eventTable* and the *logTable*.

Each event in the event table defines the event parameters that can be triggered by an alarm associated to the event. An event can be configured to create an entry in the *logs* table in cases where it is triggered. It can also be configured to send SNMP *trap* messages.

An entry on the event table is as follows:

```
EventEntry ::= SEQUENCE {
    eventIndex          INTEGER (1..65535),
    eventDescription    DisplayString (SIZE (0..127)),
    eventType           INTEGER,
    eventCommunity      OCTET STRING (SIZE (0..127)),
    eventLastTimeSent   TimeTicks,
    eventOwner          OwnerString,
    eventStatus         EntryStatus
}
```

eventIndex: An index that clearly identifies an entry in the event table.

eventDescription: A comment describing this event.

eventType: The type of notification that is generated when the event is triggered. Its value can be: *none*(1), *log*(2), *snmp-trap*(3), *log-and-trap*(4). This configures the creation of an entry in the *logs* table every time an event is triggered, a SNMP *trap* is sent, or both.

eventCommunity: If an SNMP *trap* is to be sent, it will be sent to the community (when using SNMPv1/2c) or the user specified by this parameter (when using SNMPv3).

eventLastTimeSent: This is the system time value at the time this event last triggered. If it has never been triggered, this value is zero.

eventOwner: This is the name of the entity that configured this entry and is using it. If the device's SNMP agent is the owner of the entry, its value begins with the word *'monitor'*.

eventStatus: This is the entry status. If this value is different to *valid*(1), all the associated log entries are deleted. The values accepted are the same as those for an alarm.

The table stores the *logs* that have been created by an event identifying them through the duple (*eventIndex*,*logIndex*). An entry in the **logs** table looks like this:

```
LogEntry ::= SEQUENCE {
    logEventIndex           INTEGER (1..65535),
    logIndex                INTEGER (1..2147483647),
    logTime                 TimeTicks,
    logDescription          DisplayString (SIZE (0..255))
}
```

logEventIndex: This is an index that identifies the event the entry belongs to.

logIndex: This, together with the above value, identifies the entry in the table, and identifies the entry amongst those generated by the same event.

logTime: This is the value of the system time when this entry was created.

logDescription: This is a comment describing the specific event created by the entry.

Note

The implementation executed in the RMON Router complies with the IETF RFC1757 recommendation.

Chapter 2 Configuration

2.1 Configuring RMON

RMON configuration is executed in the RMON global configuration menu. You can configure RMON alarms, RMON events and the maximum size of the *logs* table in this menu.

Access the RMON configuration menu through the router's configuration console. To access the menu, execute the following sequence of commands:

```
*config

configuration environment
Config>feature rmon

-- Remote Network Monitoring configuration --
RMON config>
```

You can also access this menu through the device's dynamic configuration console:

```
*running-config

Config$feature rmon

-- Remote Network Monitoring configuration --
RMON config$
```

The commands available in the RMON configuration menu are as follows:

```
RMON config>?
  alarm             Configure a RMON alarm
  event             Configure a RMON event
  list              List RMON info
  log-table-size    Configure LOG table size
  no                Negate a command or set its defaults
  exit
RMON config>
```

2.2 Configuration Commands

This section describes the RMON configuration commands and includes all the possible options.

The configuration commands available in the RMON configuration menu are as follows:

Command	Function
? (HELP)	Lists the available commands or their options.
ALARM	Configures an RMON alarm.
EVENT	Configures an RMON event.
LIST	Lists the current RMON configuration, including the default values.
LOG-TABLE-SIZE	Maximum size of the *logs* table for each event.
NO	Deletes a command or sets its default value.
EXIT	Returns to the configuration menu.

2.2.1 ? (HELP)

You can use the **?** (HELP) command to list all the valid commands at the layer where the router is configured. This command can be used after a specific command to list all the options available.

Syntax:

```
RMON config>?
```

Example:

```
RMON config>?
  alarm              Configure a RMON alarm
  event              Configure a RMON event
  list               List RMON info
  log-table-size     Configure LOG table size
  no                 Negate a command or set its defaults
  exit
RMON config>
```

2.2.2 ALARM

The alarm command allows you to configure an RMON alarm defining all its parameters. Once the entry is valid, the first alarm can be rising or falling.

Syntax:

```
RMON config>alarm <alarmid> <interval> <variable OID> <delta|absolute> rising-threshold <n>
 [event <id>] falling-threshold <n> [event <id>] [owner <name>]
```

Example:

```
RMON config>alarm 4 10 1.3.6.1.2.1.2.2.1.16.1 delta rising-threshold 1000 event 8
falling-threshold 1 event 9 owner palonso
RMON config>
```

The parameters for the *alarm* command are described, in order, hereunder.

2.2.2.1 Alarm index

This parameter is mandatory and identifies the alarm. You cannot have another alarm configured with the same identifier value. To modify an alarm, you must have previously deleted it using the *no* command. This value must be between 1 and 100.

2.2.2.2 Sample Interval

This parameter is mandatory and corresponds to the variable sampling interval in seconds. This value must be between 1 and 1,000 seconds.

2.2.2.3 Variable OID

This parameter is mandatory and corresponds to the object identifier (OID) that singles out the variable you wish to monitor. The value must be entered in the number nomenclature separated by points and must match the variable whose value is INTEGER. The object identifier must be between 1 and 128 characters long.

2.2.2.4 Sample type

This parameter is mandatory and defines the type of sampling you want the monitored variable to execute. It can take two values: *delta* or *absolute*.

Absolute: Defines the absolute value of the variable obtained when sampling. This is compared against the defined rising and falling thresholds. If the variable value is higher than, or equal to, the rising threshold, a rising event is generated. If the variable value is lower than, or equal to, the falling threshold, a falling event is generated. After a rising event is generated, another such event will not be generated until the sampled value falls below this threshold and reaches the falling threshold (and vice versa).

Delta: The value that is going to be compared against the defined thresholds is the difference between the current absolute value and the previously obtained variable. If the variable value is higher than, or equal to, the rising threshold, a rising event is generated. If the variable value is lower than, or equal to, the falling threshold, a falling event is generated. After a rising event is generated, another such event will not be generated until the sampled value falls below this threshold and reaches the falling threshold (and vice versa).

2.2.2.5 Rising-threshold

This parameter is mandatory and defines the alarm rising threshold value. This value must be between 0 and 4.294.967.295.

2.2.2.6 Rising event index

This parameter is optional and allows you to associate a specific event to the rising event alarm. If this triggers a rising event and the event corresponding to the identifier exists and is valid, the action defined in the event is executed (regardless of whether it is sending an SNMP *trap* or storing a *log*). This value must be between 0 and 100, the 0 value being equivalent to not setting this option.

2.2.2.7 Falling-threshold

This parameter is mandatory and defines the falling threshold alarm value. This value must be between 0 and 4.294.967.295.

2.2.2.8 Falling event index

This parameter is optional and allows you to associate a specific event to the falling event alarm. If this triggers a falling event and the event corresponding to the identifier exists and is valid, the action defined in the event is executed (regardless of whether it is sending an SNMP *trap* or storing a *log*). This value must be between 0 and 100, the 0 value being equivalent to not setting this option.

2.2.2.9 Owner

This parameter is optional and allows you to define the alarm owner entity. If you don't configure this parameter, it takes the **'monitor-owned'** value by default. We recommend that you configure this parameter if various management stations can access the same device and are going to configure their own alarms. The text you enter must be between 1 and 32 characters long.

2.2.3 EVENT

The *event* command allows you to configure an RMON event and define all its parameters.

Syntax:

```
RMON config>event <eventid> [log] [trap <community/user>] [description <text>]
[owner <name>]
```

Example:

```
RMON config>event 11 log trap RMON-RPD description "Potential broadcast storm detected" owner palonso
RMON config>
```

The parameters for the *event* command are described, in order, hereunder.

2.2.3.1 Event index

This parameter is mandatory and identifies the event. You cannot have another event configured with the same identifier value. To modify an event, you must first delete it using the ' *no*' command. This value must be between 1 and 100.

2.2.3.2 Log

This parameter is optional and allows you to activate the creation and storing of *logs* each time the event triggers. If this parameter is included, each time an event is triggered an entry with a description is stored in the *logs* table.

```
Direction: OID = value <=> threshold : alarmID, eventID
```

Direction: This can be *rising* or *falling*, depending on whether this is a rising or falling event.

OID: This is the identifier for the variable object that has generated the event.

value: Value of the variable when generating the event (it is the delta value or the absolute value, depending on the type of sampling configured in the alarm).

<=>: it is ">=" in a rising event and "<=" in a falling event.

threshold: This is the value of the threshold that has been reached.

alarmID: This is the identifier for the alarm associated to the event.

eventID: This is the identifier for the event that has generated the *log*.

The events table can be remotely consulted through an SNMP client or through the RMON monitoring menu.

Note

Deleting an event also erases all its log table entries. This is the only way of deleting the entries in the logs table, as they are read only.

2.2.3.3 Trap

This parameter is optional and activates the sending of SNMP *traps*. SNMP *traps* are sent to the community (when using SNMPv1/v2c) or the user configured in this parameter (when using SNMPv3). The community name must be between 1 and 64 characters long.

In order to send the *traps*, you need to define the community/user in the SNMP configuration menu, as well as the *traps* sending parameters. For example:

```
   protocol snmp
; -- SNMP user configuration --
      engineid local 23456789
      user palonso
;
;
      community RMON-RPD access write-read-trap
      community RMON-RPD subnet 192.168.212.14 255.255.255.255
;
      host 192.168.212.14 trap version v1 RMON-RPD all
;
      trap sending-parameters time 0s
      trap sending-parameters number 1
      trap sending-parameters reachability-checking ip-route
   exit
```

The following console can serve as an example when using SNMPv3:

```
   protocol snmp
; -- SNMP user configuration --
      engineid local 1234567890
      user maria-user auth md5 ciphered 0xD76B5D15371A70CB00C72869FA01F7BA priv des
ciphered 0xD76B5D15371A70CB00C72869FA01F7BA
;
      group maria-user v3-usm maria-group
;
      access maria-group v3-usm priv read-view _all_ write-view _all_
;
      host 192.168.212.19 trap version v3 priv maria-user all
;
      trap sending-parameters reachability-checking icmp
   exit
```

You must configure an IP management address in the IP protocol menu for the *traps* to take the agent source address.

```
   protocol ip
; -- Internet protocol user configuration --

      management-ip-address 192.168.212.133
      classless
;
   exit
;
```

Note

For further information on how to configure the router SNMP agent, please see manual Teldat-Dm712-I SNMP Agent.

Command history:

Version	Modification
10.09.24.20.08	From version 10.09.24.20.08 onwards, RMON *traps* can be sent using SNMPv3.
10.09.26	From version 10.9.26 onwards, RMON *traps* can be sent using SNMPv3.
11.00.05	From version 11.00.05 onwards, RMON *traps* can be sent using SNMPv3.
11.01.00	From version 11.01.00 onwards, RMON *traps* can be sent using SNMPv3.

2.2.3.4 Description

This parameter is optional and allows you to add a comment describing the event. The text must be between 1 and 64 characters long. If it contains spaces, it must be placed between inverted commas.

2.2.3.5 Owner

This parameter is optional and allows you to define the event owner entity. If you don't configure this parameter, it takes the *'monitor-owned'* value by default. We recommend that you configure this parameter if various management stations can access the same device and are going to configure their own events. The text you enter must be between 1 and 32 characters long.

2.2.4 LIST

This command lists the configured alarms and events, including the default values that do not appear in the commands entered in the configuration menu. You can list a specific alarm or event, by indicating its identifier, all the alarms, all the events, or all the configured alarms and events.

Syntax:

```
RMON config>list [alarm <all|n>|event <all|n>]
```

Example:

```
RMON config>list ?
  alarm    List RMON alarm info
  event    List RMON event info
  <cr>     List all RMON configuration info
RMON config>list event 11

Event identifier: 11
        Description: Potential broadcast storm detected
        Event type: Log & Trap
        Trap community/user: RMON-RPD
        Owner: palonso
RMON config>list alarm 1

Alarm identifier: 1
        Variable: 1.3.6.1.4.1.2007.4.1.2.2.2.21.1.1.7.1.1
        Sampling Type: delta
        Sampling period: 10
        Rising threshold: 1000
        Rising event identifier: 11
        Falling threshold: 200
        Falling event identifier: 12
        Owner: palonso
RMON config>
```

Command history:

Version	Modification
10.09.24.20.08	The command output has been modified as of version 10.09.24.20.08 to show the *Trap user* (since RMON traps may be sent using SNMPv3).
10.09.26	The command output has been modified as of version 10.9.26 to show the *Trap user* (since RMON traps may be sent using SNMPv3).
11.00.05	The command output has been modified as of version 11.00.05 to show the *Trap user* (since RMON traps may be sent using SNMPv3).
11.01.00	The command output has been modified as of version 11.01.00 to show the *Trap user* (since RMON traps may be sent using SNMPv3).

2.2.5 LOG-TABLE-SIZE

Use this command to configure the maximum size of the *logs* table for any event (i.e. the maximum number of *log* identifiers for an event). If the table is full, it will begin to reuse *log* identifiers. Default is 200 and you can configure any value between 20 and 1000.

Syntax:

```
RMON config>log-table-size <n>
```

Example:

```
RMON config>log-table-size ?
  <20..1000>    Size of the RMON Log table
RMON config>log-table-size 300
RMON config>
```

Chapter 3 Monitoring

3.1 Accessing the RMON Monitoring

To access RMON Monitoring, first access the general monitoring table and then go to the RMON feature.

```
*monitor
Console Operator
+feature rmon
-- RMON (Remote Network Monitoring) console --
RMON+
```

3.2 RMON Monitoring

This section lists and describes the RMON monitoring commands.

Command	Function
? (HELP)	Allows you to view the commands available or their available options.
CLEAR	Deletes the statistics information.
LIST	Lists the statistics information.
EXIT	Returns to the previous menu.

3.2.1 ? (HELP)

You can use the **?** (HELP) command to list all the valid commands at the layer where the router is configured. This command can be used after a specific command to list all the available options.

Syntax:

```
RMON+?
```

Example:

```
RMON+?
  clear    Clear RMON statistics
  list     Show RMON statistics
  exit
RMON+
```

3.2.2 CLEAR

Use this command to clear the statistics for an alarm or an event. Select it through its identifier or delete the RMON global statistics.

Syntax:

```
RMON+clear <alarm <n>|event <n>|statistics>
```

Example:

```
RMON+?
RMON+clear ?
  alarm         Clear statistics of one RMON alarm
  event         Clear statistics of one RMON event
  statistics    Clear RMON statistics
RMON+clear alarm 5
RMON+clear event 10
RMON+clear statistics
RMON+
```

3.2.3 LIST

Use this command to display the data for the alarms, events and *logs* table, as well as the statistics for each alarm, event or the ROM global statistics.

Syntax:

```
RMON+list <alarm <id|all>|event <id|all>|alarm-table|event-table|log-table|statistics>
```

The available options are as follows:

```
RMON+list ?
  alarm          Show detailed statistics of one or all RMON alarms
  alarm-table    List all RMON Alarm Table information
  event          Show detailed statistics of one or all RMON events
  event-table    List all RMON Event Table information
  log-table      List all RMON Log Table information
  statistics     Show RMON statistics
RMON+
```

3.2.3.1 list alarm-table

This option displays the content of the alarms table. The meaning of each column is as follows:

ID: alarm identifier.

Status: alarm status.

Variable: object identifier (OID) for the monitored variable.

Type: type of sampling.

Intvl: sampling interval in seconds.

RiThr: rising threshold.

RiEv: rising event identifier.

FaThr: falling threshold.

FaEv: falling event identifier.

Owner: name of the alarm owner.

Example:

```
RMON+list alarm-table

 Alarm Table information:

  ID   Status    Variable                          Type    Intvl  RiThr    RiEv  FaThr    FaEv    Owner
 ------------------------------------------------------------------------------------------------------
  1  valid   .1.3.6.1.4.1.2007
             .4.1.2.2.2.21.1.1.7.1.1                delta   10s   1000     11    200        12    palonso
  2  valid   .1.3.6.1.2.1.2.2.1.10.1                delta   10s   2000      3    100        4     palonso
  3  valid   .1.3.6.1.2.1.2.2.1.16.1                delta   10s   2000      5    100        6     palonso

RMON+
```

3.2.3.2 list alarm <id>

This option displays the data and statistics for an alarm. The meaning of each line is as follows:

Alarm ID: alarm identifier.

Variable: object identifier (OID) for the monitored variable.

Owner: name of the alarm owner.

Status: alarm status.

Sample Type: type of sampling.

Interval: sampling interval in seconds.

Rising Thr: rising threshold.

Falling Thr: falling threshold.

Rising Event ID: rising event identifier.

Falling Event ID: falling event identifier.

Rising Alarms: number of rising alarms that have been generated since device start up, or since the last time the statistics for this alarm were cleared.

Falling Alarms: number of falling alarms that have been generated since device start up, or since the last time the statistics for this alarm were cleared.

Total: total number of alarms that have been generated since device start up, or since the last time the statistics for this alarm were cleared.

Last Alarm: time elapsed since the last alarm was generated.

Max (delta|absolute) value: maximum value that the monitored variable has reached. Depending on the type of sampling, this can be either the absolute or the delta value from the previous period.

Min (delta|absolute) value: minimum value that the monitored variable has reached. Depending on the type of sampling, this can be either the absolute or the delta value from the previous period.

Example:

```
RMON+list alarm 1

 Alarm ID: 1
        Variable: .1.3.6.1.4.1.2007.4.1.2.2.2.21.1.1.7.1.1
        Owner: palonso
        Status: valid             Sample Type: delta         Interval: 10s
        Rising Thr: 1000          Falling Thr: 200
        Rising Event ID: 11       Falling Event ID: 12
        Rising Alarms: 0          Falling Alarms: 1          Total: 1
        Last Alarm: 5d21h29m57s
        Max delta value: 1        Min delta value: 0

RMON+
```

3.2.3.3 list alarm all

This option is similar to the previous one, but displays the information on all the alarms present in the device in consecutive order.

3.2.3.4 list event-table

This option displays the content of the events table. The meaning of each column in the table is as follows:

ID: event identifier.

Status: event status.

Description: comment describing the event.

Type: type of event.

Community/User: if the event generates SNMP *traps*, this is the name of the community (when using SNMPv1/v2c) or user (with SNMPv3) the SNMP *traps* are sent to.

Last: system time at the point when the event was last triggered.

Owner: name of the event owner.

Example:

```
RMON+list event-table
```

```
Event Table information:
 ID  Status  Description                          Type          Community/User        Last       Owner

-------------------------------------------------------------------------------------------------------------
  3 valid   Event due to high incoming traffic    log&trap        RMON-RPD        5d15h1m57s     palonso
  4 valid   Event due to low incoming traffic     log                             5d15h0m17s     palonso
  5 valid   Event due to high outgoing traffic    trap            RMON-RPD        3d21h22m21s    palonso
  6 valid   Event due to low outgoing traffic     none                            3d21h22m41s    palonso
  11 valid  Potential broadcast storm detected    log&trap        RMON-RPD        0s             palonso
  12 valid  End of broadcast storm                log&trap        RMON-RPD        10s            palonso

RMON+
```

Command history:

Version	Modification
10.09.24.20.08	The command output has been modified as of version 10.09.24.20.08 to show the *Trap user* (since RMON traps may be sent using SNMPv3).
10.09.26	The command output has been modified as of version 10.9.26 to show the *Trap user* (since RMON traps may be sent using SNMPv3).
11.00.05	The command output has been modified as of version 11.00.05 to show the *Trap user* (since RMON traps may be sent using SNMPv3).
11.01.00	The command output has been modified as of version 11.01.00 to show the *Trap user* (since RMON traps may be sent using SNMPv3).

3.2.3.5 list event <id>

This command displays the data and statistics for an event selected through its identifier.

The meaning of each line is as follows:

Event ID: event identifier.

Status: event status.

Description: comment describing this event.

Type: type of event.

Community/User: if the event generates SNMP *traps*, this is the name of the community (when using SNMPv1/v2) or user (when using SNMPv3) the SNMP *traps* are sent to.

Traps sent: number of SNMP *traps* sent by the event since device start up or since the last time the statistics for this event were cleared.

Logs stored: number of *logs* saved in the *logs* table since device start up, or since the last time the statistics for this event were cleared.

Last: time elapsed since the event was last triggered.

Stored logs: displays the content of the *logs* table for this event. Please see the *list log-table* command to see the meaning of each of the columns in the table.

Example:

```
RMON+list event 12

 Event ID: 12
        Status: valid            Description: "End of broadcast storm"
        Type: log&trap           Community/User: RMON-RPD
        Traps sent: 1            Logs stored: 1          Last: 5d15h4m10s
        Stored Logs:
        EventID/LogID   Time     Description (Dir: OID = value <=> threshold : alarmID, eventID)
        ------------------------------------------------------------------------------------------------
        12/1            10s      "Falling:.1.3.6.1.4.1.2007.4.1.2.2.2.21.1.1.7.1.1 =
                                                                                  0 <= 200 : 1, 12"

RMON+
```

Command history:

Version	Modification
10.09.24.20.08	The command output has been modified as of version 10.09.24.20.08 to show the *Trap user* (since RMON traps may be sent using SNMPv3).
10.09.26	The command output has been modified as of version 10.9.26 to show the *Trap user* (since RMON traps may be sent using SNMPv3).
11.00.05	The command output has been modified as of version 11.00.05 to show the *Trap user* (since RMON traps may be sent using SNMPv3).
11.01.00	The command output has been modified as of version 11.01.00 to show the *Trap user* (since RMON traps may be sent using SNMPv3).

3.2.3.6 list event all

This option is similar to the previous one, but displays the information on all the events present in the device in consecutive order.

3.2.3.7 list log-table

This command displays the content of the *logs* table for all the events. The meaning of each column in the table is as follows:

EventID/LogID: Event identifier duple and *log* identifier.

Time: system time elapsed at the point when the *log* is generated.

Description: comment describing this *log*.

The meaning of the *log* description is as follows:

Direction: This can be *rising* or *falling,* depending on whether this is a rising or falling event.

OID: This is the identifier for the variable object that has generated the event.

value: Value of the variable when generating the event (it is the delta value or the absolute value, depending on the type of sampling configured in the alarm).

<=>: it is ">=" in a rising event and "<=" in a falling event.

threshold: This is the value of the threshold that has been reached.

alarmID: This is the identifier for the alarm associated to the event.

eventID: This is the identifier for the event that has generated the *log*.

Example:

```
RMON+list log-table
 Log Table information:
    EventID/LogID       Time                 Description (Dir: OID = value <=> threshold : alarmID, eventID)
-------------------------------------------------------------------------------------------------------------
          3/1           4d10h17m34s          "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 2367 >= 2000 : 2, 3"
          3/2           4d10h22m54s          "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 2120 >= 2000 : 2, 3"
          3/3           4d10h38m4s           "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 2128 >= 2000 : 2, 3"
          3/4           4d10h48m14s          "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 2014 >= 2000 : 2, 3"
          3/5           4d10h59m34s          "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 2046 >= 2000 : 2, 3"
          3/6           4d11h22m54s          "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 2093 >= 2000 : 2, 3"
          3/7           4d11h53m44s          "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 2011 >= 2000 : 2, 3"
          3/8           4d12h10m14s          "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 2201 >= 2000 : 2, 3"
          3/9           4d12h33m14s          "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 2159 >= 2000 : 2, 3"
          3/10          4d12h38m4s           "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 2176 >= 2000 : 2, 3"
          3/11          4d12h40m34s          "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 2490 >= 2000 : 2, 3"
          3/12          4d12h53m44s          "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 2113 >= 2000 : 2, 3"
          3/13          4d12h59m34s          "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 2903 >= 2000 : 2, 3"
          3/14          4d13h2m4s            "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 2680 >= 2000 : 2, 3"
          3/15          4d13h4m4s            "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 2072 >= 2000 : 2, 3"
          3/16          4d13h7m24s           "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 2153 >= 2000 : 2, 3"
          3/17          4d13h17m14s          "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 2168 >= 2000 : 2, 3"
```

```
          3/18          4d13h21m34s          "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 2079 >= 2000 : 2, 3"
          3/19          4d13h26m4s           "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 2280 >= 2000 : 2, 3"
          3/20          4d13h36m14s          "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 3487 >= 2000 : 2, 3"
          3/21          4d13h51m24s          "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 2274 >= 2000 : 2, 3"
          3/22          4d14h1m34s           "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 2367 >= 2000 : 2, 3"
          3/23          4d14h2m4s            "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 2193 >= 2000 : 2, 3"
          3/24          4d14h21m34s          "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 4547 >= 2000 : 2, 3"
          3/25          4d14h46m54s          "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 2108 >= 2000 : 2, 3"
          3/26          4d15h2m14s           "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 2254 >= 2000 : 2, 3"
          3/27          4d15h6m34s           "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 3262 >= 2000 : 2, 3"
          3/28          4d15h26m4s           "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 3766 >= 2000 : 2, 3"
          3/29          4d15h38m44s          "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 4696 >= 2000 : 2, 3"
          3/30          4d15h51m24s          "Rising:.1.3.6.1.2.1.2.2.1.10.1 = 2021 >= 2000 : 2, 3"
          4/1           4d10h19m14s          "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 70 <= 100 : 2, 4"
          4/2           4d10h37m14s          "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 70 <= 100 : 2, 4"
          4/3           4d10h40m14s          "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 70 <= 100 : 2, 4"
          4/4           4d10h59m14s          "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 70 <= 100 : 2, 4"
          4/5           4d11h19m14s          "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 70 <= 100 : 2, 4"
          4/6           4d11h45m44s          "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 70 <= 100 : 2, 4"
          4/7           4d12h0m44s           "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 70 <= 100 : 2, 4"
          4/8           4d12h24m14s          "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 70 <= 100 : 2, 4"
          4/9           4d12h34m24s          "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 0 <= 100 : 2, 4"
          4/10          4d12h38m54s          "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 0 <= 100 : 2, 4"
          4/11          4d12h41m14s          "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 70 <= 100 : 2, 4"
          4/12          4d12h55m24s          "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 0 <= 100 : 2, 4"
          4/13          4d13h1m24s           "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 64 <= 100 : 2, 4"
          4/14          4d13h3m44s           "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 70 <= 100 : 2, 4"
          4/15          4d13h4m44s           "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 70 <= 100 : 2, 4"
          4/16          4d13h13m54s          "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 64 <= 100 : 2, 4"
          4/17          4d13h19m24s          "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 64 <= 100 : 2, 4"
          4/18          4d13h21m54s          "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 0 <= 100 : 2, 4"
          4/19          4d13h33m54s          "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 0 <= 100 : 2, 4"
          4/20          4d13h39m54s          "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 0 <= 100 : 2, 4"
          4/21          4d13h59m54s          "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 64 <= 100 : 2, 4"
          4/22          4d14h1m54s           "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 64 <= 100 : 2, 4"
          4/23          4d14h18m54s          "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 0 <= 100 : 2, 4"
          4/24          4d14h39m54s          "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 0 <= 100 : 2, 4"
          4/25          4d14h57m24s          "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 0 <= 100 : 2, 4"
          4/26          4d15h3m54s           "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 0 <= 100 : 2, 4"
          4/27          4d15h18m54s          "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 0 <= 100 : 2, 4"
          4/28          4d15h38m24s          "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 0 <= 100 : 2, 4"
          4/29          4d15h45m24s          "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 0 <= 100 : 2, 4"
          4/30          4d15h57m54s          "Falling:.1.3.6.1.2.1.2.2.1.10.1 = 64 <= 100 : 2, 4"
         12/1           10s                  "Falling:.1.3.6.1.4.1.2007
                                             .4.1.2.2.2.21.1.1.7.1.1 = 0 <= 200 : 1,12"

       Current max table size: 30
       System uptime: 5d15h4m28s

RMON +
```

3.2.3.8 list statistics

This command displays the RMON global statistics. The meaning of each line is as follows:

Alarms fired: total number of alarms that have been generated since device start up, or since the last time the global statistics were cleared.

Rising: total number of rising alarms that have been generated since device start up, or since the last time the global statistics were cleared.

Falling: total number of falling alarms that have been generated since device start up, or since the last time the global statistics were cleared.

Last: time elapsed since the last alarm was generated.

Traps sent: total number of SNMP *traps* sent since device start up, or since the last time the global statistics were cleared.

Last: time elapsed since the last SNMP *trap* was sent.

Logs stored: total number of *logs* stored in the *logs* table since device start up, or since the last time the global statistics were cleared.

Last: time elapsed since the last *log* was stored.

System uptime: time elapsed since the last device start up.

Example:

```
RMON+list statistics
 RMON statistics:
        Alarms fired: 8749       Rising: 4374     Falling: 4375    Last: 59s
        Traps sent: 4375         Last: 59s
        Logs stored: 1536        Last: 59s
        System uptime: 5d15h12m17s
RMON+
```

Chapter 4 Configuration Example

4.1 Configuring an alarm and two RMON events

As an example, we are going to configure an alarm and two RMON events to detect a broadcast storm in port 1 of a switch device.

The first thing we need to know when configuring an alarm is the object identifier (OID) that belongs to the variable we want to monitor. In this case, we're going to monitor the *teldatSwitchOutBroadTraffic* variable so that port 1 of the switch device has the following OID 1.3.6.1.4.1.2007.4.1.2.2.2.21.1.1.7.1.1. We then need to select the rising and falling thresholds for a broadcast storm. For the purposes of this example, we shall configure a rising threshold of 10,000 packets and a falling threshold of 200 packets in a 10 second interval. As for events, the rising event will generate *log* and *trap*, and the falling event will only generate *log*.

With this data, the configuration looks like this:

```
   log-command-errors
   no configuration
   set hostname RMON-EJEMPLO
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.212.133 255.255.254.0
;
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      management-ip-address 192.168.212.133
      classless
;
   exit
;
;
   protocol snmp
; -- SNMP user configuration --
      engineid local 23456789
      user palonso
;
;
      community RMON-COMM access write-read-trap
      community RMON-COMM subnet 192.168.212.14 255.255.255.255
;
      host 192.168.212.14 trap version v1 RMON-COMM all
;
      trap sending-parameters time 0s
      trap sending-parameters number 1
      trap sending-parameters reachability-checking ip-route
   exit
;
   feature rmon
; -- Remote Network Monitoring configuration --
      alarm 1 10 1.3.6.1.4.1.2007.4.1.2.2.2.21.1.1.7.1.1 delta rising-threshold 1000 event 1
falling-threshold 200 event 2 owner palonso
;
      event 1 log trap RMON-COMM description "Potential broadcast storm detected" owner palonso
      event 2 log description "End of broadcast storm" owner palonso
;
      log-table-size 30
   exit
;
   dump-command-errors
   end
```