

Fault Management Northbound API

Cloud Execution Environment

INTERWORK DESCRIPTION

Copyright

© Ericsson AB 2016–2018. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



Contents

1	Introduction	1
1.1	Prerequisites	1
2	Overview of API Components	2
3	CLI	4
3.1	Active Alarm List	4
3.2	Alarm and Alert History	5
3.3	Adding SNMP Trap Endpoint	8
3.4	Listing SNMP Trap Endpoints	9
3.5	Removing SNMP Trap Endpoint	10
4	FM Northbound REST API	13
4.1	Get Active Alarm List	13
4.2	Get Alarm and Alert History	14
4.3	Add SNMP Trap Endpoint	16
4.4	Get SNMP Trap Endpoints	16
4.5	Remove SNMP Trap Endpoint	17
5	SNMP Interface	18
5.1	SNMP - Active Alarm List	18
5.2	SNMP Trap	22
6	Additional Information	23
	Reference List	25





1 Introduction

This document describes the Fault Management (FM) Northbound (NB) Application Programming Interface (API) used in the Cloud Execution Environment (CEE).

The FM framework enables CEE components to request human intervention in case of fault situations that cannot be automatically recovered. It provides an active alarm list and a northbound interface for exposing alarms and alerts for tenants, cloud infrastructure providers, and for telecommunications and IT providers.

FM also enables converting stateless IT-type notifications to stateful telecommunications-type alarms for components and third party equipment in case alarm state information is provided by those devices.

The central FM component in CEE collects alarms and alerts from devices operating in different main components of the system.

1.1 Prerequisites

The user must have the watchmen role in Keystone and must be authenticated by getting a V2 Keystone authorization token.



2 Overview of API Components

This section provides an overview of the FM NB API components.

FM NB API Functions

The following functions are available:

- Active alarm list
- Alarm and alert history log
- Sending Simple Network Management Protocol (SNMP) traps including heartbeat
- Adding SNMP trap receiver endpoints
- Listing SNMP trap receiver endpoints
- Removing SNMP trap receiver endpoints

FM NB API Interfaces

The following interfaces are used:

- CLI
See Section 3.1 on page 4.
- Representational State Transfer (REST) API
See Section 4 on page 13.
- SNMP
See Section 5 on page 18.

Not all the functions are supported on each interface. Some functions are available at more than one interfaces. Table 1 shows which function is available at which interfaces. For the detailed descriptions, see the sections referred in the table.



Table 1 API Components

FUNCTIONS	INTERFACES		
	CLI	REST API	SNMP
Active alarm list	Section 3.1 on page 4	Section 4.1 on page 13	Section 5.1 on page 18
Alarm and alert history log	Section 3.2 on page 5	Section 4.2 on page 14	-
Sending SNMP traps including heartbeat	-	-	Section 5.2 on page 22
Adding SNMP trap receiver endpoint	Section 3.3 on page 8	Section 4.3 on page 16	-
Listing SNMP trap receiver endpoint	Section 3.4 on page 9	Section 4.4 on page 16	-
Removing SNMP trap receiver endpoint	Section 3.5 on page 10	Section 4.5 on page 17	-



3 CLI

The FM CLI can be reached from the virtual Cloud Infrastructure Controllers (vCICs) and from the Compute Nodes by using a terminal.

Note: The `watchmen-client` command can also be used for configuring SNMP. For more information, refer to the [Fault Management Configuration Guide](#).

Watchmen supports bash completion.

The following operations are supported at the CLI:

- Fetching the active alarm list
See Section 3.1 on page 4.
- Fetching the alarm and alert history
See Section 3.2 on page 5.
- Adding an SNMP trap endpoint
See Section 3.3 on page 8.
- Listing SNMP trap endpoints
See Section 3.4 on page 9.
- Removing an SNMP trap endpoint
See Section 3.5 on page 10.

3.1 Active Alarm List

The active alarm list can be fetched by using the `watchmen-client` command with the `active-alarm-list` subcommand. The active alarm list is filtered by the ID of the tenant that executes the `watchmen-client` command. No filtering is applied if the command is executed by the admin tenant.

3.1.1 Command Description

3.1.1.1 Syntax

```
watchmen-client [--os-username <OS_Username>]
                [--os-password <OS_Password>]
                [--os-tenant-name <OS_Tenant_Name>]
                [--os-auth-url <OS_Auth_URL>]
                active-alarm-list

watchmen-client [-h]
watchmen-client [--help]
```




```
watchmen-client active-alarm-list [-h] [-tz "<time-zone>"]
watchmen-client active-alarm-list [--help] [--time-zone =>
"<time-zone>"]
```

3.1.1.2

Optional Arguments

This section describes the optional arguments that can be used with the commands.

Used either with the main command `watchmen-client` **or the sub-command** `active-alarm-list`:

-h, --help	Displays information about the <code>watchmen-client</code> command or the <code>active-alarm-list</code> subcommand.
-------------------	---

Used with the main command `watchmen-client`:

--os-username	Specifies the OS user name. It can be any user that has <code>watchmen</code> role.
--os-password	Specifies the OS password.
--os-tenant-name	Specifies the OS tenant name.
--os-auth-url	Specifies the OS authentication URL.

Used with the sub-command `active-alarm-list`:

-tz, --time-zone	Displays events with specified time-zone. Default time-zone: UTC
-------------------------	--

The `--os-username`, `--os-password`, `--os-tenant-name`, and `--os-auth-url` must be specified in the command each time they are required by a subcommand.

Note: Watchmen CLI also supports the `OS_Username`, `OS_Password`, `OS_Tenant_Name`, and `OS_Auth_URL` OpenStack environment variables.

3.1.1.3

Example

```
root@cic-2:~# watchmen-client --os-username admin --os-password admin =>
--os-tenant-name admin --os-auth-url http://192.168.2.31:5000/v2.0 =>
active-alarm-list --time-zone "Europe/Stockholm"
```

3.2

Alarm and Alert History

The alarm and alert history can be fetched by using the `watchmen-client` command with the `alarm-history` subcommand.



The alarm and alert history is filtered by the ID of the tenant that executes the `watchmen-client` command. No filtering is applied if the command is executed by the admin tenant.

3.2.1 Command Description

3.2.1.1 Syntax

```
watchmen-client [--os-username <OS_Username>]
                [--os-password <OS_Password>]
                [--os-tenant-name <OS_Tenant_Name>]
                [--os-auth-url <OS_Auth_URL>]
                alarm-history
```

```
watchmen-client [-h]
watchmen-client [--help]
```

```
watchmen-client alarm-history [-h] [-f YYYY-MM-DD[[-hh]-mm]-ss]] =>
[-t YYYY-MM-DD[[-hh]-mm]-ss]] [-s] [-o {asc,desc}] [-e {alarm,alert}] =>
[-tz "<time-zone>"]=>
watchmen-client alarm-history [--help] [--from YYYY-MM-DD[[-hh]-mm]-ss]] =>
[--to YYYY-MM-DD[[-hh]-mm]-ss]] [--sort-by] [FIELD_NAME] =>
[--sort-order {asc,desc}] [--event-type {alarm,alert}] =>
[--time-zone "<time-zone>"]
```

Where one of the following values is used for `FIELD_NAME`:
`active_severity`, `additional_info`, `additional_text`, `event_type`,
`is_stateful`, `last_event_time`, `major_type`, `minor_type`, `probable_cause`,
`sequence_no`, `source`, `specific_problem`.
Default value: `last_event_time`

Note: The date and time values used in the `--from` and `--to` (`-f` and `-t`) sections of the command, define a half-open interval. The `--from` date and time is included in the interval. The `--to` date and time is excluded.

3.2.1.2 Optional Arguments

This section describes the optional arguments that can be used with the commands.

Used either with the main command `watchmen-client` or the sub-command `alarm-history`:

-h, --help	Displays information about the <code>watchmen-client</code> command or the <code>alarm-history</code> subcommand.
-------------------	---

Used with the main command `watchmen-client`:



--os-username	Specifies the OS user name. It can be any user that has watchmen role.
--os-password	Specifies the OS password.
--os-tenant-name	Specifies the OS tenant name.
--os-auth-url	Specifies the OS authentication URL.

Used with the sub-command `alarm-history`:

-f, --from	Defines the beginning of the time period of the request. This optional argument can be used separately or together with <code><-t, --to></code> . If none of them is given, the output shows the event history of the actual day.
-t, --to	Defines the end of the time period of the request. This optional argument can be used separately or together with <code><-f, --from></code> . If none of them is given, the output shows the event history of the actual day.
-s, --sort-by	Sort by a given column. Supported values are the ones listed for <code>FIELD_NAME</code> in Section 3.2.1.1 on page 6.
-o, --sort-order	Sort order. Supported values: <code>asc</code> for ascending and <code>desc</code> for descending order.
-e, --event-type	Filters events, either for alarms or alerts. Supported values: <code>alarm</code> and <code>alert</code>
-tz, --time-zone	Displays events with specified time-zone. Default time-zone: UTC

Note: The `--os-username`, `--os-password`, `--os-tenant-name`, and `--os-auth-url` must be specified in the command each time they are required by a subcommand.

Watchmen CLI also supports the `OS_Username`, `OS_Password`, `OS_Tenant_Name`, and `OS_Auth_URL` OpenStack environment variables.

3.2.1.3 Example

```
root@cic1:~# watchmen-client --os-username admin --os-password admin ⇒
--os-tenant-name admin --os-auth-url http://192.168.2.25:5000/v2.0 ⇒
alarm-history --from 2016-01-25 --to 2016-02-01 --sort-by active_severity ⇒
--sort-order desc --event-type alert -tz "Europe/Stockholm"
```



3.3 Adding SNMP Trap Endpoint

An SNMP trap endpoint can be added by using the `watchmen-client` command with the `snmp-trap-config-add` subcommand.

The events sent to the configured SNMP trap endpoint are filtered by the ID of the tenant that executes the `watchmen-client` command. No filtering is applied if the command is executed by the admin tenant.

3.3.1 Command Description

3.3.1.1 Syntax

```
watchmen-client [--os-username <OS_Username>]
                [--os-password <OS_Password>]
                [--os-tenant-name <OS_Tenant_Name>]
                [--os-auth-url <OS_Auth_URL>]
                snmp-trap-config-add
```

```
watchmen-client [-h]
watchmen-client [--help]
```

```
watchmen-client snmp-trap-config-add [-h] -c <command_string> [-e]
watchmen-client snmp-trap-config-add [--help] ⇒
--command <command_string> [--enable-append-info]
```

3.3.1.2 Required Arguments

The following argument is required for the subcommand `snmp-trap-config-add`:

-c, --command	SNMP trap command. For more information, refer to the Fault Management Configuration Guide.
----------------------	---

3.3.1.3 Optional Arguments

This section describes the optional arguments that can be used with the commands.

Used either with the main command `watchmen-client` or the sub-command `snmp-trap-config-add`:

-h, --help	Displays information about the <code>watchmen-client</code> command or the <code>snmp-trap-config-add</code> subcommand.
-------------------	--



Used with the main command `watchmen-client`:

<code>--os-username</code>	Specifies the OS user name. It can be any user that has watchmen role.
<code>--os-password</code>	Specifies the OS password.
<code>--os-tenant-name</code>	Specifies the OS tenant name.
<code>--os-auth-url</code>	Specifies the OS authentication URL.

Used with the sub-command `snmp-trap-config-add`:

<code>-e, --enable-append-info</code>	Enable SNMP Appendinfo Trap
---------------------------------------	-----------------------------

3.3.1.4 Example

```
root@cic1:~# watchmen-client --os-username admin --os-password admin ⇒
--os-tenant-name admin --os-auth-url http://192.168.2.25:5000/v2.0 ⇒
snmp-trap-config-add --command <SNMP_TRAP_COMMAND>
```

3.4 Listing SNMP Trap Endpoints

The SNMP trap endpoints can be listed by using the `watchmen-client` command with the `snmp-trap-config-list` subcommand.

The SNMP trap endpoints are filtered by the ID of the tenant that executes the `watchmen-client` command. No filtering is applied if the command is executed by the admin tenant.

3.4.1 Command Description

3.4.1.1 Syntax

```
watchmen-client [--os-username <OS_Username>]
                [--os-password <OS_Password>]
                [--os-tenant-name <OS_Tenant_Name>]
                [--os-auth-url <OS_Auth_URL>]
                snmp-trap-config-list
```

```
watchmen-client [-h]
watchmen-client [--help]
```

```
watchmen-client snmp-trap-config-list [-h]
watchmen-client snmp-trap-config-list [--help]
```



3.4.1.2 Optional Arguments

This section describes the optional arguments can be used with the commands.

Used either with the main command `watchmen-client` or the sub-command `snmp-trap-config-list`:

-h, --help	Displays information about the <code>watchmen-client</code> command or the <code>snmp-trap-config-list</code> subcommand.
-------------------	---

Used with the main command `watchmen-client`:

--os-username	Specifies the OS user name. It can be any user that has watchmen role.
--os-password	Specifies the OS password.
--os-tenant-name	Specifies the OS tenant name.
--os-auth-url	Specifies the OS authentication URL.

The `--os-username`, `--os-password`, `--os-tenant-name`, and `--os-auth-url` must be specified in the command each time they are required by a subcommand.

Note: Watchmen CLI also supports the `OS_Username`, `OS_Password`, `OS_Tenant_Name`, and `OS_Auth_URL` OpenStack environment variables.

3.4.1.3 Example

```
root@cic1:~# watchmen-client --os-username admin --os-password admin =>
--os-tenant-name admin --os-auth-url http://192.168.2.25:5000/v2.0 =>
snmp-trap-config-list
```

3.5 Removing SNMP Trap Endpoint

An SNMP endpoint can be removed by using the `watchmen-client` command with the `snmp-trap-config-remove` subcommand.

The tenant executing the `watchmen-client` command can only remove those SNMP trap endpoints that were added by the same tenant. There is no such a restriction if the command is executed by the admin tenant.

3.5.1 Command Description

3.5.1.1 Syntax

```
watchmen-client [--os-username <OS_Username>]
```



```

[--os-password <OS_Password>]
[--os-tenant-name <OS_Tenant_Name>]
[--os-auth-url <OS_Auth_URL>]
snmp-trap-config-remove

watchmen-client [-h]
watchmen-client [--help]

watchmen-client snmp-trap-config-remove [-h] -id <id_of_endpoint>
watchmen-client snmp-trap-config-remove [--help] --config-id =>
<id_of_endpoint>

```

3.5.1.2 Required Arguments

The following argument is required for the subcommand `snmp-trap-config-add`:

-id, --config-id	ID of the SNMP trap endpoint configuration to be removed
-------------------------	--

3.5.1.3 Optional Arguments

This section describes the optional arguments that can be used with the commands.

Used either with the main command `watchmen-client` **or the sub-command** `snmp-trap-config-remove`:

-h, --help	Displays information about the <code>watchmen-client</code> command or the <code>snmp-trap-config-remove</code> subcommand.
-------------------	---

Used with the main command `watchmen-client`:

--os-username	Specifies the OS user name. It can be any user that has watchmen role.
--os-password	Specifies the OS password.
--os-tenant-name	Specifies the OS tenant name.
--os-auth-url	Specifies the OS authentication URL.

The `--os-username`, `--os-password`, `--os-tenant-name`, and `--os-auth-url` must be specified in the command each time they are required by a subcommand.

Note: Watchmen CLI also supports the `OS_Username`, `OS_Password`, `OS_Tenant_Name`, and `OS_Auth_URL` OpenStack environment variables.



3.5.1.4 Example

```
root@cic1:~# watchmen-client --os-username admin --os-password admin =>  
--os-tenant-name admin --os-auth-url http://192.168.2.25:5000/v2.0 =>  
snmp-trap-config-remove --config-id 16
```




4 FM Northbound REST API

The FM Northbound REST API provides the northbound interface toward the used management system.

The FM Web Server Gateway Interface (WSGI) Server is listening on the 8052 port of the vCICs. The FM WSGI Server is running in active-active mode, and it can be reached at the vCIC HA address on port 8052.

FM RESTful port: 8052

Required user role in Keystone: watchmen

The following operations are supported:

- Get active alarm list
See Section 4.1 on page 13.
- Get alarm and alert history
See Section 4.2 on page 14.
- Add SNMP Trap Endpoint
See Section 4.3 on page 16.
- Get SNMP Trap Endpoints
See Section 4.4 on page 16.
- Remove SNMP Trap Endpoint
See Section 4.5 on page 17.

Note: For configuring SNMP Trap Endpoint, refer to the [Fault Management Configuration Guide](#).

4.1 Get Active Alarm List

Method: GET

Headers:

- X-Auth-Token: <keystone_v2_auth_token_id>
- X-Tenant-Name: <tenant_name_for_the_token>

Uniform Resource Locator (URL): /v1/active_alarm_list

Description: Returns the alarms currently active in FM. The alarms are filtered by the ID of the tenant set in the request. No filter on tenant ID is used if the admin tenant is set in the request.

Normal response code: 200



Standard Hypertext Transfer Protocol (HTTP) error codes used:
401, 403, 500

Parameters: -

JavaScript Object Notation (JSON) Response Example:

```
{
  "active_alarm_list": [
    {
      "active_severity": 5,
      "additional_info": "None",
      "additional_text": "NTP error",
      "event_type": 1,
      "is_stateful": true,
      "last_event_time": "2016-11-22 14:00:05+00:00",
      "major_type": 193,
      "minor_type": 2031708,
      "probable_cause": 70,
      "sequence_no": 11,
      "source": "Region=SAPC_1,CeeFunction=1,Node=cic-0-1⇒
,UpstreamNTPServerConnection=1",
      "specific_problem": "NTP Stratum Level Failure"
    }
  ]
}
```

4.2 Get Alarm and Alert History

Method: GET

Headers:

- X-Auth-Token:<keystone_v2_auth_token_id>
- X-Tenant-Name:<tenant_name_for_the_token>

Supported URLs:

/v1/alarm_history

v1/alarm_history/from/<date>

v1/alarm_history/to/<date>

v1/alarm_history/sort_by/<FIELD_NAME>

See the possible values for FIELD_NAME in Section 3.2.1.1 on page 6.

v1/alarm_history/sort_order/<order>

The possible values for order are asc and desc.

v1/alarm_history/page_number/<number>



v1/alarm_history/page_size/<number>

v1/alarm_history/event_type/<event_type>

The possible values for event_type are alarm and alert.

The supported URLs can be combined in any way that makes sense, like in the following example:

v1/alarm_history/sort_by/sequence_no/sort_order/desc⇒
/page_size/4/event_type/alert/to/2016-11-22-14-13

Description: The alarm_history returns all the alarms and alerts received on a given day. The other URLs return the subset of alarms and alerts specified by the given filter. Furthermore, the alarms and alerts are also filtered by the ID of the tenant set in the request. No filter on tenant ID is used if the admin tenant is set in the request.

Normal response code: 200

Standard HTTP error codes used: 401, 403, 404, 500

Optional Parameter: date yyyy-mm-dd

Note: If the date parameter is not specified, the alarm and alert history of the actual day will be returned.

JSON Response Example:

```
{
  "alarm_history": [
    {
      "active_severity": 1,
      "additional_info": "info",
      "additional_text": "original text",
      "event_type": 2,
      "is_stateful": true,
      "last_event_time": "2016-11-22 14:05:35+00:00",
      "major_type": 193,
      "minor_type": 2031686,
      "probable_cause": 158,
      "sequence_no": 40,
      "source": "Region=SAPC_1,Equipment=1,TopOfRackSwitch=1⇒
,PowerSupply=9NJZMD0D",
      "specific_problem": "Test event in WFT"
    }
  ]
}
```



4.3 Add SNMP Trap Endpoint

Method: POST

Headers:

- X-Auth-Token:<keystone_v2_auth_token_id>
- X-Tenant-Name:<tenant_name_for_the_token>

Data:

```
'{"snmp_trap_config": {"command": "<SNMP_TRAP_COMMAND>", =>
"appendinfo": <HAS_APPEND_INFO>}}'
```

Uniform Resource Locator (URL): /v1/snmp_trap_config

Description: Adds SNMP Trap Endpoint configuration. Events sent towards that endpoint are filtered by the ID of the tenant set in the request. No filter on tenant ID is used if the admin tenant is set in the request.

Normal response code: 200

Standard Hypertext Transfer Protocol (HTTP) error codes used:
401, 403, 500

Parameters: -

JavaScript Object Notation (JSON) Response Example:

```
{
  "STATUS": "OK"
}
```

4.4 Get SNMP Trap Endpoints

Method: GET

Headers:

- X-Auth-Token:<keystone_v2_auth_token_id>
- X-Tenant-Name:<tenant_name_for_the_token>

Uniform Resource Locator (URL): /v1/snmp_trap_config

Description: Gets all SNMP Trap Endpoint configurations. SNMP Trap Endpoint configurations are filtered by the ID of the tenant set in the request. No filter on tenant ID is used if the admin tenant is set in the request.

Normal response code: 200

Standard Hypertext Transfer Protocol (HTTP) error codes used:
401, 403, 500



Parameters: -

JavaScript Object Notation (JSON) Response Example:

```
{
  "snmp_trap_config": [
    {
      "appendinfo": "<HAS_APPEND_INFO>",
      "command": "<SNMP_TRAP_COMMAND>",
      "config_id": 3,
      "tenant_id": "<TENANT_ID>",
      "tenant_name": "<TENANT_NAME>"
    }
  ]
}
```

4.5 Remove SNMP Trap Endpoint

Method: DELETE

Headers:

- X-Auth-Token:<keystone_v2_auth_token_id>
- X-Tenant-Name:<tenant_name_for_the_token>

Uniform Resource Locator (URL):

/v1/snmp_trap_config/<snmp_trap_config_id>

Description: Deletes SNMP Trap Endpoint configuration with the specified ID. Deleting is only allowed for those entries that were created by the tenant set in the request. There is no such a restriction if the admin tenant is set in the request.

Normal response code: 200

Standard Hypertext Transfer Protocol (HTTP) error codes used:

401, 403, 500

Parameters: -

JavaScript Object Notation (JSON) Response Example:

```
{
  "STATUS": "OK"
}
```



5 SNMP Interface

The only supported SNMP version is SNMP v2.

SNMP is used for the following purposes at the FM NB Interface:

- Fetching the active alarm list, see Section 5.1 on page 18.
- Sending SNMP traps, see Section 5.2 on page 22.

See Section 6 on page 23 for an overview of the main system components reporting alarms and alerts.

5.1 SNMP - Active Alarm List

This section describes how to fetch the list of active alarms by using the SNMP Interface.

The FM Northbound SNMP agent is running in active-active mode, and it can be reached at the vCIC HA address on port 30165.

5.1.1 Supported PDUs

The active alarm list can be fetched by sending SNMP requests.

The FM Northbound SNMP Agent answers the following Protocol Data Units (PDUs):

- GetBulkRequest
- GetNextRequest

5.1.2 Command Examples

This section provides examples for commands that can be used for sending the PDUs:

- `snmptable`
See Section 5.1.2.1 on page 19.
- `snmpwalk`
See Section 5.1.2.2 on page 20.

These examples assume that the following conditions are met:



- Commands are executed on a host where the NB address (HAProxy IP address, that is, the public VIP of the vCIC) is reachable (for example, the vCIC hosts)
- The required MIBs are available on the Net-SNMP default MIB path (for example, /usr/share/snmp/mibs/ on the vCIC hosts)

Note: The application is part of the Net-SNMP package. For more information about configuration and usage, refer to the Net-SNMP documentation, Reference [1].

5.1.2.1 snmptable

The `snmptable` command sends a GetBulkRequest PDU.

Syntax

With the default agent configuration, the `snmptable` command can be used with the following syntax:

```
$ snmptable -v 2c -c traps -m ALL <public_vip>:⇒
30165 eriAlarmActiveAlarmTable
```

Response

If the active alarm list is empty, the command returns with the following text:

```
ERICSSON-ALARM-MIB::eriAlarmActiveAlarmTable: No entries
```

If the active alarm list contains alarms, the alarms are visible in the command response as shown in the example below:

SNMP table: ERICSSON-ALARM-MIB::eriAlarmActiveAlarmTable

```
eriAlarmActiveMajorType eriAlarmActiveMinorType eriAlarmActiveSpecificProblem eriAlarmActiveManagedObject⇒
eriAlarmActiveEventType eriAlarmActiveEventTime eriAlarmActiveOriginalEventTime⇒
eriAlarmActiveProbableCause eriAlarmActiveSeverity eriAlarmActiveOriginalSeverity⇒
eriAlarmActiveAdditionalText⇒ eriAlarmActiveOrigAdditionalText eriAlarmActiveResourceId
1 timeDomainViolation 1 2000-12-5,0:0:0.0 problem 2000-12-5,0:0:0.0 source⇒
critical critical i610LocEndToEnd⇒
SNMPv2-SMI::zeroDotZero text text⇒
2 timeDomainViolation 2 2000-12-5,0:0:0.0 problem 2000-12-5,0:0:0.0 source⇒
critical critical i610LocEndToEnd⇒
SNMPv2-SMI::zeroDotZero text text⇒
3 timeDomainViolation 3 2000-12-5,0:0:0.0 problem 2000-12-5,0:0:0.0 source⇒
critical critical i610LocEndToEnd⇒
SNMPv2-SMI::zeroDotZero text text⇒
4 timeDomainViolation 4 2000-12-5,0:0:0.0 problem 2000-12-5,0:0:0.0 source⇒
critical critical i610LocEndToEnd⇒
SNMPv2-SMI::zeroDotZero text text⇒
```



Limitations

The `snmptable` command sends a `GetBulkRequest` PDU, that was designed to fetch tabular data in a single request-response transaction. SNMP clients are usually configured with short timeout, that is, a few seconds, and only five retry attempts. The bigger the active alarm list, the longer it takes for the agent to respond even it performs only one database query. If the table contains approximately more than 120 rows, the client might time out and resend the request until the maximum number of retries is reached. It usually ends with the following result:

Timeout: No Response from <public_vip>:30165.

Note: Setting a higher timeout in the SNMP client can provide a solution.

5.1.2.2

`snmpwalk`

The `snmpwalk` command sends `GetNextRequest` PDUs.

Syntax

With the default agent configuration, the `snmpwalk` command can be used with the following syntax:

```
$ snmpwalk -v 2c -c traps -m ALL <public_vip>:⇒  
30165 eriAlarmActiveAlarmTable
```

Response

If the active alarm list is empty, the command returns with the following text:

```
ERICSSON-ALARM-MIB::eriAlarmActiveResourceId.0 = No more⇒  
variables left in this MIB View (It is past the end of⇒  
the MIB tree)
```

If the active alarm list contains alarms, the alarms are visible in the command response as shown in the example below:



```

ERICSSON-ALARM-MIB::eriAlarmActiveIndex.1 = Gauge32: 1
ERICSSON-ALARM-MIB::eriAlarmActiveIndex.2 = Gauge32: 2
ERICSSON-ALARM-MIB::eriAlarmActiveIndex.3 = Gauge32: 3
ERICSSON-ALARM-MIB::eriAlarmActiveIndex.4 = Gauge32: 4
ERICSSON-ALARM-MIB::eriAlarmActiveMajorType.1 = Gauge32: 1
ERICSSON-ALARM-MIB::eriAlarmActiveMajorType.2 = Gauge32: 2
ERICSSON-ALARM-MIB::eriAlarmActiveMajorType.3 = Gauge32: 3
ERICSSON-ALARM-MIB::eriAlarmActiveMajorType.4 = Gauge32: 4
ERICSSON-ALARM-MIB::eriAlarmActiveMinorType.1 = Gauge32: 1
ERICSSON-ALARM-MIB::eriAlarmActiveMinorType.2 = Gauge32: 2
ERICSSON-ALARM-MIB::eriAlarmActiveMinorType.3 = Gauge32: 3
ERICSSON-ALARM-MIB::eriAlarmActiveMinorType.4 = Gauge32: 4
ERICSSON-ALARM-MIB::eriAlarmActiveSpecificProblem.1 = STRING: problem
ERICSSON-ALARM-MIB::eriAlarmActiveSpecificProblem.2 = STRING: problem
ERICSSON-ALARM-MIB::eriAlarmActiveSpecificProblem.3 = STRING: problem
ERICSSON-ALARM-MIB::eriAlarmActiveSpecificProblem.4 = STRING: problem
ERICSSON-ALARM-MIB::eriAlarmActiveManagedObject.1 = STRING: source
ERICSSON-ALARM-MIB::eriAlarmActiveManagedObject.2 = STRING: source
ERICSSON-ALARM-MIB::eriAlarmActiveManagedObject.3 = STRING: source
ERICSSON-ALARM-MIB::eriAlarmActiveManagedObject.4 = STRING: source
ERICSSON-ALARM-MIB::eriAlarmActiveEventType.1 = INTEGER: timeDomainViolation(11)
ERICSSON-ALARM-MIB::eriAlarmActiveEventType.2 = INTEGER: timeDomainViolation(11)
ERICSSON-ALARM-MIB::eriAlarmActiveEventType.3 = INTEGER: timeDomainViolation(11)
ERICSSON-ALARM-MIB::eriAlarmActiveEventType.4 = INTEGER: timeDomainViolation(11)
ERICSSON-ALARM-MIB::eriAlarmActiveEventTime.1 = STRING: 2000-12-5,0:0:0.0
ERICSSON-ALARM-MIB::eriAlarmActiveEventTime.2 = STRING: 2000-12-5,0:0:0.0
ERICSSON-ALARM-MIB::eriAlarmActiveEventTime.3 = STRING: 2000-12-5,0:0:0.0
ERICSSON-ALARM-MIB::eriAlarmActiveEventTime.4 = STRING: 2000-12-5,0:0:0.0
ERICSSON-ALARM-MIB::eriAlarmActiveOriginalEventTime.1 = STRING: 2000-12-5,0:0:0.0
ERICSSON-ALARM-MIB::eriAlarmActiveOriginalEventTime.2 = STRING: 2000-12-5,0:0:0.0
ERICSSON-ALARM-MIB::eriAlarmActiveOriginalEventTime.3 = STRING: 2000-12-5,0:0:0.0
ERICSSON-ALARM-MIB::eriAlarmActiveOriginalEventTime.4 = STRING: 2000-12-5,0:0:0.0
ERICSSON-ALARM-MIB::eriAlarmActiveProbableCause.1 = INTEGER: i610LocEndToEnd(600)
ERICSSON-ALARM-MIB::eriAlarmActiveProbableCause.2 = INTEGER: i610LocEndToEnd(600)
ERICSSON-ALARM-MIB::eriAlarmActiveProbableCause.3 = INTEGER: i610LocEndToEnd(600)
ERICSSON-ALARM-MIB::eriAlarmActiveProbableCause.4 = INTEGER: i610LocEndToEnd(600)
ERICSSON-ALARM-MIB::eriAlarmActiveSeverity.1 = INTEGER: critical(3)
ERICSSON-ALARM-MIB::eriAlarmActiveSeverity.2 = INTEGER: critical(3)
ERICSSON-ALARM-MIB::eriAlarmActiveSeverity.3 = INTEGER: critical(3)
ERICSSON-ALARM-MIB::eriAlarmActiveSeverity.4 = INTEGER: critical(3)
ERICSSON-ALARM-MIB::eriAlarmActiveOriginalSeverity.1 = INTEGER: critical(3)
ERICSSON-ALARM-MIB::eriAlarmActiveOriginalSeverity.2 = INTEGER: critical(3)
ERICSSON-ALARM-MIB::eriAlarmActiveOriginalSeverity.3 = INTEGER: critical(3)
ERICSSON-ALARM-MIB::eriAlarmActiveOriginalSeverity.4 = INTEGER: critical(3)
ERICSSON-ALARM-MIB::eriAlarmActiveAdditionalText.1 = STRING: text
ERICSSON-ALARM-MIB::eriAlarmActiveAdditionalText.2 = STRING: text
ERICSSON-ALARM-MIB::eriAlarmActiveAdditionalText.3 = STRING: text
ERICSSON-ALARM-MIB::eriAlarmActiveAdditionalText.4 = STRING: text
ERICSSON-ALARM-MIB::eriAlarmActiveOrigAdditionalText.1 = STRING: text
ERICSSON-ALARM-MIB::eriAlarmActiveOrigAdditionalText.2 = STRING: text
ERICSSON-ALARM-MIB::eriAlarmActiveOrigAdditionalText.3 = STRING: text
ERICSSON-ALARM-MIB::eriAlarmActiveOrigAdditionalText.4 = STRING: text
ERICSSON-ALARM-MIB::eriAlarmActiveResourceId.1 = OID: SNMPv2-SMI::zeroDotZero
ERICSSON-ALARM-MIB::eriAlarmActiveResourceId.2 = OID: SNMPv2-SMI::zeroDotZero
ERICSSON-ALARM-MIB::eriAlarmActiveResourceId.3 = OID: SNMPv2-SMI::zeroDotZero
ERICSSON-ALARM-MIB::eriAlarmActiveResourceId.4 = OID: SNMPv2-SMI::zeroDotZero
ERICSSON-ALARM-MIB::eriAlarmActiveResourceId.4 = No more variables left in this MIB View (It is past the⇒
end of the MIB tree)

```

Limitations

The `snmpwalk` command walks a part of the management information base (MIB) tree by sending `GetNextRequest` PDUs. The response only contains a single varbind and a reference to the next object identifier (OID). From this, the client will know which OID to fetch in the next round. The conversation ends when there is no more next OID left. This way it does not matter how many rows are there in the active alarm list, only one OID instance is returned per transaction. Currently, the active alarm list contains 14 columns, so if there are 50 rows, then the number of request-response transactions is $14 \times 50 = 700$. Consequently, this



method generates higher load on the system and the network. The agent fetches data from the database only once, when the first OID is requested.

Concurrent client requests are not possible.

5.2 SNMP Trap

For each incoming alarm and alert, and for the updates of the active alarm list, an SNMP trap with a unique sequence number is sent on the FM Northbound interface to one or more Network Management Stations (NMSs). The FM alarm or alert is translated into SNMP trap in compliance with the Ericsson Alarm MIB. The Net-SNMP command `snmptrap` is used for generating traps.

Refer to the [Fault Management Configuration Guide](#) for more information.

5.2.1 Heartbeat

The FM service periodically sends `eriAlarmHeartBeatNotif` traps via SNMP.

This message contains the latest sequence numbers and receive times for alarms and alerts. These values are saved in `heartbeat_info`.

The default value for the frequency of the heartbeat is one minute.



6 Additional Information

Each alarm and alert text contains information about the triggering Managed Object (MO) class as shown in the following example:

```
Managed Object Class:⇒  
Region=<name_of_the_region>,CeeFunction=1,Node=<hostname_of_the_node>,⇒  
Network=<network>,Aggregator=<aggr>,EthernetPort=<port>
```

Figure 1 provides an overview of the main system components that send alarms and alerts to FM.

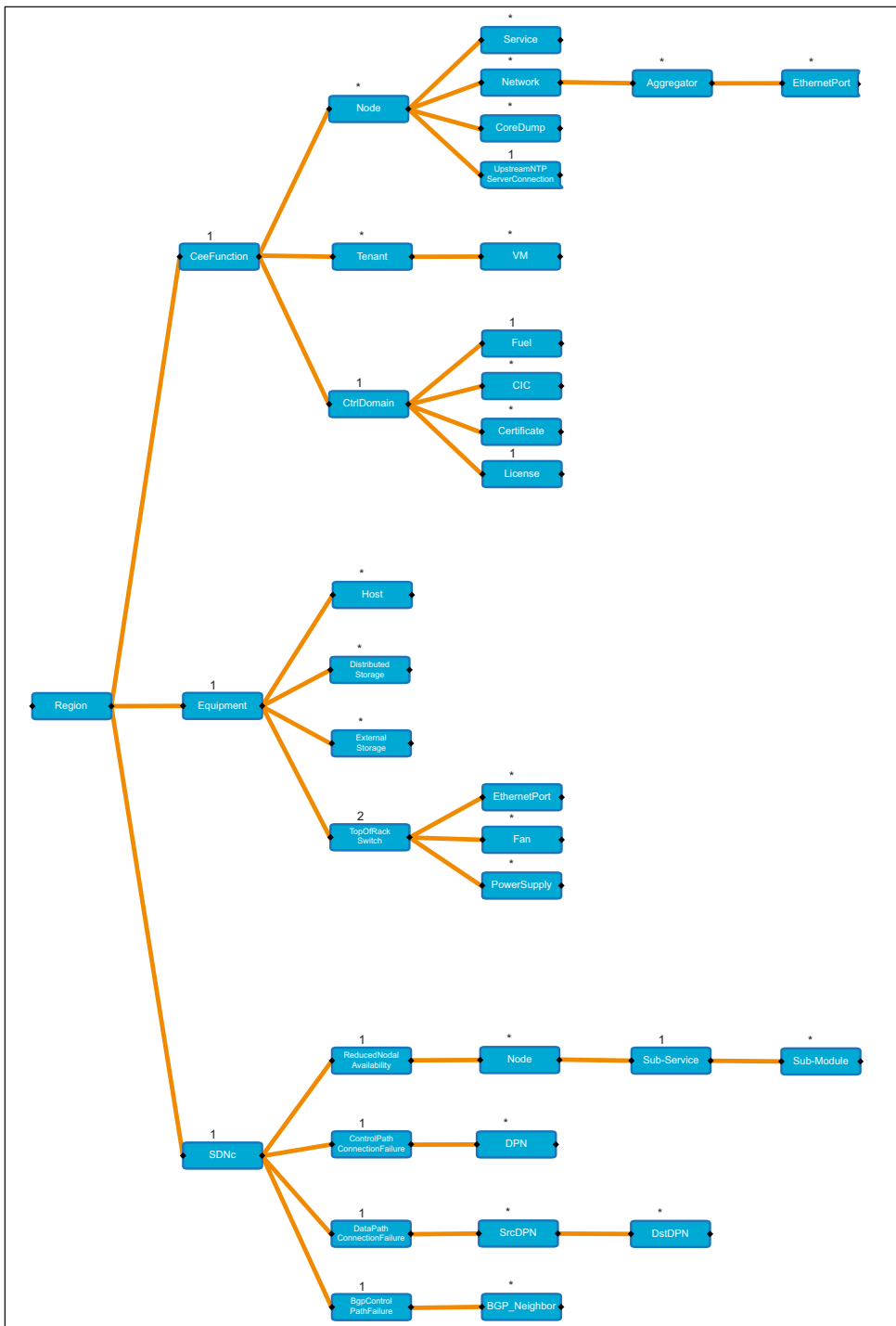


Figure 1 Main System Components Sending Alarms and Alerts to FM



Reference List

- [1] Net-SNMP website www.net-snmp.org