

# ENM FM SNMP Northbound Interface Integration Programmers Guide

User Instructions

## **Copyright**

© Ericsson AB 2017-2020. 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

<b>1</b>	<b>ENM FM SNMP Northbound Interface Integration Programmers Guide</b>	<b>1</b>
1.1	SNMP Versions Supported	2
1.2	MIB Supported	2
<b>2</b>	<b>Secure FM SNMP NBI User Roles</b>	<b>3</b>
<b>3</b>	<b>Initialize FM SNMP NBI</b>	<b>4</b>
<b>4</b>	<b>Spontaneous Alarms and Alerts Communication</b>	<b>5</b>
<b>5</b>	<b>Alarm and Alert Synchronization</b>	<b>7</b>
<b>6</b>	<b>Mapping of Ericsson Alarm MIB OIDs to Alarm Monitor and Alarm CLI attributes</b>	<b>8</b>
<b>7</b>	<b>Alarm List Rebuilt</b>	<b>18</b>
<b>8</b>	<b>Autosuspension</b>	<b>19</b>
<b>9</b>	<b>Target Additional Information Format</b>	<b>20</b>
<b>10</b>	<b>ENM CLI Commands for SNMP Subscriptions Configuration and Management</b>	<b>28</b>
10.1	ENM CLI Syntax for SNMPv2c	30
10.2	ENM CLI Syntax for SNMPv3	31
<b>11</b>	<b>ENM CLI Commands for Basic FM SNMP NBI Configuration and Management</b>	<b>34</b>
<b>12</b>	<b>ENM FM SNMP NBI and OSS-RC FM SNMP NBI Differences</b>	<b>36</b>
	<b>Reference List</b>	<b>39</b>





# 1 ENM FM SNMP Northbound Interface Integration Programmers Guide

Description of how to interact with the ENM FM SNMP NBI from external systems (referred as NMS) and the features that are supported.

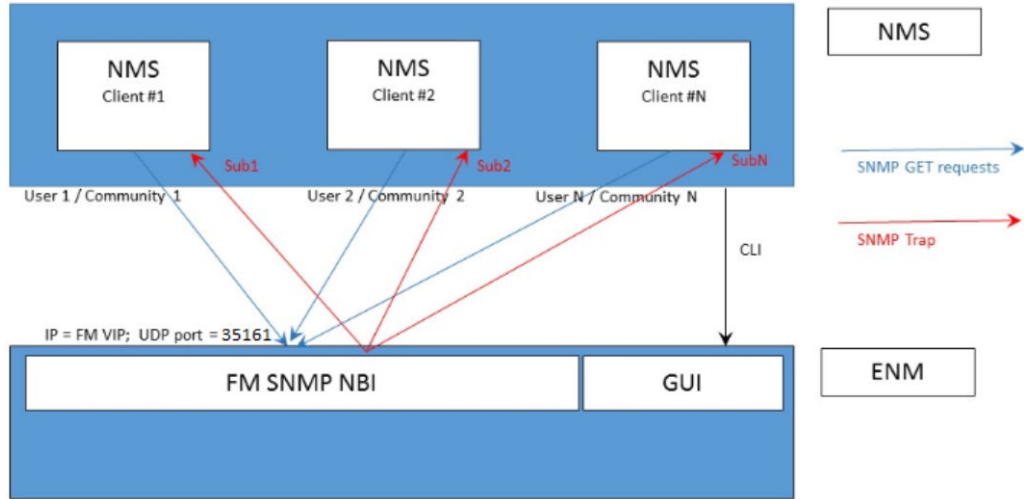
The main purpose of NBI is to provide an interface to the external system to subscribe to alarms and alerts from ENM. There are operations supported by the SNMP NBI in ENM that can be initiated from an external system and performed on alarms and alerts in ENM.

To interact with the ENM FM SNMP NBI from an external system, you need the following:

- A fully deployed ENM system.
- FM VIP IPv4/IPv6 external address to connect to the ENM FM SNMP NBI through SNMP protocol. The needed open ports must be opened. For more details, see [\[1\]](#).
- A valid user ID and password to access the ENM System:
  - Security System Administrator: to create users of NbiFmSnmConfig\_Administrator, NbiFmSnmConfig\_Operator, and NbiFmSnmManager roles.
  - NbiFmSnmConfig\_Administrator: to create, modify, delete, and monitor SNMP FM subscriptions and general configuration parameters through ENM CLI.
  - NbiFmSnmConfig\_Operator: to monitor SNMP FM subscriptions and general configuration parameters through ENM CLI.



### FM SNMP NBI



- Notes:
- 1) User1-N meaningful only in case of SNMPv3
  - 2) Community1-N meaningful only in case of SNMPv2c
  - 3) Sub 1-N are the trap destination addresses declared in FM SNMP subscriptions

## 1.1 SNMP Versions Supported

Table 1 SNMP Version Supported and Description

SNMP Version	Description
SNMPv2c	Community-based Simple Network Management Protocol Version 2
SNMPv3	Simple Network Management Protocol version 3

## 1.2 MIB Supported

Table 2 MIB Supported

MIB Version	Description	Version
ERICSSON-ALARM-MIB	iso.org.dod.internet.private.enterprises.ericsson.ericssonModules.ericssonAlarmMIB	2.1



## 2 Secure FM SNMP NBI User Roles

### Prerequisites

The following steps must be performed by a user with security system administrator rights.

### Steps

1. Open User Management application.
2. Create a user with NbiFmSnmConfig\_Administrator role and users with NbiFmSnmConfig\_Operator role (if needed).
3. Create as many users with NbiFmSnmManager role as the number of expected NMS clients operating with SNMPv3.

**Note:** The flow to remove users of NbiFmSnmManager role (associated to subscriptions) is to remove the subscription and then to remove the user. The subscription remains active even if the associated user is removed.



## 3 Initialize FM SNMP NBI

### Prerequisites

In SNMPv3, the FM subscription must be uniquely associated to a user of NbiFmSnmPManager role.

The following steps must be performed by a user with NbiFmSnmPConfig\_Administrator role.

### Steps

1. By ENM CLI application, create as many SNMP FM subscriptions as the number of expected NMS clients.
2. By ENM CLI application, verify that the administrative state of the SNMPNBI agent is unlocked to be able to get MIB objects and to receive traps.
3. By ENM CLI application, verify that the automatic suspension administrative state of the FM SNMP NBI is enabled to activate suspension or disabled to deactivate suspension.
4. (Optional) Perform an SNMP get operation to check the connectivity and the subscription on the AlarmActiveLastSequenceNo  
OID: .1.3.6.1.4.1.193.183.4.1.3.1.0

**Note:** The SNMP request must contain a community (in SNMPv2c) or a user (in SNMPv3) in line with the SNMP FM subscriptions created at [Step 1](#)

### Example 1 snmpget Using Public Community

```
[root@svc-4-nbfmsnmp test_client]# snmpget -v2c -c public
127.0.0.1:35161 .1.3.6.1.6.3.10.2.1.3.0.1.3.6.1.6.3.10.2.1.3.0 =
Gauge32: 3
```



# 4 Spontaneous Alarms and Alerts Communication

NMS clients receive asynchronously alarms and alerts through SNMP at the IP address specified in the related subscription.

## Alarms Notifications

Alarms notifications are, as follows:

- `.iso.org.dod.internet.private.enterprises.ericsson.ericssonModules.ericssonAlarmMIB.eriAlarmNotifications.eriAlarmNotifsPrefix.eriAlarmIndeterminate`
- `.iso.org.dod.internet.private.enterprises.ericsson.ericssonModules.ericssonAlarmMIB.eriAlarmNotifications.eriAlarmNotifsPrefix.eriAlarmCritical`
- `.iso.org.dod.internet.private.enterprises.ericsson.ericssonModules.ericssonAlarmMIB.eriAlarmNotifications.eriAlarmNotifsPrefix.eriAlarmMajor`
- `.iso.org.dod.internet.private.enterprises.ericsson.ericssonModules.ericssonAlarmMIB.eriAlarmNotifications.eriAlarmNotifsPrefix.eriAlarmMinor`
- `.iso.org.dod.internet.private.enterprises.ericsson.ericssonModules.ericssonAlarmMIB.eriAlarmNotifications.eriAlarmNotifsPrefix.eriAlarmWarning`
- `.iso.org.dod.internet.private.enterprises.ericsson.ericssonModules.ericssonAlarmMIB.eriAlarmNotifications.eriAlarmNotifsPrefix.eriAlarmCleared`

If the size of the `eriAlarmNObjAdditionalText` or the `eriAlarmNObjAdditionalInfo` attribute of the traps is longer than 256 bytes, a second trap of the following type is sent:

- `.iso.org.dod.internet.private.enterprises.ericsson.ericssonModules.ericssonAlarmMIB.eriAlarmNotifications.eriAlarmNotifsPrefix.eriAlarmAppendInfo`

## Alerts Notifications

Alerts notifications are, as follows:



- `.iso.org.dod.internet.private.enterprises.ericsson.ericssonModules.ericssonAlarmMIB.eriAlarmNotifications.eriAlarmNotifsPrefix.eriAlarmIndAlert`
- `.iso.org.dod.internet.private.enterprises.ericsson.ericssonModules.ericssonAlarmMIB.eriAlarmNotifications.eriAlarmNotifsPrefix.eriAlarmCriticalAlert`
- `.iso.org.dod.internet.private.enterprises.ericsson.ericssonModules.ericssonAlarmMIB.eriAlarmNotifications.eriAlarmNotifsPrefix.eriAlarmMajorAlert`
- `.iso.org.dod.internet.private.enterprises.ericsson.ericssonModules.ericssonAlarmMIB.eriAlarmNotifications.eriAlarmNotifsPrefix.eriAlarmMinorAlert`
- `.iso.org.dod.internet.private.enterprises.ericsson.ericssonModules.ericssonAlarmMIB.eriAlarmNotifications.eriAlarmNotifsPrefix.eriAlarmWarnAlert`

If the size of the `eriAlarmNObjAdditionalText` or the `eriAlarmNObjAdditionalInfo` attribute of the trap is longer than 256 bytes, a second trap of the following type is sent:

- `.iso.org.dod.internet.private.enterprises.ericsson.ericssonModules.ericssonAlarmMIB.eriAlarmNotifications.eriAlarmNotifsPrefix.eriAlarmAppendAlertInfo`



## 5 Alarm and Alert Synchronization

NMS clients synchronize open alarms and alerts performing SNMP Get or GetNext commands.

### Alarms Synchronization

NMS clients synchronize open alarms performing SNMP Get/GetNext commands on the following table:

- `.iso.org.dod.internet.private.enterprises.ericsson.ericssonModules.ericssonAlarmMIB.eriAlarmObjects.eriAlarmActiveAlarms.eriAlarmActiveAlarmTable`

The size of the `eriAlarmActiveAdditionalText` and the `eriAlarmActiveAdditionalInfo` attribute of the alarm instance in the table are cut to 512 bytes.

### Alerts Synchronization

NMS clients synchronize alerts performing SNMP Get/GetNext commands on the following table:

- `iso.org.dod.internet.private.enterprises.ericsson.ericssonModules.ericssonAlarmMIB.eriAlarmObjects.eriAlarmAlerts.eriAlarmAlertTable`

The size of the `eriAlarmAlertAdditionalText` and the `eriAlarmAlertAdditionalInfo` attribute of the alert instance in the table are cut to 512 bytes.



## 6 Mapping of Ericsson Alarm MIB OIDs to Alarm Monitor and Alarm CLI attributes

The ERICSSON-ALARM-MIB is identified as .iso.org.dod.internet.private.enterprises.ericsson.ericssonModules.ericssonAlarmMIB (.1.3.6.1.4.1.193.183.4).

The ERICSSON-ALARM-MIB trap attributes and ERICSSON-ALARM-MIB table columns (either eriAlarmActiveAlarmTable and eriAlarmAlertTable) store the OID used in mapping respectively for trap and SNMP GET response. The Value field stores a description of the conversion applied to map value of the attribute.

Table 3 General Mapping Table

Alarm Monitor / Alarm CLI attributes	Alarm Notifications	eriAlarmActiveAlarmTable (.1.3.6.1.4.1.193.183.4.1.3.5)	Alert Notifications	eriAlarmAlertTable (.1.3.6.1.4.1.193.183.4.1.4.5)	Value
Object of Reference	eriAlarmActiveManagedObject .1.3.6.1.4.1.193.183.4.1.3.5.1.5	eriAlarmActiveManagedObject .1.3.6.1.4.1.193.183.4.1.3.5.1.5	eriAlarmAlertManagedObject .1.3.6.1.4.1.193.183.4.1.4.5.1.5	eriAlarmAlertManagedObject .1.3.6.1.4.1.193.183.4.1.4.5.1.5	The "3GPP TS 32.106-8 V3.2, Name convention for Managed Objects" is used. The size is in the range (4..150)
Event Time	see <a href="#">Mapping Table for Time Attributes</a>	see <a href="#">Mapping Table for Time Attributes</a>	see <a href="#">Mapping Table for Time Attributes</a>	see <a href="#">Mapping Table for Time Attributes</a>	Date conversion according to RFC 2579 standard <a href="https://tools.ietf.org/html/rfc2579">https://tools.ietf.org/html/rfc2579</a>
Cease Time	see <a href="#">Mapping Table for Time Attributes</a>	see <a href="#">Mapping Table for Time Attributes</a>	see <a href="#">Mapping Table for Time Attributes</a>	see <a href="#">Mapping Table for Time Attributes</a>	Date conversion according to RFC 2579 standard <a href="https://tools.ietf.org/html/rfc2579">https://tools.ietf.org/html/rfc2579</a>
Ack Time (available only in alarm CLI reports)	see <a href="#">Mapping Table for Time Attributes</a>	see <a href="#">Mapping Table for Time Attributes</a>	see <a href="#">Mapping Table for Time Attributes</a>	see <a href="#">Mapping Table for Time Attributes</a>	Date conversion according to RFC 2579 standard <a href="https://tools.ietf.org/html/rfc2579">https://tools.ietf.org/html/rfc2579</a>
commentTime (available only in alarm CLI reports)	see <a href="#">Mapping Table for Time Attributes</a>	see <a href="#">Mapping Table for Time Attributes</a>	see <a href="#">Mapping Table for Time Attributes</a>	see <a href="#">Mapping Table for Time Attributes</a>	Date conversion according to RFC 2579 standard <a href="https://tools.ietf.org/html/rfc2579">https://</a>



Alarm Monitor / Alarm CLI attributes	Alarm Notifications	eriAlarmActiveAlarmTable (.1.3.6.1.4.1.193.183.4.1.3.5)	Alert Notifications	eriAlarmAlertTable (.1.3.6.1.4.1.193.183.4.1.4.5)	Value
					<a href="https://tools.ietf.org/html/rfc2579">tools.ietf.org/html/rfc2579</a>
lastUpdated	see <a href="#">Mapping Table for Time Attributes</a>	see <a href="#">Mapping Table for Time Attributes</a>	see <a href="#">Mapping Table for Time Attributes</a>	see <a href="#">Mapping Table for Time Attributes</a>	Date conversion according to RFC 2579 standard <a href="https://tools.ietf.org/html/rfc2579">https://tools.ietf.org/html/rfc2579</a>
Severity	see <a href="#">Mapping Table for Trap Type OIDs</a>	eriAlarmActiveSeverity .1.3.6.1.4.1.193.183.4.1.3.5.1.1 0	see <a href="#">Mapping Table for Trap Type OIDs</a>	eriAlarmAlertSeverity .1.3.6.1.4.1.193.183.4.1.4.5.1.9	For eriAlarmActiveSeverity and eriAlarmAlertSeverity the mapping from String to integer is made as per ItuPerceivedSeverity Integer values defined into RFC 3877 <a href="https://tools.ietf.org/html/rfc3877">https://tools.ietf.org/html/rfc3877</a> : cleared (1) is not applicable for eriAlarmAlertSeverity
Previous Severity	Not applicable	eriAlarmActiveOriginalSeverity .1.3.6.1.4.1.193.183.4.1.3.5.1.1 1 See also <a href="#">Mapping Table for Severity Attributes</a>	Not applicable	Not applicable	The original severity as reported by the alarm raise notification mapping from String to integer is made as per ItuPerceivedSeverity Integer values defined into RFC 3877 <a href="https://tools.ietf.org/html/rfc3877">https://tools.ietf.org/html/rfc3877</a> : cleared (1) is not applicable for eriAlarmActiveOriginalSeverity
Probable Cause	eriAlarmActiveProbableCause .1.3.6.1.4.1.193.183.4.1.3.5.1.9	eriAlarmActiveProbableCause .1.3.6.1.4.1.193.183.4.1.3.5.1.9	eriAlarmAlertProbableCause .1.3.6.1.4.1.193.183.4.1.4.5.1.8	eriAlarmAlertProbableCause .1.3.6.1.4.1.193.183.4.1.4.5.1.8	Mapping from String to integer as per EriProbableCause as defined in ERICSSON-ALARM-PC-MIB. See <a href="#">Probable Cause Mapping</a>
Specific Problem	eriAlarmActiveSpecificProblem .1.3.6.1.4.1.193.183.4.1.3.5.1.4	eriAlarmActiveSpecificProblem .1.3.6.1.4.1.193.183.4.1.3.5.1.4	eriAlarmActiveSpecificProblem .1.3.6.1.4.1.193.183.4.1.3.5.1.4	eriAlarmActiveSpecificProblem .1.3.6.1.4.1.193.183.4.1.3.5.1.4	One-to-one mapping. The size is in the range (4..64)
Event Type	eriAlarmActiveEventType	eriAlarmActiveEventType	eriAlarmActiveEventType	eriAlarmActiveEventType	Mapping from String to



Alarm Monitor / Alarm CLI attributes	Alarm Notifications	eriAlarmActiveAlarmTable (.1.3.6.1.4.1.193.183.4.1.3.5)	Alert Notifications	eriAlarmAlertTable (.1.3.6.1.4.1.193.183.4.1.4.5)	Value
	.1.3.6.1.4.1.193.183.4.1.3.5.1.6	.1.3.6.1.4.1.193.183.4.1.3.5.1.6	.1.3.6.1.4.1.193.183.4.1.3.5.1.6	.1.3.6.1.4.1.193.183.4.1.3.5.1.6	integer as per IANAItuEventT type map defined into RFC 3877 <a href="https://tools.ietf.org/html/rfc3877">https://tools.ietf.org/html/rfc3877</a> See Event Type Mapping.
EventPoId (available only in alarm CLI reports)	see eventPoId Mapping	see eventPoId Mapping	see eventPoId Mapping	see eventPoId Mapping	
Problem Text	eriAlarmNObjAdditionalText .1.3.6.1.4.1.193.183.4.1.2.1	eriAlarmActiveAdditionalText .1.3.6.1.4.1.193.183.4.1.3.5.1.1.2	eriAlarmNObjAdditionalText .1.3.6.1.4.1.193.183.4.1.2.1	eriAlarmAlertAdditionalText .1.3.6.1.4.1.193.183.4.1.4.5.1.1.0	For eriAlarmActiveTable and eriAlarmAlertTable it is one-to-one mapping if ("Problem Text" size < 512 bytes) otherwise only the first 512 bytes are stored. For alarm and alert traps, if ("Problem Text" > 256 bytes) the first 256 bytes are sent with the original trap, other 256 bytes are sent with the eriAlarmAppendInfo or eriAlarmAppendAlertInfo respectively. If ("Problem Text" size > 512 bytes) the first 512 bytes are stored and sent the remaining ones are discarded.
	eriAlarmNObjMoreAdditionalText .1.3.6.1.4.1.193.183.4.1.2.2	Not Applicable	eriAlarmNObjMoreAdditionalText .1.3.6.1.4.1.193.183.4.1.2.2	Not Applicable	if ("Problem Text" > 256 bytes) eriAlarmNObjMoreAdditionalText is set to True (1) and eriAlarmAppendInfo or eriAlarmAppendAlertInfo is sent, otherwise it is set to False (2)
Not applicable	eriAlarmNObjResourceId .1.3.6.1.4.1.193.183.4.1.2.3	eriAlarmActiveResourceId .1.3.6.1.4.1.193.183.4.1.3.5.1.1.4	eriAlarmNObjResourceId .1.3.6.1.4.1.193.183.4.1.2.3	eriAlarmAlertResourceId .1.3.6.1.4.1.193.183.4.1.4.5.1.1.1	Always set to False (2)



Alarm Monitor / Alarm CLI attributes	Alarm Notifications	eriAlarmActiveAlarmTable (.1.3.6.1.4.1.193.183.4.1.3.5)	Alert Notifications	eriAlarmAlertTable (.1.3.6.1.4.1.193.183.4.1.4.5)	Value
Not applicable	eriAlarmActiveLastSequenceNo .1.3.6.1.4.1.193.183.4.1.3.3	Not Applicable	eriAlarmAlertLastSequenceNo .1.3.6.1.4.1.193.183.4.1.4.3	Not Applicable	The last used sequence number for an alarm state notification or an alert. This can be used to detect a lost notification.
Additional Information (available in Alarm Details page and in the additionalInformation CLI attribute)	eriAlarmNObjAdditionalInfo .1.3.6.1.4.1.193.183.4.1.2.4	eriAlarmActiveAdditionalInfo .1.3.6.1.4.1.193.183.4.1.3.5.1.15	eriAlarmNObjAdditionalInfo .1.3.6.1.4.1.193.183.4.1.2.4	eriAlarmAlertAdditionalInfo .1.3.6.1.4.1.193.183.4.1.4.5.1.12	targetAdditionalInformation attribute and, optionally, user defined additional information attributes are forwarded as trap and made available in either eriAlarmActiveAdditionalInfo (eriAlarmActiveTable) and eriAlarmAlertAdditionalInfo (eriAlarmAlertTable). For alarm and alert traps, if ("eriAlarmActiveAdditionalInfo" size > 256 bytes) the first 256 bytes are sent with the original trap, the remaining 256 bytes are sent with the eriAlarmAppendInfo eriAlarmAppendAlertInfo respectively.
	eriAlarmNObjMoreAdditionalInfo .1.3.6.1.4.1.193.183.4.1.2.5	Not Applicable	eriAlarmNObjMoreAdditionalInfo .1.3.6.1.4.1.193.183.4.1.2.5	Not Applicable	If the targetAdditionalInformation exceeds 256 bytes it is set to True (1) and an AeriAlarmAppendInfo or eriAlarmAppendAlertInfo is sent, otherwise it is set to False (2)

**Note:** All other attributes available in the Alarm Monitor and in the Alarm CLI, which are not reported in the previous table, are not available over the FM SNMP NBI.



Table 4 Mapping Table for Trap Type OIDs

Present Severity	Record type	Notification type OID
WARNING	ALARM   ALARM_SUPPRESSED_ALARM   CLEARALL   HB_FAILURE_NO_SYNCH   HEARTBEAT_ALARM   NODE_SUSPENDED   NON_SYNCHABLE_ALARM   OSCILLATORY_HB_ALARM   OUT_OF_SYNC   REPEATED_ALARM   REPEATED_NON_SYNCHABLE   SYNCHRONIZATION_ALARM   TECHNICIAN_PRESENT   UPDATE	eriAlarmWarning .1.3.6.1.4.1.193.183.4.2.0.2
WARNING	ERROR_MESSAGE   REPEATED_ERROR_MESSAGE   CLEAR_LIST	eriAlarmWarnAlert .1.3.6.1.4.1.193.183.4.2.0.11
MINOR	ALARM   ALARM_SUPPRESSED_ALARM   CLEARALL   HB_FAILURE_NO_SYNCH   HEARTBEAT_ALARM   NODE_SUSPENDED   NON_SYNCHABLE_ALARM   OSCILLATORY_HB_ALARM   OUT_OF_SYNC   REPEATED_ALARM   REPEATED_NON_SYNCHABLE   SYNCHRONIZATION_ALARM   TECHNICIAN_PRESENT   UPDATE	eriAlarmMinor .1.3.6.1.4.1.193.183.4.2.0.3
MINOR	ERROR_MESSAGE   REPEATED_ERROR_MESSAGE   CLEAR_LIST	eriAlarmMinorAlert .1.3.6.1.4.1.193.183.4.2.0.12
MAJOR	ALARM   ALARM_SUPPRESSED_ALARM   CLEARALL   HB_FAILURE_NO_SYNCH   HEARTBEAT_ALARM   NODE_SUSPENDED   NON_SYNCHABLE_ALARM   OSCILLATORY_HB_ALARM   OUT_OF_SYNC   REPEATED_ALARM   REPEATED_NON_SYNCHABLE   SYNCHRONIZATION_ALARM   TECHNICIAN_PRESENT   UPDATE	eriAlarmMajor .1.3.6.1.4.1.193.183.4.2.0.4
MAJOR	ERROR_MESSAGE   REPEATED_ERROR_MESSAGE   CLEAR_LIST	eriAlarmMajorAlert .1.3.6.1.4.1.193.183.4.2.0.13
INDETERMINATE	ALARM   ALARM_SUPPRESSED_ALARM   CLEARALL   HB_FAILURE_NO_SYNCH   HEARTBEAT_ALARM   NODE_SUSPENDED   NON_SYNCHABLE_ALARM   OSCILLATORY_HB_ALARM   OUT_OF_SYNC   REPEATED_ALARM   REPEATED_NON_SYNCHABLE   SYNCHRONIZATION_ALARM   TECHNICIAN_PRESENT   UPDATE	eriAlarmIndeterminate .1.3.6.1.4.1.193.183.4.2.0.1
INDETERMINATE	ERROR_MESSAGE   REPEATED_ERROR_MESSAGE	eriAlarmIndAlert .1.3.6.1.4.1.193.183.4.2.0.10



Present Severity	Record type	Notification type OID
	SYNCHRONIZATION_ABORTED   SYNCHRONIZATION_IGNORED	
CRITICAL	ALARM   ALARM_SUPPRESSED_ALARM   CLEARALL   HB_FAILURE_NO_SYNCH   HEARTBEAT_ALARM   NODE_SUSPENDED   NON_SYNCHABLE_ALARM   OSCILLATORY_HB_ALARM   OUT_OF_SYNC   REPEATED_ALARM   REPEATED_NON_SYNCHABLE   SYNCHRONIZATION_ALARM   TECHNICIAN_PRESENT   UPDATE	eriAlarmCritical .1.3.6.1.4.1.193.183.4.2.0.5
CRITICAL	ERROR_MESSAGE   REPEATED_ERROR_MESSAGE   CLEAR_LIST	eriAlarmCriticalAlert .1.3.6.1.4.1.193.183.4.2.0.14
CLEARED	ALARM   ALARM_SUPPRESSED_ALARM   CLEARALL   HB_FAILURE_NO_SYNCH   HEARTBEAT_ALARM   NODE_SUSPENDED   NON_SYNCHABLE_ALARM   OSCILLATORY_HB_ALARM   OUT_OF_SYNC   REPEATED_ALARM   REPEATED_NON_SYNCHABLE   SYNCHRONIZATION_ALARM   TECHNICIAN_PRESENT   UPDATE	eriAlarmCleared .1.3.6.1.4.1.193.183.4.2.0.7

The Severity and the Record Type attributes contribute to the definition of the Notification type OID to be forwarded.

Table 5 Mapping Table for Time Attributes

lastAlarmOperation	eriAlarmActiveEventTime (traps) .1.3.6.1.4.1.193.183.4.1.3.5.1.7	eriAlarmActiveEventTime (eriAlarmActiveAlarmTable entry) .1.3.6.1.4.1.193.183.4.1.3.5.1.7
CLEAR	Cease Time	Cease Time
NEW	Event Time	Event Time
CHANGE	lastUpdated	lastUpdated
ACKSTATE_CHANGE	lastUpdated that matches with Ack Time	lastUpdated that matches with Ack Time
COMMENT	lastUpdated that matches with commentTime	lastUpdated that matches with commentTime

The Alarm Operation determines which time attribute is to be used for mapping.

Table 6 Mapping Table for Severity Attributes

Alarm Operation	eriAlarmActiveSeverity (entry)	eriAlarmActiveOriginalSeverity (entry)
NEW	Severity	Severity
CLEAR	Severity	unmodified
CHANGE	Severity	unmodified



The Alarm Operation determines which severity has to be used for mapping. In case of change of severity, the trap OID is modified in accordance with the mapping table for the notification type OIDs (see [Mapping Table for Trap Type OIDs](#)).

Table 7 Alarm Operations

Operation	Alarm Monitor	Ericsson Alarm MIB
Acknowledge	Ack Time not empty AND Ack not empty are used for the text	An eriAlarmAppendAlertInfo trap is sent. The attribute Acknowledge information is included in eriAlarmNObjAdditionalText (.1.3.6.1.4.1.193.183.4.1.2.1). The OSS-RC format is used: <Action performed by>;<Time Of Action>;<Action> . Example: (eriAlarmNObjAdditionalText) VALUE : suser1;2016-11-17 08:16:35;Acknowledge.
Unacknowledge	Ack Time not empty AND Ack not empty are used for the text	An eriAlarmAppendAlertInfo trap is sent. The attribute Unacknowledge information is included in eriAlarmNObjAdditionalText (.1.3.6.1.4.1.193.183.4.1.2.1). The OSS-RC format is used: <Action performed by>; <Time Of Action>; <Action> . Example: (eriAlarmNObjAdditionalText) VALUE : suser1;2016-11-17 08:16:35;Unacknowledge.
Comment	commentText not empty used for the text	An eriAlarmAppendAlertInfo trap is sent. The comment is included in eriAlarmNObjAdditionalText (.1.3.6.1.4.1.193.183.4.1.2.1). In OSS-RC, the trap is forwarded without any reference that the comment field has been updated

Alarm operations are forwarded to the SNMP Manager using the OSS-RC approach with the purpose to minimize impacts at the least possible.

### eventPoId Mapping

The eventPoId is mapped to eriAlarmActiveMajorType and eriAlarmActiveMinorType for alarm notifications and to eriAlarmAlertMajorType and eriAlarmAlertMinorType for alert notifications and in either the EriAlarmActiveAlarmEntry and the EriAlarmAlertEntry.

eventPoId is a 64 bit long value and the mapping is as follows:

- The attribute eriAlarmActiveMajorType is set to upper 32 bits of eventPoId
- The attribute eriAlarmActiveMinorType is set to lower 32 bits of eventPoId.



Example 2 Alarm eventPoId in ENM CLI: 281605391015068 which is 0x1001E5D4C5C9C in Hex  
Lower 32 bits are: 0x5D4C5C9C which is 1565285532

Upper 32 bits are: 0x0001001E which is 65566

Alarm eventPoId mapped in SNMP NBI:

- ERICSSON-ALARM-MIB::eriAlarmActiveMajorType = 65566
- ERICSSON-ALARM-MIB::eriAlarmActiveMinorType = 1565285532

Steps to get eventPoId from fields eriAlarmActiveMajorType and eriAlarmActiveMinorType:

1. Convert eriAlarmActiveMajorType value to Hexadecimal:  
eriAlarmActiveMajorType = 65566 which is 0x0001001E
2. Convert eriAlarmActiveMinorType value to Hexadecimal:  
eriAlarmActiveMinorType = 1565285532 which is 0x5D4C5C9C
3. Combine both eriAlarmActiveMajorType and eriAlarmActiveMinorType Hexadecimal values and convert into Decimal:  
eriAlarmActiveMajorType eriAlarmActiveMinorType = 0x0001001E5D4C5C9C which is 281605391015068 (same as eventPoId reported in ENM CLI)

Example 3 Alarm eventPoId in ENM CLI: 1500148 which is 0x0016E3F4 in Hex

Lower 32 bits are: 0x0016E3F4 which is 1500148

Upper 32 bits are: 0x00000000 which is 0

Alarm eventPoId mapped in SNMP NBI:

- ERICSSON-ALARM-MIB::eriAlarmActiveMajorType = 0
- ERICSSON-ALARM-MIB::eriAlarmActiveMinorType = 1500148

Steps to get eventPoId from fields eriAlarmActiveMajorType and eriAlarmActiveMinorType:

1. Convert eriAlarmActiveMajorType value to Hexadecimal:  
eriAlarmActiveMajorType = 0 which is 0x00000000
2. Convert eriAlarmActiveMinorType value to Hexadecimal:  
eriAlarmActiveMinorType = 1500148 which is 0x0016E3F4
3. Combine both eriAlarmActiveMajorType and eriAlarmActiveMinorType Hexadecimal values and convert into Decimal:



`eriAlarmActiveMajorType eriAlarmActiveMinorType =0x00000000016E3F4` which is 1500148 (same as `eventPoId` reported in ENM CLI)

Example 4 Alarm `eventPoId` in ENM CLI: -2 which is `0xFFFFFFFFFFFFFFFE` in Hex

Lower 32 bits are: `0xFFFFFFFF` which is 4294967294

Upper 32 bits are: `0xFFFFFFFF` which is 4294967295

Alarm `eventPoId` mapped in SNMP NBI:

— `ERICSSON-ALARM-MIB::eriAlarmActiveMajorType = 4294967295`

— `ERICSSON-ALARM-MIB::eriAlarmActiveMinorType = 4294967294`

Steps to get `eventPoId` from fields `eriAlarmActiveMajorType` and `eriAlarmActiveMinorType`:

1. Convert `eriAlarmActiveMajorType` value to Hexadecimal:

`eriAlarmActiveMajorType = 4294967295` which is `0xFFFFFFFF`

2. Convert `eriAlarmActiveMinorType` value to Hexadecimal:

`eriAlarmActiveMinorType = 4294967294` which is `0xFFFFFFFFE`

3. Combine both `eriAlarmActiveMajorType` and `eriAlarmActiveMinorType` Hexadecimal values and convert into Decimal in signed 2's complement:

`eriAlarmActiveMajorType eriAlarmActiveMinorType = 0xFFFFFFFFFFFFFFFE` which is -2 (same as `eventPoId` presented in ENM CLI)

**Note:** The same examples can be applied to alert notifications for `eriAlarmAlertMajorType` and `eriAlarmAlertMinorType` attributes.

### Probable Cause Mapping

The NBI tentatively maps every possible probable cause string coming from any node to the field `eriAlarmActiveProbableCause` for alarms or to the field `eriAlarmAlertProbableCause` for alerts through an Ericsson Alarm MIB enum (`EriProbableCause`).

In case the mapping is not possible the value `m3100Indeterminate(0)` is reported; the value of the mismatched probable cause is anyway reported in the `eriAlarmNObjAdditionalText` field of the trap

The format is: `ProbableCause=<probable cause>`



## Event Type Mapping

The NBI tentatively maps every possible event type string coming from any node to the field `eriAlarmActiveEventType` for alarms or to the field `eriAlarmAlertEventType` for alerts through an Ericsson Alarm MIB enum (IANAItuEventType).

In case this mapping is not possible the value `other(1)` is reported; the value of the mismatched event type is anyway reported in the `eriAlarmNObjAdditionalText` field of the trap.

The format is: `EventType=<event type>`

**Note:** The variable bindings order in the traps cannot be used to identify the attributes.

## eriAlarmNObjAdditionalInfo Configuration

The content of the `eriAlarmNObjAdditionalInfo` field can be manipulated using two configuration parameters:

- `additionalInformationBlocker` to add or remove the `targetAdditionalInformation` from the `eriAlarmNObjAdditionalInfo` field.
- `nbiSnmplibUserAdditionalInfo` to add `additionalInformation` attributes visible in alarm details or in CLI to the `eriAlarmNObjAdditionalInfo` field.

For example the `additionalInformation` attribute `unreachableIpAddress` (received from a Network Element) can be mapped to `UNREACHABLEIP` and the value can be sent through the SNMP NBI.

The registered names are shown in the table of [Target Additional Information Format](#) on page 20.

See ENM Monitoring System Administrator Guide, Reference [3] for more details to set these configuration parameters.



## 7 Alarm List Rebuilt

NMS clients can become aware that the active alarm list has reached a stable situation after a restart or after a system internal audit process, receiving a notification, as follows:

- `.iso.org.dod.internet.private.enterprises.ericsson.ericssonModules.ericssonAlarmMIB.eriAlarmNotifications.eriAlarmNotifsPrefix.eriAlarmAlarmListRebuilt`

The notification is an indication for NMS to start an alarm resynchronization procedure.



## 8 Autosuspension

When creating an NMS subscription on ENM FM SNMP NBI, it is possible to specify if the automatic suspension is enabled or disabled. The "suspension admin state" attribute can be set, as follows:

- enabled (autosuspension is active).
- disabled (autosuspension is not active).

When the "suspension admin state" is set to "disabled", there is no automatic suspension of NMS user subscription. NMS continues to receive traps from the SNMP NBI agent.

When the "suspension admin state" is set to "enabled", the time limit (not configurable) for the automatic suspension of NMS user subscription is 15 minutes.

If heartbeat set to "PULL mode" and ENM FM SNMP NBI does not receive any request (for example SNMP GET on scalars or on tables) from the subscribed NMS within a specified time limit, the subscription is automatically suspended. The "suspension operational state" is changed from "not suspended" to "suspended". No alarm notifications (traps) are broadcasted to the suspended NMS subscription.

SNMP Agent replies to SNMP GET operations requested by the NMS subscription (on both scalars variables and tables). So if the SNMP Agent receives any request from a suspended NMS subscription, the NMS subscription is automatically resumed by ENM FM SNMP NBI. The "suspension operational state" is changed from "suspended" to "not suspended". If the GET operation request is performed on a scalar variable, the `eriAlarmAlarmListRebuilt` is sent to NMS at first to ensure that NMS can realign alarms.

**Note:** The 15-minute time period is not a configurable parameter. The 15-minute time period is applied as persistence time interval to change the "autosuspension operational state" from "not suspended" to "suspended". The status change can occur within a minute after the 15-minute interval.



## 9 Target Additional Information Format

The `targetAdditionalInformation` is a string composed by one or more name and value pairs separated by a semicolon  
`name=value;name=value;name=value....`

The following table describes the name of the string, the meaning, and the type of the value.

Name	Meaning	Value Type
CI	Correlation Information	JSON object
Direction	RF branch in RadioNode and its direction	String
DN<n>	Distinguished Name	String
LK	License Key	String
Logical Name<n>	Logical Name	String
PLMN	Public Land Mobile Network (PLMN) identifier	JSON object
PLMN ID-eNB ID <n>	Global eNodeB Identifier	String
PLMN ID-gNB ID <n>	Global gNodeB Identifier	String
ST	Source Type of the Node that reported the alarm (according to ENM).	String
UUID	Universally Unique Identifier	String
UNREACHABLEIP	IP address of an unreachable Network Element	String

### Direction

Direction is UL, DL, or both UL and DL.

The RF branch is identified by the RDN (defined in 3GPP TS 32.106-8 V3.2), consisting of instances of MO `AntennaUnitGroup` and MO `RfBranch`.

Value string format = the value begins with `<UL_DL_dir>` to indicate if the error occurs in Uplink (UL), Downlink (DL), or both (UL and DL). The value is followed by instances for the associated `AntennaUnitGroup` MO and `RfBranch` MO.

Examples:

```
Direction = UL and DL AntennaUnitGroup=2,RfBranch=1;
Direction = DL AntennaUnitGroup=2,RfBranch=2;
```

### Distinguished Name

Distinguished Name DN<n> is an integer between two and five.



Distinguished Name is limited by the space available in the SNMP trap and affected by the average size of the values of the DN. Distinguished Name is not application controlled. If necessary, truncation is performed by platform.

Value string format = Distinguished Name, as defined in 3GPP TS 32.106-8 V3.2, *Name convention for Managed Objects*.

Examples:

```
DN2 = ManagedElement=1,Equipment=1,RbsSubrack=R UW1,RbsSlot=5,AuxPlugInUnit=R UW-2 →
,DeviceGroup=R UW,AiDeviceSet=1,AiDevice=1;
DN3 = ManagedElement=1,Equipment=1,RbsSubrack=R UW1,RbsSlot=5,AuxPlugInUnit=R UW-2 →
,DeviceGroup=R UW,TrDeviceSet=1,TrDevice=1;
```

### License Key

Value string format = IDs (CXC numbers) associated with the license keys about to expire.

Examples:

```
LK = CXC4011075,CXC9879876;
```

### Logical Name

Logical Name <n> is an integer (1 to 70).

Value string format = Logical name on MO ManagedElement in the external node.

Undefined value, an empty string, is used for a node that has no configured logicalName.

NOT\_AVAILABLE is used when no information is available from the external node.

NOT\_AVAILABLE is also used for MULTI\_STANDARD node that does not have a logical name onMO ManagedElement.

Examples:

```
Logical Name 1 = NOT_AVAILABLE;
Logical Name 2 = enb108;
```

### Public Land Mobile Network (PLMN) Identifier

Maximum number of PLMNs (per 3GPP 36.300) is six, and one to six PLMNs are listed.

Value string format = Public Land Mobile Network (PLMN) identifier JSON object.



The PLMN JSON object is an array of strings in format of a MCC/MNC tuple. Where PLMN, Mobile Country Code (MCC), and Mobile Network Code (MNC) are defined in ITU-T Recommendation E.212.

MCC is a three-digit decimal number.

MNC is a two-digit or three-digit decimal number.

Since MCC can start with digit "0", the JSON object must be a string.

JSON PLMN object is a five characters to six characters string. The first three characters represent MCC, and the last two or three characters represent MNC.

For example in case PLMN object= 23415, then MCC=234 and MNC=15.

JSON structure schema for PLMN:

```
{
  "$schema": "http://json-schema.org/draft-06/schema#",
  "type": "array",
  "minItems": 1,
  "maxItems": 6,
  "items": {
    "type": "string",
    "minLength": 5,
    "maxLength": 6
  }
}
```

Examples:

```
PLMN = ["24085", "210501", "200512", "19052", "240481", "00101"];
```

### Global eNodeB Identifier

Global eNodeB Identifier PLMN ID-eNB ID <n> is an integer (1 to 9).

Value string format = Global eNodeB identifier as defined in 3GPP 36.300.

Example:

```
PLMN ID-eNB ID 1 = 26280-605103;
```

### Global gNodeB Identifier

Global gNodeB Identifier PLMN ID-gNB ID <n> is an integer (1 to 9).

Value string format = Global gNodeB identifier as defined in 3GPP 38.300.

Example:



```
PLMN ID-gNB ID 1 = 26280-605103;
```

## Source Type

Source Type ST of the Node that reported the alarm, according to ENM.

Source Type is added by ENM when making an alarm available northbound to NMS systems.

Value string format = SourceType string.

Examples:

```
ST = MGW;
```

## Universally Unique Identifier

Universally Unique Identifier UUID, as defined in ITU-T Rec. X.667 | ISO/IEC 9834-8, *Generation and registration of Universally Unique Identifiers (UUIDs)*.

Example:

```
UUID = 123e4567-e89b-12d3-a456-426655440000;
```

## Correlation Information (CI)

Correlation Information is a structured information built by the source of the alarm. Correlation information allows a management system to correlate the alarm with other alarms, including alarms from other sources.

The " character is allowed. The name and value pairs in the JSON object are allowed to contain any characters except for = and ;.

Within the JSON structure the following conventions are observed:

- Names that begin with upper case (or have a single character that is upper case) have correlation values that are interpreted as globally unique.
- Names that begin with lower case (or have a single character that is lower case) have correlation values that are interpreted as unique to the ME.

The naming convention does not apply to compound structures that can be made up of globally and locally unique correlation identifiers.

In the JSON structure, the specific letters are reserved to have specific meaning. The following letters are encoded by the fault notifier of the ME and interpreted by the EM or the NMS:

Reserved Letter	Mandatory or Not Mandatory
C is short for Compute resource.	Mandatory
I is short for Identifier.	Mandatory



Reserved Letter	Mandatory or Not Mandatory
n is short for a locally unique Name.	Mandatory
P is short for Primary and is used to indicate primary alarms.	Mandatory if it is known that there is or can be a primary or secondary alarm relationship. Not mandatory if there cannot be a secondary alarm (but can be present).
S is short for Secondary and is used to indicate secondary alarms or alarms that are not the primary alarm.	Mandatory for secondary alarms.

A "Compute resource" provides the functionality to run an instance of an operating system. The term is applicable to physical ("bare metal") and virtual deployments. For virtual deployments, a Compute resource corresponds to a virtual machine.

At least one Compute resource "C" must always be present. There are cases where the indication if the alarm is primary (P) or secondary (S) and the Alarm Correlation UUID are not present. There can be multiple instances of correlators for Compute resources and for Secondary alarms.

**Note:** If it is known that an alarm is not the cause of a secondary alarm, the "P" and Alarm Correlation UUID can be omitted.

For example:

```
CI = {"C": [{"I": "ea9a4000-531b-11e8-8e92-98a404439f11", "n": "RadioNode"}]}
```

Compute resource objects and Secondary alarm objects are stored in an array. This is illustrated in the schema and the examples as follows.

JSON Structure Schema for CI:

```
{
  "$schema": "http://json-schema.org/draft-06/schema#",
  "title": "Product set",
  "type": "object",
  "properties": {
    "C": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "I": {"type": "string"},
          "n": {"type": "string"}
        },
        "required": ["I", "n"]
      },
      "minItems": 1,
      "uniqueItems": true
    },
    "P": {
      "type": "string",
      "description": "correlation UUID"
    },
    "S": {
      "type": "array",
      "description": "correlation UUID(s)",
      "items": {
```



```

        "type" : "string"
      },
      "minItems": 1,
      "uniqueItems": true
    }
  },
  "required": ["C"]
}

```

### Example of Primary Alarm from Compute Resource Named RadioNode

Primary alarm from Compute Resource named RadioNode:

```

{
  "C": [{
    "I": "191f0558-94bd-49d1-9851-ac99ceac95c2",
    "n": "RadioNode"
  }],
  "P": "f81d4fae-7dec-11d0-a765-00a0c91e6bf6"
}

```

An example of the main alarm fields, including the targetAdditionalInformation name=value object and containing the JSON object represented in string format, is shown:

```

Specific Problem: "Resource Allocation Failure"
Severity: Minor
EventType: PROCESSING_ERROR_ALARM
ProbableCause: CONFIGURATION_OR_CUSTOMIZING_ERROR
AdditionalText: "No RF power available in radio equipment."
targetAdditionalInformation: CI = {"C": [{"I": "191f0558-94bd-49d1-9851-ac99ceac95c2", "n": "RadioNode"}], "P": "f81d4fae-7dec-11d0-a765-00a0c91e6bf6"};

```

The targetAdditionalInformation string represents Correlation Information for two sets of correlators:

- Compute resource with the global Identifier specified and a locally unique Name of RadioNode.
- Primary alarm with the global Alarm Correlation identifier specified.

### Example of Multiple Correlation UUIDs Associated with a Secondary Alarm

Correlation UUIDs from multiple primary alarms associated with a secondary alarm from vRC in EUTRANCellFDD:

```

{
  "C": [{
    "I": "201f0123-88ca-23a2-7451-8B5872ac457b",
    "n": "vRC"
  }],
  "S": [
    "81d4fae-7dec-11d0-a765-00a0c91e6bf6",
    "f91a6e32-e523-b217-7C3912ad3012"
  ]
}

```



An example of the main alarm fields, including the targetAdditionalInformation name=value object and containing the JSON object represented in string format, is shown:

```
Specific Problem: "Service Degraded"
Severity: Minor
EventType: QUALITY_OF_SERVICE_ALARM
ProbableCause: PERFORMANCE_DEGRADED
AdditionalText: ""
targetAdditionalInformation: CI = {"C": [{"I": "201f0123-88ca-23a2-7451-8B5872ac →
457b", "n": "vRC"}], "S": ["81d4fae-7dec-11d0-a765-00a0c91e6bf6", "f91a6e32-e523-b2 →
17-7C3912ad3012"]};
```

The targetAdditionalInformation string indicates that the alarm is for a:

- Compute resource with the global Identifier specified and a locally unique Name of vRC.
- Secondary alarm with the correlation UUIDs associated with two primary alarms.

### Example of Performance Alarm That Is Not Primary and Not Secondary

Performance Alarm from ENM with compute-correlation info for three compute-hosts that make up a VNF.

This is a performance alarm raised by ENM on behalf of ENIQ. It does not have the concept of primary or secondary, so this information is not present.

```
{
  "C": [{
    "I": "5ddff111-3301-40c8-a1fe-4d4e0084be0b",
    "n":
      "RCRM"
  },
  {
    "I": "4ccff111-3303-40c8-a1fe-4d4e0084be0c",
    "n": "INSE1"
  },
  {
    "I": "3bbff111-3305-40c8-a1fe-4D4e0084be0d",
```



```
        "n": "RCE1"
      }
    ]
  }
```

Stripped of spaces, targetAdditionalInformation is expressed as:

```
CI = {"C": [{"I": "5ddff111-3301-40c8-a1fe-4d4e0084be0b", "n": "RCRM"}, {"I": "4cc →
ff111-3303-40c8-a1fe-4d4e0084be0c", "n": "INSE1"}, {"I": "3bbff111-3305-40c8-a1fe- →
4D4e0084be0d", "n": "RCE1"}]}
```

### Unreachable IP Address

Unreachable IP address is a string.

Value string format = Value of the Unreachable IP address can be either an IPv4 or IPv6 address, for which the format is already defined by RFC (see <https://tools.ietf.org/html/rfc4291> for IPv6).

Example:

```
UNREACHABLEIP = 159.107.199.199;
```



## 10 ENM CLI Commands for SNMP Subscriptions Configuration and Management

The *NbiFmSnmpConfig\_Administrator* user role performs the following actions on the NMS subscriptions:

- Create.
- Delete.
- List.
- Modify administrative state.
- Display parameters.

The *NbiFmSnmpConfig\_Operator* user role performs the following actions on the NMS subscriptions:

- List.
- Display parameters.

### Create

The subcommand `create` is used to create subscriptions.

The positional arguments `v2c` and `v3` refer to the different protocol types and follow the keyword `create`.

In `snmpv3` protocol type the possible security level (`nosec`, `authnopriv`, `authpriv`) is specified as a second positional argument that follows `v3`.

### Delete

The subcommand `delete` is used to delete subscriptions.

### Set

The subcommand `set` is used to modify the subscriptions parameters.

### Get

The subcommand `get` is to list subscription names and to display the subscription parameters, as follows.

- To list subscriptions names, the positional\_arg keyword to be used is `nmslist`.



- To display the subscription parameters, the `positional_arg` keyword to be used is `target`.

### Subscription Name

A subscription is identified by a name during the creation and specified as a string value of the `named_param` name (`--name` or `-n`). The `named_param` is mandatory must be specified when the user wants to identify a particular NMS subscription.

Once created a subscription, further creations of subscriptions with the same name are rejected. Subscription names are case-sensitive.

### Subscription Address

The IP address where the NBI sends traps is identified by the `named_param` `address` (`--address` or `-ip`). The `named_param` must be specified at creation as second parameter in the command line and cannot be later changed.

### Subscription UDP Port

The UDP port where the NBI sends traps is identified by the `named_param` `port` (`--port` or `-p`), which is not mandatory.

Default value is 162. The `named_param` can be specified at creation and cannot be later changed.

### Subscription Administrative State

The administrative state of the subscription is identified by the `named_param` `adminstate` (`--adminstate` or `-as`), which is not mandatory.

Allowed values are, as follows:

- `locked`
- `unlocked`

If `locked`, no SNMP Traps are forwarded for the specified NMS subscription. Default value is `unlocked`.

### Subscription Automatic Suspension Administrative State

The automatic suspension administrative state is identified by the `named_param` `suspensionadministrativestate` (`--suspensionadministrativestate` or `-sp`), which is not mandatory.

Allowed values are, as follows:

- `disabled` (default, automatic suspension is not active).



- enabled (automatic suspension is active).

## 10.1 ENM CLI Syntax for SNMPv2c

At SNMPv2c subscription creation, the following mandatory parameter is needed:

### **community** (**--community**; **-cm**)

The community is used for read and write operations; all Protocol DataUnits (PDUs) including different communities are discarded. A community is unique and cannot be shared among different subscriptions. The parameter must be specified as third named parameter, after the address, and cannot be further changed.

```
fmsnmp create v2c -name <name> --address <ip addr> --community <community> [--adminstate <adminstate>] [--port <port>] [--suspensionadministrativestate <spstate>]
```

Using the short form:

```
fmsnmp create v2c -n <name> -ip <ip addr> -cm <community> [-as <adminstate>] [-p <port>] [-sp <spstate>]
```

### Example 5 Create a SNMPv2c NMS Subscription

```
fmsnmp create v2c -n <name> -ip <ipaddress> -cm <community> -p <port> -as unlocked
```

### Example 6 Create a SNMPv2c NMS Subscription with Automatic Suspension Enabled

```
fmsnmp create v2c -n <name> -ip <ipaddress> -cm <community> -p <port> -as unlocked -sp enabled
```

### Example 7 Modify a SNMPv2c NMS Subscription Administrative State

```
fmsnmp set target -n <name> -as locked
```

### Example 8 Delete a SNMPv2c NMS Subscription

```
fmsnmp delete -n <name>
```



### Example 9 Display a SNMPv2c NMS Subscription Attributes

```
fmsnmp get target -n <name>
```

### Example 10 Display the List of All Configured SNMPv2c NMS Subscriptions

```
fmsnmp get nmslist
```

## 10.2 ENM CLI Syntax for SNMPv3

At SNMPv3 subscription creation the following mandatory parameters are needed:

#### **user (--user; -u)**

The SNMPv3 username for the SNMPv3 message. A user is unique and cannot be shared in different subscriptions. The user must be previously created and assigned to the *NbiFmSnmpManager* role. The parameter is mandatory to include the user in the SNMPv3 packets for read and write operations. Any deviation from the rule causes an error response.

#### **authkey (--authkey; -ak)**

The authentication key used to sign SNMP notifications. Authentication is performed by using the `authkey` of a user to sign the messages being sent. The length of the key must be at least eight characters. The parameter is mandatory if security mode was specified as `authnopriv` or `authpriv` at creation.

#### **privkey (--privkey; -pk)**

The privacy key used for encryption. Encryption is performed by using the `privkey` of a user to encrypt the data portion of the message being sent. The length of the key must be at least eight characters. The parameter is mandatory if security mode was specified as `authpriv` at creation.

At SNMPv3 subscription creation, the following additional parameters, which are not mandatory, are needed:

#### **authprotocol (--authprotocol; -ap)**

The used authentication protocol. The parameter is needed if authorization is enabled.

Allowed values are, as follows:

- MD5 (default)
- SHA

**privacyprotocol (--privprotocol; -pp)**

The used privacy protocol. The parameter is needed if privacy is enabled.

Allowed values are, as follows:

- DES (default)
- 3DES
- AES
- AES128
- AES192
- AES256

```
fmsnmp create v3 nsec -name <name> --address <ip addr> --user <user> [--adminstate <adminstate>] [--port <port>] [--suspensionadministrativestate <spstate>]
```

Using the short form:

```
fmsnmp create v3 nsec -n <name> -ip <ip addr> -u <user> [-as <adminstate>] [-p <port>] [-sp <spstate>]
```

**Example 11 Create a SNMPv3 NMS Subscription**

```
fmsnmp create v3 nsec -n <name> -ip <ipaddress> -u <user> -p <port> -as unlocked
```

**Example 12 Create a SNMPv3 NMS Subscription with Security Level Auth\_no\_priv**

```
fmsnmp create v3 authnopriv -n <name> -ip <ipaddress> -u <user> -ak <authKey> -p <port> -as unlocked
```

**Example 13 Create a SNMPv3 NMS Subscription with Security Level Authpriv**

```
fmsnmp create v3 authpriv -n <name> -ip <ipaddress> -u <user> -ak <authKey> -pk <privKey> -p <port> -as unlocked
```

**Example 14 Modify a SNMPv3 NMS Subscription Administrative State**

```
fmsnmp set target -n <name> -as locked
```



#### Example 15 Delete a SNMPv3 NMS Subscription

```
fmsnmp delete target -n <name>
```

#### Example 16 Display a SNMPv3 NMS Subscription Attributes

```
fmsnmp get target -n <name>
```



# 11 ENM CLI Commands for Basic FM SNMP NBI Configuration and Management

The administrative state of the NBI can be locked meaning that no SNMP traps are generated or unlocked standing for SNMP NBI normal behavior.

A user with `NbiFmSnmpConfig_Administrator` role is allowed to set and display the administrative state of the NBI by ENM CLI.

A user with `NbiFmSnmpConfig_Operator` role is allowed to display the administrative state of the NBI by ENM CLI.

## Syntax

The allowed subcommands are, as follows:

- set to configure.
- get to display.

The positional\_arg keyword to be used is agent.

The named\_params and values are `--adminstate locked` or `--adminstate unlocked`.

Using the short form `-as locked` or `-as unlocked`

The default is unlocked.

The set command syntax:

```
fmsnmp set agent [--adminstate <locked/unlocked>]
```

Using the short form:

```
fmsnmp set agent [-as <locked/unlocked>]
```

The get command syntax:

```
fmsnmp get agent
```

Example 17 Stop the NBI SNMP Agent:

```
fmsnmp set agent --adminstate locked
```



Example 18 Resume the NBI SNMP Agent

```
fmsnmp set agent --adminstate unlocked
```

Example 19 Display All the NBI SNMP Agent Parameters

```
fmsnmp get agent
```



## 12 ENM FM SNMP NBI and OSS-RC FM SNMP NBI Differences

### Alarm and Alert Synchronization

OSS-RC FM SNMP NBI supports the feature, as follows:

1. NMS sends a `getNext` request to a dedicated OID (`eriAlarmActiveMinorType = 1.3.6.1.4.1.193.183.4.1.3.5.1.3`).
2. OSS-RC NBI returns a `getNextResponse` stating `Synch` is initiated.
3. OSS-RC NBI returns a `getResponse` stream to port 162:
  - One `getResponse` for each alarm in the `eriAlarmActiveAlarmTable`.
  - One `getResponse` for each alert in the `eriAlarmAlertTable`.
  - The `getResponse` are formatted as traps that contain just the `varbinds` of the trap and not all the fields of the corresponding `AlarmActiveTable` row.
  - If there are acknowledged alarms, an `eriAlarmAppendInfo` is sent as well as the original trap.

The string `NULL SYNCH` is appended to the `SpecificProblem` attribute to each trap.

### Heartbeat

NMS clients using OSS-RC FM SNMP NBI must implement the heartbeat in pull mode by polling OID number `1.3.6.1.1.1.0`. All other OID numbers are treated as non-readable. The OID number `1.3.6.1.1.1.0` does not belong to Ericsson Alarm MIB and so it cannot be used in SNMP FM NBI in ENM.

NMS clients using ENM FM SNMP NBI are allowed to implement the heartbeat in pull mode by polling OIDs related to any scalar value defined as readable in the `EricssonAlarmMib`.

### Alarm and Alert Comment

In OSS-RC, once the alarms and alerts comments are inserted, the trap is sent another time without any indications about the modified comment.

In ENM, once an alarm or alert comment is inserted then `eriAlarmAppendAlertInfo (.1.3.6.1.4.1.193.183.4.2.0.15)` trap is sent. The



comment is included in `eriAlarmNObjAdditionalText` (1.3.6.1.4.1.193.183.4.1.2.1)

### Alarm and Alert Sequence

In ENM FM SNMP NBI, the couples `eriAlarmActiveMajorType/eriAlarmActiveMinorType` and `eriAlarmAlertMajorType/eriAlarmAlertMinorType` attributes are used to carry the eventPoId as follows:

- The attribute `eriAlarmActiveMajorType/eriAlarmAlertMajorType` is used to carry upper 32 bits of eventPoId.
- The attribute `eriAlarmActiveMinorType/eriAlarmAlertMinorType` is used to carry lower 32 bits of eventPoId.

In OSS-RC FM SNMP NBI, the Sequence Number management is, as follows:

1. Alarms raise:  
`eriAlarmActiveMajorType=eriAlarmActiveMinorType=<sequence number>`
2. Alarms clear: `eriAlarmActiveMinorType=<original sequence number>; eriAlarmActiveMajorType=<new sequence number for the clear>`
3. Acknowledge or unacknowledge AppendInfo:  
`eriAlarmActiveMajorType=eriAlarmActiveMinorType=<original sequence number>`

**Note:** The same behavior can be applied to alerts (`eriAlarmAlertMajorType` and `eriAlarmAlertMinorType` attributes).

The `eriAlarmActiveLastSequenceNo` and `eriAlarmAlertLastSequenceNo` attributes are always set to zero.

ENM alarm synchronization is different from OSS-RC, for more information see [Alarm and Alert Synchronization](#) on page 7.

The UDP port where OSS-RC SNMP FM NBI is listening to synchronous requests from NMS is different from the port used in ENM.

### General Notes

- When for any reason the NMS is suspended (for example when the NMS is set in locked state) then SNMP traps are NOT pushed to NMS. Any alarm synchronization or get action from NMS can result in inconsistent information sent by NBI and received by the NMS.
- Since the identification of the NMS requesting a get or a synchronization to NBI is based on Community (SNMPv2c) or Username-password (SNMPv3), each NMS must have different Community and different usernames.



- Acknowledge and unacknowledge operations from ENM operator are possible. The behavior in terms of SNMP traps sent to NMS is Ericsson proprietary. This is done similarly to what is done in OSS-RC.
- If an alarm (or alert) has a problem text exceeding 256 characters, then two traps are sent. The second trap is `eriAlarmAppendInfo` for alarm and `eriAlarmAppendAlertInfo` for alert similarly to what is done in OSS-RC.



## Reference List

- [1] *ENM Network Integration Guide*, 1/102 72-aom 901 151 Uen
- [2] *Supplementary Documentation for ENM FM SNMP Northbound Interface Integration Programmers Guide*, 190 89-lzn7080882-12 Uen
- [3] ENM Monitoring System Administrator Guide, 1/1543-aom 901 151-2