

# Ericsson Alarm Interface

## INTERWORK DESCRIPTION

**Copyright**

© Ericsson AB 2017,2019. 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>Understanding the Ericsson Alarm Interface</b>	<b>1</b>
1.1	The Ericsson Alarm Interface in a Management Solution	1
1.2	Key Features for Ericsson Alarm Interface	2
1.3	Ericsson Alarm Interface Concepts	2
<b>2</b>	<b>Alarm Interface Capabilities NBI Exposure</b>	<b>5</b>
<b>3</b>	<b>Ericsson Alarm MIB Modules Overview</b>	<b>7</b>
<b>4</b>	<b>Alarm Type Information Mapping to NBI</b>	<b>9</b>
<b>5</b>	<b>Alarm Instances Information Mapping to Notification</b>	<b>11</b>
<b>6</b>	<b>Heartbeat Information Mapping to NBI</b>	<b>13</b>
<b>7</b>	<b>Mapping Between IETF MIB Framework and ECLI/NETCONF Model</b>	<b>15</b>
<b>8</b>	<b>Alarm Log and Alert Format</b>	<b>17</b>





# 1 Understanding the Ericsson Alarm Interface

## 1.1 The Ericsson Alarm Interface in a Management Solution

The Ericsson Alarm Interface provides a unified way to expose the alarm status of an Ericsson Managed Element (ME) to a management system or a Command-Line Interface (CLI) client.

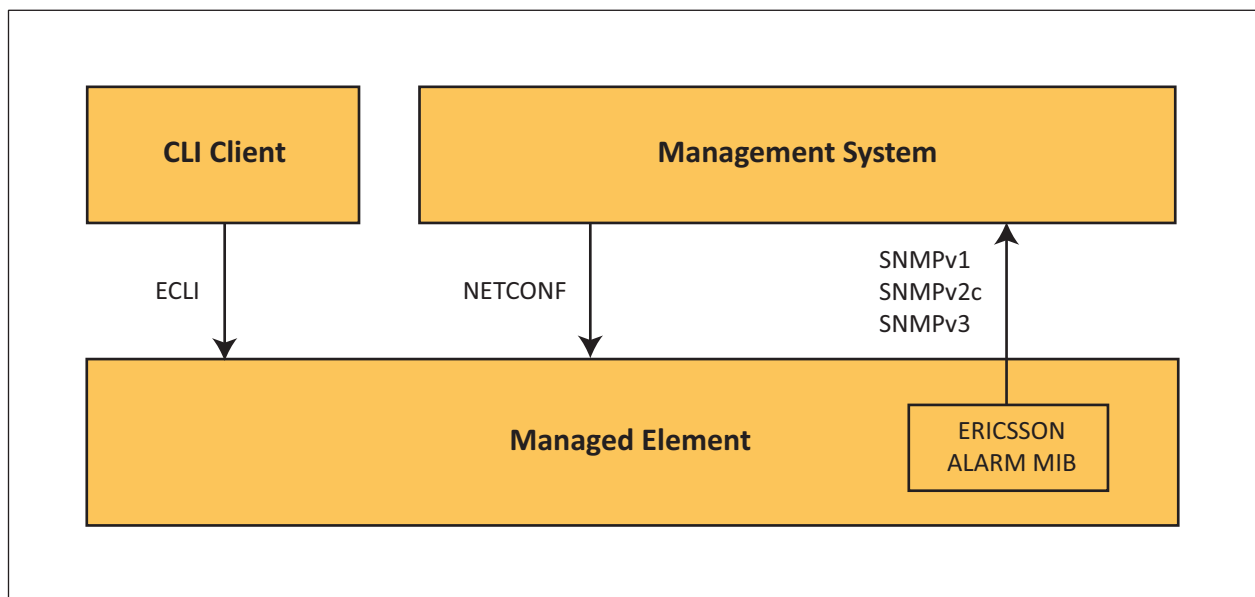


Figure 1 Use of the Ericsson Alarm Interface

The Ericsson Alarm Interface consists of alarm handling capabilities exposed over the following interfaces:

- Ericsson Command-Line Interface – A Man–Machine Interface realized as a CLI reachable through the SSH
- Ericsson NETCONF Interface – A Machine-Machine Interface based on the IETF NETCONF standard (RFC 4741) with SSH as transport (RFC 4742)
- Ericsson Alarm MIB over SNMPv1, SNMPv2, and SNMPv3 – SNMP is used to report failures to the management system

Alarm information is available in the ECLI, NETCONF, and SNMP interfaces and in log files.

For a complete description about the ECLI, refer to [Ericsson Command-Line Interface](#).

For a complete description of the Ericsson Network Configuration (NETCONF) interface, refer to [Ericsson NETCONF Interface](#).



## 1.2 Key Features for Ericsson Alarm Interface

Table 1 Key Features for Ericsson Alarm Interface

Feature	Description
Stateful and stateless notifications	Alarm state changes and alert are spontaneously notified over SNMP.
Support for multiple notification receivers	Multiple SNMP targets can be defined to receive SNMP notifications.
Active alarm list	The active alarm list is exposed over SNMP and NETCONF to enable a management system to synchronize or resynchronize the alarm status with the managed element. The active alarm list is also exposed over the ECLI.
Alarm and alert history	Alarm and alert logs keep in a chronological order all alarm state changes and alerts.
Alarm interface monitoring	A heartbeat mechanism makes it possible to detect problems with the alarm interface (interface down or loss of individual alarms).
Strict alarm identification	An alarm instance is unambiguously identified based on a combination of its source and its alarm type.
Operating Instruction identification	Each alarm type has an Operating Instruction document titled according to its specific problem value.
Alarm filtering	When an alarm is detected as toggling between raised and clear state at a too high frequency, the message "The alarm is toggling" is appended to the additional text. In such situation, the individual state changes are no longer reported until the alarm is cleared.
System UUID identification	An alarm instance may carry the system UUID of the host where it was detected as part of additionalText or additionalInfo.

## 1.3 Ericsson Alarm Interface Concepts

An alarm is an indication of a fault that has occurred or is likely to occur and that requires human intervention and action. An alarm is stateful which means that the state can either be raised or cleared. The severity level can be changed by the system by either increase or decrease the severity of the alarm. An alarm is cleared when the underlying fault no longer exists.

An alert is a stateless alarm, that is, an alarm that can only have the raised state.

An alarm notification is an alarm state change sent over SNMP. Alarm notifications are sent when an alarm is initially raised, updates its severity, updates its additional text, and is eventually cleared.

Alarm types define what alarms a ME can actually report. An alarm is uniquely identified by the combination of its source with an alarm type. An alarm type is uniquely identified based on the {major type, minor type} combination or the {specific problem, probable cause, event type} combination.



Table 2 Alarm Type to NBI Mapping

Concept	NETCONF/ECLI
Alarm type	FmAlarmType
major type	majorType
minor type	minorType
Specific Problem	specificProblem
Event Type	eventType
Probable Cause	probableCause
Additional Text	additionalText
Perceived Severity	defaultSeverity
	configuredSeverity
Managed Object Class	moClasses
Alarm/Alert	isStateFul

### 1.3.1 Alarm List Rebuilt Event

This event is reported as a SNMP notification when the ME active alarm list reaches a stable situation after a restart or after an ME internal audit process. It is an indication to the SNMP targets to perform the following:

- Retrieve the ME active alarm list.
- Compare the retrieved list with their own list of active alarms.
- Appropriately handle any change between the two lists.

### 1.3.2 Heartbeat Event

Alarms and alerts may not always reach a management system because of network issues. Such situation can lead to longer service deterioration or unavailability. The heartbeat mechanism enables a management system to detect quickly alarm/alert loss so the ME is not left unattended during a too long period.

Two heartbeat mechanisms are supported: pull heartbeat and push heartbeat.

The heartbeat mechanism enables a management system to detect quickly if some alarms or alerts have been lost.

With the pull mechanism, a management system regularly polls the following information on the ME:

- The last event time stamp, that is, when any alarm was last changed.
- The last used sequence number for an alarm state change notification.



A management system can pull this information using NETCONF or SNMP. The ECLI is not recommended in this case.

With the push mechanism, the ME instead reports Heartbeat events to a management system at a regular time interval. Heartbeat events contain the same information as in the pull mechanism. The push mechanism is supported only over SNMP using an SNMP notification. It can therefore only be used by management systems acting as SNMP targets.

### **1.3.3 Other Events**

Reporting of all other events is done over NETCONF notifications and is not handled by FM.





## 2 Alarm Interface Capabilities NBI Exposure

Table 3 Alarm Interface Capabilities NBI Exposure

Capability	Exposed over SNMP	Exposed over NETCONF/ECLI
SNMP Targets	Yes	Yes
Alarm Notification	Yes	No
Active Alarm List	Yes	Yes
Alarm Types	No	Yes
Heartbeat	Yes (configuration & notifications)	Yes (configuration only)
Alarm/alert Log	No	Yes
Strict Alarm Identification	Yes	Yes
Alarm List Rebuilt Notification	Yes	No
System UUID	Yes (if included in additional text)	Yes





## 3 Ericsson Alarm MIB Modules Overview

The Ericsson Alarm MIB consists of five different MIB modules.

General imported modules:

— Ericsson-TOP-MIB

The Ericsson enterprise OID.

— Ericsson-TC-MIB

Textual convention for Managed Objects and other Ericsson-specific generic textual conventions.

Alarm MIB modules:

— Ericsson-ALARM-MIB

The main MIB with alarm tables, alarm notifications, heartbeat, and so on.

**Note:** The Ericsson Alarm Management Information Base (MIB) is supported with the following exemptions:

- The alert table, `eriAlarmAlertTable`, is always empty.
- `eriAlarmActiveTableURL` always returns an empty string.
- `eriAlarmAlertTableURL` always returns an empty string.

— Ericsson-ALARM-PC-MIB

Separate module for probable cause.

— ERICSSON-ALARM-TC-MIB

Textual convention used in the alarm MIBs.





## 4 Alarm Type Information Mapping to NBI

Alarm information is available in the SNMP, CLI, and NETCONF interfaces and in log files, as described in this section.

A summary of the number of active alarms is available as follows:

- `eriAlarmActiveAlarms`, `eriAlarmSumCritical`, `eriAlarmSumMajor`, `eriAlarmSumMinor`, and `eriAlarmSumWarning` SNMP objects of the `ERICSSON-ALARM-MIB`

**Note:** The number of alarms of indeterminate severity is available in the SNMP as `eriAlarmSumIndeterminate` only.

- Attributes `totalActive`, `sumCritical`, `sumMajor`, `sumMinor`, and `sumWarning` of MO Fm in the COM CLI and NETCONF

**Note:** While reading the Fm subtree in a particular transaction, the statistics attributes in Fm (that is `totalActive`, `sumMajor`, `sumMinor`, and so on) cannot always be in sync with the actual FmAlarm instances.

Information about the last issued alarm is available as follows:

- SNMP objects `eriAlarmActiveLastSequenceNo` and `eriAlarmActiveLastChanged` of the `ERICSSON-ALARM-MIB`
- Attributes `lastChanged` and `lastSequenceNo` of MO Fm in the COM CLI and NETCONF

Column `FmAlarmType` shows what static alarm model information is visible over NETCONF and the ECLI. Column `FmAlarm` shows what active alarm information is visible over NETCONF and the ECLI. Column `ERICSSON-ALARM-MIB` is mainly for reference and indicates how the information is mapped on the SNMP interface.

Table 4 Alarm Information to NBI Mapping

Concept	NETCONF/ECLI	Ericsson Alarm MIB
Alarm Information	FmAlarm	<code>eriAlarmActiveAlarmEntry</code>
major type	<code>majorType</code>	<code>eriAlarmActiveMajorType</code>
minor type	<code>minorType</code>	<code>eriAlarmActiveMinorType</code>
Managed Object Instance/Source	<code>source</code>	<code>eriAlarmActiveManagedObject</code>
SpecificProblem	<code>specificProblem</code>	<code>eriAlarmActiveSpecificProblem</code>
Sequence Number	<code>sequenceNumber</code>	-
Event Type	<code>eventType</code>	<code>eriAlarmActiveEventType</code>
Event Time	<code>lastEventTime</code>	<code>eriAlarmActiveEventTime</code>
Perceived Severity	<code>activeSeverity</code>	<code>eriAlarmActiveSeverity</code>
Probable Cause	<code>probableCause</code>	<code>eriAlarmActiveProbableCause</code>



Table 4 Alarm Information to NBI Mapping

Concept	NETCONF/ECLI	Ericsson Alarm MIB
Additional Text	additionalText	eriAlarmActiveAdditionalText
Additional Info	additionalInfo	eriAlarmActiveAdditionalInfo
Original Additional Text	originalAdditionalText	eriAlarmActiveOrigAdditionalText
Resource Id	-	eriAlarmActiveResourceId
Original Event time	originalEventTime	eriAlarmActiveOriginalEventTime
Original severity	originalSeverity	eriAlarmActiveOriginalSeverity
number of active alarms	totalActive	eriAlarmActiveNumber
number of active critical alarms	sumCritical	eriAlarmSumCritical
number of active major alarms	sumMajor	eriAlarmSumMajor
number of active minor alarms	sumMinor	eriAlarmSumMinor
number of active warning alarms	sumWarning	eriAlarmSumWarning
number of indeterminate alarms	-	eriAlarmSumIndeterminate
Heartbeat Interval	heartbeatInterval	eriAlarmHeartbeatInterval
Latest Sequence Number	lastSequenceNo	eriAlarmActiveLastSequenceNo
		eriAlarmAlertLastSequenceNo
number of alerts	-	eriAlarmAlertNumber



## 5 Alarm Instances Information Mapping to Notification

Table 5 Summary Alarm Instances Available in Multiple Interfaces

Concept	Ericsson Alarm MIB			
Alarm state change	eriAlarm* Notification	eriAlarm*Alert Notification	eriAlarmAppendInfo Notification	eriAlarmAppendAlert Info Notification
Managed Object Instance/Source	eriAlarmActiveManagedObject	eriAlarmAlertManagedObject	eriAlarmActiveManagedObject	eriAlarmAlertManagedObject
major type	eriAlarmActiveMajorType	eriAlarmAlertMajorType	eriAlarmActiveMajorType	eriAlarmAlertMajorType
minor type	eriAlarmActiveMinorType	eriAlarmAlertMinorType	eriAlarmActiveMinorType	eriAlarmAlertMinorType
Specific Problem	eriAlarmActiveSpecificProblem	eriAlarmAlertSpecificProblem	-	-
Latest sequence Number	eriAlarmActiveLastSequenceNo	eriAlarmAlertLastSequenceNo	-	-
Event Type	eriAlarmActiveEventType	eriAlarmAlertEventType	-	-
Event Time	eriAlarmActiveEventTime	eriAlarmAlertEventTime	-	-
Probable cause	eriAlarmActiveProbableCause	eriAlarmAlertProbableCause	-	-
Additional text	eriAlarmNObjAdditionalText	eriAlarmNObjAdditionalText	eriAlarmNObjAdditionalText	eriAlarmNObjAdditionalText
	eriAlarmNObjMoreAdditionalText	eriAlarmNObjMoreAdditionalText	-	-
Resource ID	eriAlarmNObjResourceId	eriAlarmNObjResourceId	eriAlarmActiveResourceId	eriAlarmAlertResourceId
Additional Info	eriAlarmNObjAdditionalInfo	eriAlarmNObjAdditionalInfo	eriAlarmNObjAdditionalInfo	eriAlarmNObjAdditionalInfo
	eriAlarmNObjMoreAdditionalInfo	eriAlarmNObjMoreAdditionalInfo	-	-
New or updated alarm	eriAlarmNObjRecordType	-	-	-







## 6 Heartbeat Information Mapping to NBI

In Table 6, the columns NETCONF/ECLI and ERICSSON ALARM MIB show what information a management system must access to implement a heartbeat pull over NETCONF and SNMP, respectively.

Table 6 Heartbeat Information Mapping to NBI

Heartbeat Information	NETCONF/ECLI	Ericsson Alarm MIB
Latest time stamp	lastChanged	eriAlarmActiveLastChanged
		eriAlarmAlertLastChanged
Latest Sequence Number	lastSequenceNo	eriAlarmActiveLastSequenceNo
		eriAlarmAlertLastSequenceNo

Events `AlarmListRebuilt` and `HeartBeat` are reported using `NOTIFICATION-TYPE` `eriAlarmHeartBeatNotif` and `eriAlarmAlarmListRebuilt`, respectively, according to `ERICSSON-ALARM-MIB`.

The heartbeat interval of the COM SNMP alarm interface is configurable as follows:

- SNMP object `eriAlarmHbInterval` of the `ERICSSON-ALARM-MIB`
- Attribute `heartbeatInterval` of the Fm Managed Object (MO) in the COM CLI or NETCONF





## 7 Mapping Between IETF MIB Framework and ECLI/NETCONF Model

Table 7 Mapping IETF MIB Framework and ECLI/NETCONF Model

Concept	NETCONF/ECLI				SNMP Object Name	SNMP OID
SNMP target	SnmpTargetV1	SnmpTargetV2C	SnmpTargetV3	SnmpTargetV3Dtls	SnmpTargetAddrEntry	1.3.6.1.6.3.12.1.2.1
	address	address	address	address	snmpTargetAddrTAddress	1.3.6.1.6.3.12.1.2.1.3
	port	port	port	port		
	-	informTimeout	informTimeout	informTimeout	snmpTargetAddrTimeout	1.3.6.1.6.3.12.1.2.1.4
	-	informRetryCount	informRetryCount	informRetryCount	snmpTargetAddrRetryCount	1.3.6.1.6.3.12.1.2.1.5
	community	community	user	user	snmpTargetParamsSecurityName	1.3.6.1.6.3.12.1.3.1.4
	-	-	snmpSecurityLevel	-	snmpTargetParamsSecurityLevel	1.3.6.1.6.3.12.1.3.1.5
			authProtocol	-	usmUserAuthProtocol	1.3.6.1.6.3.15.1.2.2.1.5
			privProtocol	-	usmUserPrivProtocol	1.3.6.1.6.3.15.1.2.2.1.8
SNMP view	SnmpViewV1	SnmpViewV2C	SnmpViewV3		snmpVacmMIB	1.3.6.1.6.3.16





## 8 Alarm Log and Alert Format

The Alarm Log and Alert Log files are exposed by File Management as two file groups named AlarmLogs and AlertLogs, respectively.

Alarm Log files are rotated. Internal limits set the maximum file size and the maximum number of files. When the maximum number of files is exceeded, the oldest file is deleted automatically. The same behavior applies to Alert Log files.

The Alarm Log and Alert Log records are encoded in a common XML format. The log record consists of two elements. The first element indicates the time the record is logged. The second element contains specific information about the alarm or alert and is formatted as a semicolon-separated string.

Table 8 Alarm and Alert Log Record Format

Tags and Information	Description
<FmLogRecord>	Log record start
<LogTimestamp>	
Time stamp tag	The time the record is logged Format: <YYYY-MM-DDThh:mm:ss>Z
</LogTimestamp>	
<Alarm> or <Alert>	
Alarm-specific or alert-specific information	Formatted as a string separated by semicolons containing the following tokens:  1: stateful 2: eventTime 3: source 4: majorType 5: minorType 6: specificProblem 7: probableCause 8: severity 9: additionalText 10: sequenceNumber 11: eventType 12: originalEventTime 13: originalSeverity 14: originalAdditionalText 15: originalSequenceNumber 16: additionalInfoSize 17: additionalInfo (name;value)



Table 8 Alarm and Alert Log Record Format

Tags and Information	Description
</Alarm> or </Alert>	
</FmLogRecord>	Log record end

**Note:** Possible values of stateful token are

- 0 - indicates Alert
- 1 - indicates Alarm

```
<FmLogRecord>
  <LogTimestamp>2019-05-21T10:09:05.858+02:00</LogTimestamp>
  <Alarm>1;2019-05-21T10:09:04.661+02:00;ManagedElement=1,⇒
  SaAmfApplication.safApp=ERIC-ComSa,SaAmfSI.safSi=2N;18568;131077;⇒
  COM SA, AMF SI Unassigned;418;CLEARED;Previous raised alarm of⇒
  safSi=2N,safApp=ERIC-ComSa is now cleared;10;PROCESSINGERRORALARM;⇒
  2019-05-21T10:08:59.578+02:00;MAJOR;SI designated by safSi=2N,⇒
  safApp=ERIC-ComSa has no current active assignments to any SU;9;0</Alarm>
</FmLogRecord>
<FmLogRecord>
  <LogTimestamp>2019-05-21T10:16:46.637+02:00</LogTimestamp>
  <Alarm>1;2019-05-21T10:16:46.583+02:00;ManagedElement=1,⇒
  SaAmfApplication.safApp=ERIC-ComSa,SaAmfSI.safSi=2N;18568;⇒
  131077;COM SA, AMF SI Unassigned;418;MAJOR;SI designated by ⇒
  safSi=2N,safApp=ERIC-ComSa has no current active assignments ⇒
  to any SU;11;PROCESSINGERRORALARM;2019-05-21T10:16:46.583+02:00;⇒
  MAJOR;SI designated by safSi=2N,safApp=ERIC-ComSa has no current⇒
  active assignments to any SU;11;0</Alarm>
</FmLogRecord>
```

#### Example 1 Log Records in Alarm Log