

# Fault Management

---

## DESCRIPTION

## **Copyright**

© Ericsson AB 2017, 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

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Functions and Concepts</b>	<b>3</b>
2.1	Types of Operation	8
2.2	Ericsson Definition of 3GPP Perceived Severity Values	9
<b>3</b>	<b>Managed Object Model</b>	<b>11</b>
3.1	Managed Object Model — SNMP	11
3.2	Managed Object Model — Fault Management	12
<b>4</b>	<b>Configuration Management</b>	<b>15</b>
<b>5</b>	<b>IPWorks Application Fault Management</b>	<b>17</b>
<b>6</b>	<b>File Management</b>	<b>19</b>
<b>7</b>	<b>Fault Management Function</b>	<b>21</b>
	<b>Reference List</b>	<b>23</b>





# 1 Introduction

This document provides an overview of the management model and concepts associated with the Fault Management (FM) managed area.

A managed area is represented by a group of Managed Object Classes (MOCs) within the Managed Object Model (MOM).





## 2 Functions and Concepts

FM detects unexpected Managed Element (ME) behaviors and malfunctions requiring corrective actions that cannot be performed by the ME. FM raises alarms in such situations to get the user attention.

FM provides a management interface covering the following:

- Reporting of alarms through Simple Network Management Protocol (SNMP) notifications
- Enabling matching between an alarm and its corresponding alarm Operating Instructions document
- Management of targets, heartbeat, and alarm information

### FM Interfaces

An overview of the FM interfaces is shown in Figure 1.

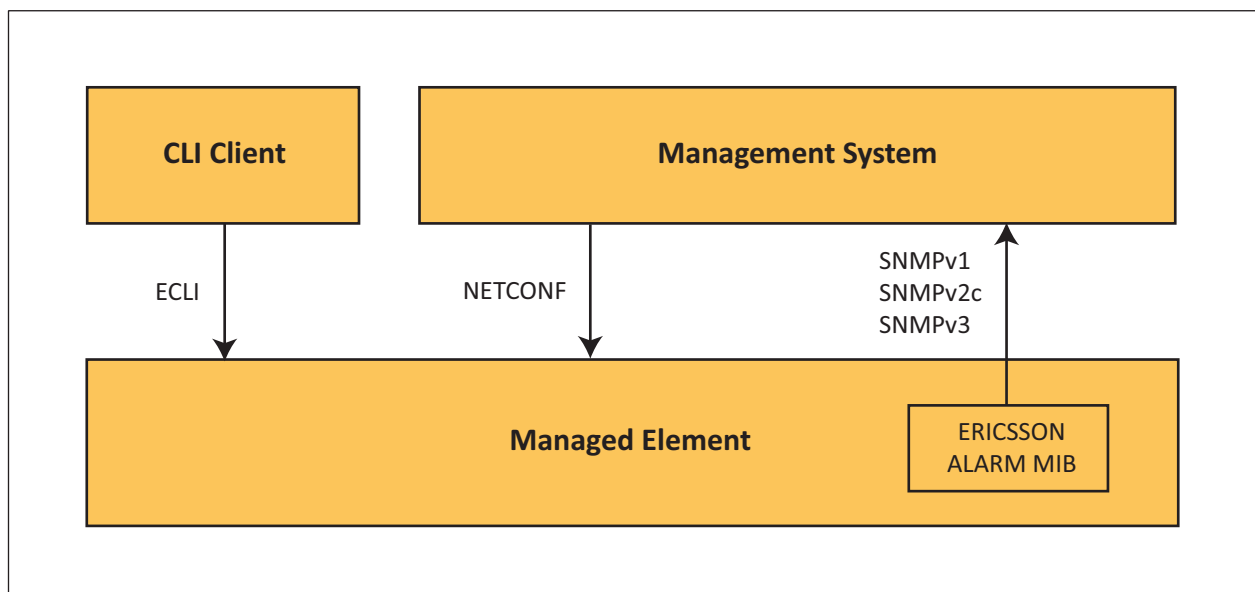


Figure 1 Fault Management Interfaces

The FM interfaces are as follows:

- Ericsson Command-Line Interface (ECLI) – A man-machine interface realized as a CLI reachable through the SSH.
- NETCONF – A machine-machine interface based on the IETF NETCONF standard (RFC 4741) with SSH as transport (RFC 4742).



- SNMPv1, SNMPv2c, SNMPv3, and the internal Ericsson standard, the *ERICSSON-ALARM-MIB* – 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.

### **Problem Resolution Workflow**

The problem resolution workflow, shown in Figure 2, consists of the following main steps:

1. When the alarm is noticed and identified based on its information, the user acknowledges it to indicate to other users that the problem is being worked on. Acknowledgment is not an ME functionality and is not further described here.
2. The user identifies the fault by looking at the Specific Problem of the alarm information.
3. The user finds the corresponding alarm Operating Instructions (OPI) document, for example, by performing a search in the Active Library Explorer library. Each alarm has an alarm OPI document titled as the Specific Problem.

The alarm OPI can also be retrieved from the GUI on a management system.

4. To solve the problem, the alarm OPI document is to be used as follows by the user:
  - a In Table 1: Study the possible alarm causes, fault reasons, fault locations, and the potential service impact.
  - b In Table 2: Analyze the alarm information (see the next section of this document). The alarm information is visible over NETCONF and the ECLI.
  - c In Section 2: Execute the procedure to eliminate the problem and eventually clear the alarm.



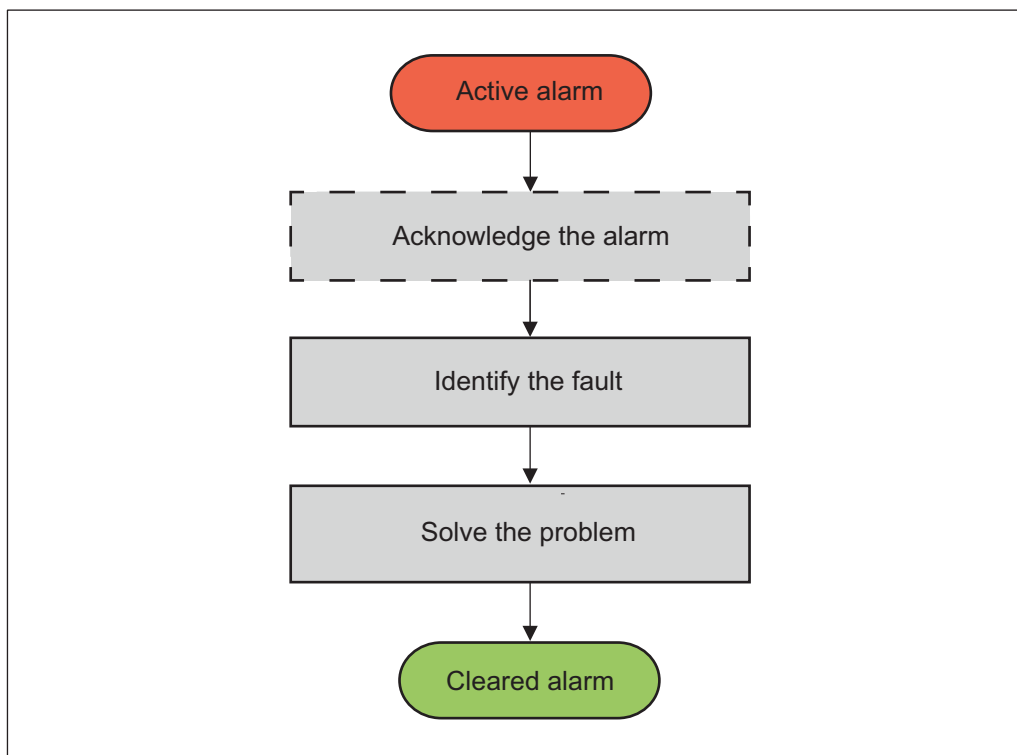


Figure 2 Problem Resolution Workflow

### Alarm Information

An alarm includes the information described in Table 1. The information that is of interest for the user is described in each alarm Operating Instructions document.

Table 1 Alarm Information

Alarm Information	Description
Major Type	The combination of Major Type and Minor Type, which are two numbers, identifies an Alarm Type, which is an alarm category, within the ME. The Alarm Type is the same in different versions of the ME.
Minor Type	
Managed Object Class	Identifies the MOC that the alarm is applicable to and issued from. Applicable only to alarms belonging to a managed area.
Managed Object Instance or Source	The Distinguished Name (DN) of the alarming object. Managed Object (MO) Instance is applicable only to alarms belonging to a managed area, else Source is used.
Specific Problem	Provides further refinement to the information given by Probable Cause and is unique within the ME. Specific Problem is the same in different versions of the ME. The alarm OPI document title exactly matches its value.



Table 1 Alarm Information

Alarm Information	Description
Event Type	The general category for the alarm. The values are defined by ITU-T X.733 and X.736 according to RFC 3877.
Probable Cause	Qualifies and provides further information on the reason for the alarm. The values are defined by ITU-T X.733, X.736, M.3100, and GSM 1211 and are included in <i>ERICSSON-ALARM-PC-MIB</i> .
Additional Text	Provides extra textual information. Normally runtime-related information.
Perceived Severity	Provides guidance on the severity of the problem, that is, possible service impact and urgency to act. The value can be changed owing to deployment scenarios or the operation situation. Perceived Severity is to be interpreted according to the Ericsson definition of the 3GPP® Perceived Severity values, see Section 2.2 Ericsson Definition of 3GPP Perceived Severity Values on page 9.
Event Time	The time when the alarm was updated, that is, the time for the latest alarm information change or severity change.
Sequence Number	Uniquely identifies the corresponding notification sent over the Northbound Interface (NBI). This identifier changes at every notification, that is, every alarm state change.

A user unambiguously identifies an alarm based on the unique combination of the following:

- MO Instance or Source
- Event Type, Probable Cause, and Specific Problem

**Note:** The combination of MO Instance/Source, Major Type, and Minor Type also unambiguously identifies an alarm but on a lower usability level.

### Alarm States

An alarm goes through the following states during its life cycle:

- Raised with initial severity at initial detection of the fault
- Raised with changed severity (not applicable to all Alarm Types)
- Cleared when the fault no longer exists

Depending on the alarm, the cleared state is either reached through an explicit user clearing operation or triggered by the ME. For example, some alarms are triggered by threshold values or specific conditions, and clear automatically when the condition is no longer true.

Alarm state changes are reported in real time as SNMP notifications to management systems listening to such notifications (also known as SNMP



targets). The current ME alarm state is maintained in the active alarm list, which contains only alarms in raised states. Alarms in cleared state are not visible in the active alarm list. All alarm state changes including cleared state are recorded in the Alarm Log.

An alarm is defined as toggling when its raise and clear conditions are met multiple times within an internally defined period. The alarm then remains in raised state and the text `The alarm is currently toggling` is appended to Additional Text.

An alert is a stateless alarm, that is, an alarm that can only have the raised state. As an alarm, an alert has an associated Operating Instructions document and is reported in real time as an SNMP notification. Alerts are recorded in the Alert Log but are not exposed in any list over the NBI.

### Heartbeat Event

The Heartbeat mechanism adds robustness to an FM solution involving the ME and a management system. Heartbeats are used by a management system to monitor the interface over which the alarms or alerts are to be sent. It is because a management system cannot assume that a “silent” ME behaves properly. Heartbeats enable a management system to detect quickly if some alarms or alerts have been lost. They also avoid leaving the ME unattended during a too long period. A loss of alarms can lead to longer service deterioration or unavailability.

Heartbeats can be used with a pull or a push mechanism. 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. For more information, refer to *HeartBeat*.

### Alarm List Rebuilt Event

FM also reports event *AlarmListRebuilt* as an SNMP notification. This event is reported 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:



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

### Other Events

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

### SNMP Targets

Reporting of SNMP notifications to multiple SNMP targets is supported.

## 2.1 Types of Operation

FM supports the following operations:

### SNMP Configuration

- Configure SNMP targets

SNMP targets contain the necessary information to report SNMP notifications to the corresponding management systems. SNMPv1, SNMPv2C, and SNMPv3 targets are supported.

For SNMPv2C and SNMPv3 targets, either an unacknowledged SNMP message mechanism (TRAP) or an acknowledged SNMP message mechanism (INFORM) can be chosen. For SNMPv3 targets, authentication and privacy mechanisms can be configured. SNMPv1 and SNMPv2C targets support only configuration of community as the authentication mechanism. The procedures in *Create SNMPv1 Target*, *Create SNMPv2C Target*, and *Create SNMPv3 Target* provide further details on how to perform these operations.

- Configure SNMP view

An SNMP view can be configured to give read or write access to specific SNMP Object Identifiers (OIDs) to SNMP users. The procedure in *Create SNMP View* provides further details on how to perform this operation.

Even though the SNMP view is an SNMPv3 feature, it can be used to decide what SNMP OID access to allow over SNMPv1, SNMPv2C, and SNMPv3. For SNMPv1 and SNMPv2C views, SNMP users are identified by the SNMP community they use to access the ME over SNMPv1 or SNMPv2C. For SNMPv3 views, SNMP users are identified by the SNMPv3 username they use to access the ME over SNMPv3.



In the absence of a specific configuration, default access is given to the following SNMP Management Information Bases (MIBs):

- *ERICSSON-ALARM-MIB*
- SNMP-FRAMEWORK-MIB (according to RFC 3411)
- MIB-2 (according to RFC 1213)

The default behavior denies access to other SNMP MIBs. An explicit view configuration must be created if this default behavior is insufficient.

### Alarm Configuration

- Get the active alarm list

The procedure in *Check Alarm Status* provides further details on how to perform this operation.

The active alarm list is typically checked in the following two situations:

- When a management system (or its user) detects or suspects some alarm loss, it can “resynchronize” with the ME by getting its alarm list. Such operation is preferably done over NETCONF but can also be achieved over the ECLI and SNMP.
- When a user needs to check the ME alarm state through a man–machine interface, the user can access the active alarm list over the ECLI.

- Configure alarm severity

The user can experience that an alarm default severity does not properly reflect the actual severity of the problem. For example, an alarm with severity Major is instead to have severity Minor. The user can configure an alternative severity in the Alarm Type to supersede the default severity. The procedure in *Change Alarm Type Severity* provides further details on how to perform this operation.

- Change the heartbeat

The default heartbeat interval for the push mechanism is 60 seconds. The user can change the value to zero to disable the push heartbeat mechanism during maintenance operations. The user can change the heartbeat value to a smaller or higher value according to the organization monitoring policy. The procedure in *Change Heartbeat Interval* provides further details on how to perform this operation.

## 2.2 Ericsson Definition of 3GPP Perceived Severity Values

The definition of Perceived Severity in alarms is described in Table 2.

*Table 2 Perceived Severity Values*

<b>Severity Level</b>	<b>Description</b>
Cleared (1)	Used to clear a previously reported alarm.
Indeterminate (2)	Not used.
Critical (3)	Indicates that a condition that affects service has occurred and an immediate corrective action is required. Such a severity can be reported, for example, when an MO becomes out of service and its capability must be restored. This severity requires an immediate action, even outside working hours.
Major (4)	Indicates that a condition that affects service has occurred and an urgent corrective action is required. Such a severity can be reported, for example, when a service degrades in the MO capacity and its full capability must be restored. This severity requires an immediate action within working hours.
Minor (5)	Indicates that a fault condition that does not affect service has occurred. A corrective action is required to prevent a more serious fault such as a service-affecting fault. Such a severity can be reported when the detected alarm condition does not currently degrade the MO capacity. This severity requires an action at a suitable time, or at least that a close observation of the situation continues.
Warning (6)	Indicates that a potential or impending fault affects service, before any significant effects have appeared. Corrective action is based on a scheduled maintenance basis.

## 3 Managed Object Model

The FM managed area is represented in the *Managed Object Model (MOM)* in the following two parts:

- SNMP
- FM alarm

For general information about the MOM, MOC, MOs, cardinality, and related concepts, refer to *Managed Object Model User Guide*.

### 3.1 Managed Object Model — SNMP

The SNMP part is represented as follows:

```
ManagedElement
+-SystemFunctions
+-SysM
+-Snm
+-SnmTargetV1
+-SnmTargetV2C
+-SnmTargetV3
+-SnmViewV1
+-SnmViewV2C
+-SnmViewV3
```

The SNMP MOCs are described in Table 3.

*Table 3 SNMP Managed Object Class Descriptions*

Managed Object Class	Description
<i>Snm</i>	The root of the SNMP model, handles the SNMP administrative state, SNMP operational state, and listen addresses for the SNMP agent.
<i>SnmTargetV1</i>	Contains the configuration for SNMP targets receiving notifications over the SNMPv1 protocol.
<i>SnmTargetV2C</i>	Contains the configuration for SNMP targets receiving notifications over the SNMPv2C protocol.
<i>SnmTargetV3</i>	Contains the configuration for SNMP targets receiving notifications over the SNMPv3 protocol.
<i>SnmViewV1</i>	Handles an SNMP view, which gives one or more SNMPv1 users access to SNMP MIBs.



Table 3 SNMP Managed Object Class Descriptions

Managed Object Class	Description
<i>SnmpViewV2C</i>	Handles an SNMP view, which gives one or more SNMPv2C users access to SNMP MIBs.
<i>SnmpViewV3</i>	Handles an SNMP view, which gives one or more SNMPv3 users access to SNMP MIBs.

## 3.2 Managed Object Model — Fault Management

The FM alarm part is represented as follows:

```
ManagedElement
+-SystemFunctions
  +-Fm
    +-FmAlarm
    +-FmAlarmModel
    +-FmAlarmType
```

The FM MOCs are described in Table 4.

Table 4 Fault Management Alarm Managed Object Class Descriptions

Managed Object Class	Description
<i>Fm</i>	The root of the FM model, describes basic alarm information and defines the heartbeat interval.
<i>FmAlarm</i>	Each <i>FmAlarm</i> instance represents an active alarm. For details, see Table 5.
<i>FmAlarmModel</i>	Container for grouping FM alarm types.
<i>FmAlarmType</i>	Defines all the values for a given alarm type. These values are static values used by alarms and alerts reported at runtime. Attribute <i>isStateful</i> defines whether the alarm type is applicable to alarms or alerts. For details, see Table 5.

The mapping of the alarm information concepts to the NBI is shown in Table 5. 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 5 Mapping of Alarm Information to NBI

Alarm Information	FmAlarmType	FmAlarm	ERICSSON ALARM MIB	
			Alarm	Alert
Major Type	majorType	majorType	eriAlarmActiveMajorType	eriAlarmAlertMajorType
Minor Type	minorType	minorType	eriAlarmActiveMinorType	eriAlarmAlertMinorType
Managed Object Class	moClasses	(1)	(1)	(1)
Managed Object Instance/Source	(2)	source	eriAlarmActiveManagedObject	eriAlarmAlertManagedObject
Specific Problem	specificProblem	specificProblem	eriAlarmActiveSpecificProblem	eriAlarmAlertSpecificProblem
Event Type	eventType	eventType	eriAlarmActiveEventType	eriAlarmAlertEventType
Probable Cause	probableCause	probableCause	eriAlarmActiveProbableCause	eriAlarmAlertProbableCause
Additional Text	additionalText	additionalText	eriAlarmNObjAdditionalText	eriAlarmNObjAdditionalText
			eriAlarmNObjMoreAdditionalText	eriAlarmNObjMoreAdditionalText
Perceived Severity	defaultSeverity	activeSeverity	Indicated by NOTIFICATION-TYPE: • eriAlarmWarning • eriAlarmMinor • eriAlarmMajor • eriAlarmCritical • eriAlarmCleared	Indicated by NOTIFICATION-TYPE: • eriAlarmWarnAlert • eriAlarmMinorAlert • eriAlarmMajorAlert • eriAlarmCriticalAlert
	configuredSeverity			
Event Time	(2)	lastEventTime	eriAlarmActiveEventTime	eriAlarmAlertEventTime
Sequence Number	(2)	sequenceNumber	eriAlarmActiveLastSequenceNo	eriAlarmAlertLastSequenceNo

(1) Not applicable (included in MO instance).

(2) Not applicable.

Take the alarm DNS, Server Failed to Start as an example:

Table 6 Mapping of Alarm Information to NBI

Alarm Information	FmAlarmType	FmAlarm	ERICSSON ALARM MIB	
			Alarm	
Major Type	majorType=193	majorType=193	eriAlarmActiveMajorType=193	
Minor Type	minorType=851971	minorType=851971	eriAlarmActiveMinorType=851971	
Managed Object Class	moClasses=IPWorksDns	(1)	(1)	
Managed Object Instance/Source	(2)	ManagedElement=<Node Name>,SystemFunctions=1,Fm=1,FmAlarmModel=ipworksDns,FmAlarmType=ipworksDnsServFatalError,HostName=<PL hostname>	eriAlarmActiveManagedObject= "ManagedElement=<Node Name>,HostName=<Hostname>,IpworksDns"	
Specific Problem	specificProblem="DNS, Server Failed to Start"	specificProblem= "DNS, Server Failed to Start"	eriAlarmActiveSpecificProblem= "DNS, Server Failed to Start"	
Event Type	eventType= PROCESSING ERRORALARM	eventType= PROCESSING ERRORALARM	eriAlarmActiveEventType= processingErrorAlarm (10)	



Table 6 Mapping of Alarm Information to NBI

Alarm Information	FmAlarmType	FmAlarm	ERICSSON ALARM MIB
			Alarm
Probable Cause	probableCause= 307	probableCause= 307	eriAlarmActiveProbableCause= 307
Additional Text	additionalText= "The Alarm is raised due to memory errors or license limitation. There can be more than one available reason at the same time."	additionalText= "The Alarm is raised due to memory errors or license limitation. There can be more than one available reason at the same time."	eriAlarmNObjAdditionalText= "The Alarm is raised due to memory errors or license limitation. There can be more than one available reason at the same time."
			eriAlarmNObjMoreAdditionalText=<Empty>
Perceived Severity	defaultSeverity= CRITICAL	activeSeverity= CRITICAL	NOTIFICATION-TYPE: eriAlarmCritical
	configuredSeverity=<Empty>		
Event Time	(2)	lastEventTime= "2014-08-25T10:58:12+02:00"	eriAlarmActiveEventTime = "2014-08-25T10:58:12+02:00"
Sequence Number	(2)	sequenceNumber=25	eriAlarmActiveLastSequenceNo=25

(1) Not applicable (included in MO instance).

(2) Not applicable.

The mapping of heartbeat information concepts to the NBI is shown in Table 7. Columns *Fm* and *ERICSSON ALARM MIB* show what information a management system must access to implement a heartbeat pull over NETCONF and SNMP, respectively.

Table 7 Mapping of Heartbeat Information to NBI

Heartbeat Information	Fm	ERICSSON ALARM MIB
Latest time stamp	lastChanged	eriAlarmActiveLastChanged
Latest sequence number	lastSequenceNo	eriAlarmActiveLastSequenceNo

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



## 4 Configuration Management

SNMP and alarm configuration is accessed using NETCONF or the ECLI to manipulate the MIB.

The following operations can be performed by the user and are described in Operating Instructions using the ECLI:

### SNMP Configuration

- *Configure SNMP Master Agent*
- *Create SNMPv1 Target*
- *Create SNMPv2C Target*
- *Create SNMPv3 Target*

**Note:** For initial configuration, only creating an SNMPv3 target is permitted.

- *Delete SNMP Target*
- *Disable SNMP Target*
- *Enable SNMP Target*
- *Create SNMP View*

### Alarm Configuration

- *Check Alarm Status*
- *Change Alarm Type Severity*
- *Change Heartbeat Interval*





## 5 IPWorks Application Fault Management





## 6 File Management

The Alarm Log and Alert Log files are exposed by File Management as two file groups named `AlarmLogs` and `AlertLogs`, respectively. For more information on file groups, refer to *File Management*.

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, see Table 8. 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
<code>&lt;FmLogRecord&gt;</code>	Log record start
<code>&lt;LogTimestamp&gt;</code>	
Time stamp tag	The time the record is logged Format: <code>&lt;YYYY-MM-DDThh:mm:ss&gt;Z</code>
<code>&lt;/LogTimestamp&gt;</code>	
<code>&lt;Alarm&gt;</code> or <code>&lt;Alert&gt;</code>	
Alarm or alert-specific information	Formatted as a semicolon-separated string containing the following tokens:  <ol style="list-style-type: none"> <li>1. Event Time</li> <li>2. Source</li> <li>3. Major Type</li> <li>4. Minor Type</li> <li>5. Specific Problem</li> <li>6. Probable Cause</li> <li>7. Severity</li> <li>8. Additional Text</li> <li>9. Sequence Number</li> <li>10. Event Type</li> </ol>
<code>&lt;/Alarm&gt;</code> or <code>&lt;/Alert&gt;</code>	
<code>&lt;/FmLogRecord&gt;</code>	Log record end

Examples of log records in an Alarm Log are shown in Example 1.



```
<FmLogRecord>
  <LogTimestamp>2014-03-06T11:20:15Z</LogTimestamp>
  <Alarm>
    2014-03-06T11:20:14Z; ManagedElement=<Node Name>,SystemFunctions=1,Brm=1,BrmBackupManager=>
    SYSTEM_DATA,BrmBackupScheduler=SYSTEM_DATA;193;327681;BRM, Scheduled Backup Failed;418;MAJOR;=>
    Scheduled Backup for SYSTEM_DATA failed with disk space error;33;OTHER
  </Alarm>
</FmLogRecord>

<FmLogRecord>
  <LogTimestamp>2014-03-06T11:20:25Z</LogTimestamp>
  <Alarm>
    2014-03-06T11:20:25Z; ManagedElement=<Node Name>,SystemFunctions=1,Brm=1,BrmBackupManager=>
    SYSTEM_DATA,BrmBackupScheduler=SYSTEM_DATA;193;327681;BRM, Scheduled Backup Failed;418;CLEARED;=>
    Scheduled Backup for SYSTEM_DATA failed with disk space error;34;OTHER
  </Alarm>
</FmLogRecord>
```

### *Example 1 Log Records in Alarm Log*





## 7 Fault Management Function

Figure 3 illustrates the IPWorks Application Fault Management (FM) framework.

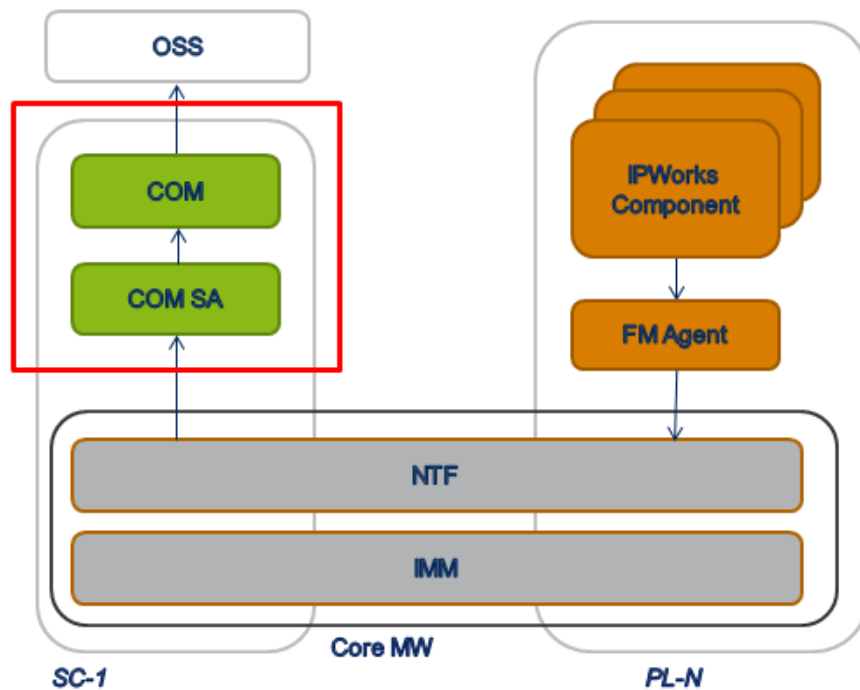


Figure 3 Overview of FM Framework

The Fault Management is featured as following:

- IPWorks application sends alarms using the Notification service (NTF) provided by Core MW.
- FM agent is used for all IPWorks components. It has unified communication mechanism and unified data structure. All alarms generated by IPWorks application are sent to FM agent firstly. Then, FM agent sends alarms to NTP service by NTF service interface.
- COM sends unified alarm notifications to OSS according to the configuration in ECLI.
- Users can list the active alarm in ECLI.

For details about IPWorks alarms, refer to *IPWorks Alarm List*.





## Reference List

- [1] *ERICSSON-ALARM-PC-MIB*
- [2] *AlarmListRebuilt*
- [3] *HeartBeat*
- [4] *ERICSSON-ALARM-MIB*
- [5] *Managed Object Model (MOM)*
- [6] *Managed Object Model User Guide*
- [7] *File Management*
- [8] *Configure SNMP Master Agent*
- [9] *Create SNMPv1 Target*
- [10] *Create SNMPv2C Target*
- [11] *Create SNMPv3 Target*
- [12] *Delete SNMP Target*
- [13] *Disable SNMP Target*
- [14] *Enable SNMP Target*
- [15] *Create SNMP View*
- [16] *Check Alarm Status*
- [17] *Change Heartbeat Interval*
- [18] *IPWorks Alarm List*