ERICSSON

ENM Monitoring System Administrator Guide

Operating Instructions

1/1543-AOM 901 151-2 Uen CG

Contents

1 ENM Monitoring System Administrator Guide

> This document describes the system administration tasks for the ENM Monitoring applications.

Ericsson Network Manager can be deployed to physical or cloud environments. The user needs to access to ENM Management Portal (EMP) if ENM is deployed on cloud environment.

Target Group

System Administrators

2 Connect to a Service

2.1 Connect to a Virtual Machine on a Physical ENM Deployment

Prerequisites

A command window is open and you have `superuser` privileges.

Steps

1. Log on to the ENM MS as **`litp-admin`** user and switch to the **`root`** user.
2. List the contents of the host file to view all connected VMs within the deployment.

```
[root@ms-1 ~]# cat /etc/hosts
...
192.168.99.20 svc-1-pmserv # Created by LITP. Please do not edit
192.168.99.26 svc-1-netex # Created by LITP. Please do not edit
192.168.99.16 svc-1-ebc # Created by LITP. Please do not edit
192.168.99.36 svc-1-mspm # Created by LITP. Please do not edit
192.168.99.28 svc-1-uiserv # Created by LITP. Please do not edit
192.168.99.14 svc-1-supervc # Created by LITP. Please do not edit
..
192.168.99.32 svc-1-mscm # Created by LITP. Please do not edit
..
192.168.99.50 svc-1-jms # Created by LITP. Please do not edit
..
192.168.99.3 logstash # Created by LITP. Please do not edit
..
192.168.99.2 httpd # Created by LITP. Please do not edit
192.168.99.40 sso # Created by LITP. Please do not edit
..
192.168.99.12 svc-1-medrout # Created by LITP. Please do not edit
192.168.99.22 svc-1-cmserv # Created by LITP. Please do not edit
192.168.99.52 svc-1-sec # Created by LITP. Please do not edit
192.168.99.8 openidm # Created by LITP. Please do not edit
```

   The aliases for the parallel VMs take the form of `<SVC host>-<service>`.

   For example: `svc-1-cmserv, svc-2-cmserv`.

   The active-passive VMs take the form of `<service>`.

   For example: `httpd, sso, openidm`.

3. To access the VM, copy the private key of the cloud-user from its secure location to the `MS` or `SVC` node.

```
[root@ms-1 ~]# /root/.ssh/vm_private_key
```

Refer to *VM Security Tasks* in the *ENM System Administrator Guide* to learn more about the `vm_private_key`.

4. Connect by `SSH` to the VM you want.

   To access the `VM`, use the `cloud-user` user ID and include the path to the `VM` private key. For example:

```
[root@ms-1 ~]# ssh -i /root/.ssh/vm_private_key cloud-user@svc-1-cmserv
Last login: Thu Feb 26 10:14:43 2015 from 192.110.0.59
[cloud-user@svc-1-cmserv ~]# sudo su - root
[root@svc-1-cmserv ~]#
```

2.1.1 Connect to each ENM Physical Node

Prerequisites

— The root password was changed during the installation process and must be known by the system administrator. This must be repeated on all newly deployed ENM nodes.

— A command window is open.

Steps

1. Log on to each physical node from the MS

```
[root@ms-1 ~]$ ssh litp-admin@<node_hostname>
litp-admin@<node_hostname>'s password:
Last login: Mon Feb 23 11:25:13 2015 from ms-1
[litp-admin@<node_hostname> ~]$ su - root
Password:
[root@<node_hostname> ~]#
```

Note: Once connected, after the initial deployment, the passwords for both the `litp-admin` and `root` users must be changed.

2.2 Connect to a Virtual Machine on an ENM on Cloud Deployment

Prerequisites

— A command window is open and you have `superuser` privileges.

— You have access to the private key file for authentication, contact your OpenStack administrator

Steps

1. List the virtual machine aliases from the consul service:

   Using the private key for authentication, copy the key to the EMP server. Log on to EMP server and list the consul members to view all connected VMs within the deployment:

```
> scp -i <cloud-user private key> <cloud-user private key> cloud-user@<EMP I →
P Address>:/var/tmp/vm_private_key
> ssh -i <cloud-user private key> cloud-user@<EMP IP Address>
[cloud-user@ostk003-emp-0 ~]$ chmod 700 /var/tmp/vm_private_key
[cloud-user@ostk003-emp-0 ~]$ sudo su -
[root@ostk003-emp-0 ~]# consul members
Node                      Address          Status  Type    Build  Protocol  →
 DC
haproxy                   10.3.2.31:8301   alive   client  0.8.1  2         →
 dc1
opendj-1                  10.3.2.83:8301   alive   client  0.8.1  2         →
 dc1
opendj-2                  10.3.2.84:8301   alive   client  0.8.1  2         →
 dc1
openidm                   10.3.2.85:8301   alive   client  0.8.1  2         →
 dc1
ostk003-accesscontrol-0   10.3.1.251:8301  alive   client  0.8.1  2         →
 dc1
ostk003-accesscontrol-1   10.3.1.252:8301  alive   client  0.8.1  2         →
 dc1
ostk003-elasticsearch-0   10.3.2.15:8301   alive   client  0.8.1  2         →
 dc1
...
ostk003-neo4j-2           10.3.2.77:8301   alive   client  0.8.1  2         →
 dc1
ostk003-nfscommon-0       10.3.0.81:8301   alive   client  0.8.1  2         →
 dc1
ostk003-nfsnrbk-0         10.3.0.83:8301   alive   client  0.8.1  2         →
 dc1
ostk003-nfspm-0           10.3.0.85:8301   alive   client  0.8.1  2         →
 dc1
ostk003-nfspm-1           10.3.0.82:8301   alive   client  0.8.1  2         →
 dc1
...
ostk003-secserv-1         10.3.2.98:8301   alive   client  0.8.1  2         →
 dc1
ostk003-serviceregistry-0 10.3.2.100:8301  alive   server  0.8.1  2         →
 dc1
ostk003-serviceregistry-1 10.3.2.101:8301  alive   server  0.8.1  2         →
 dc1
ostk003-serviceregistry-2 10.3.2.102:8301  alive   server  0.8.1  2         →
 dc1
ostk003-uiserv-0          10.3.2.116:8301  alive   client  0.8.1  2         →
 dc1
ostk003-uiserv-1          10.3.2.117:8301  alive   client  0.8.1  2         →
 dc1
ostk003-vnflaf-services   10.3.1.249:8301  alive   client  0.8.1  2         →
 dc1
...
svc-2-httpd               10.3.2.35:8301   alive   client  0.8.1  2         →
 dc1
svc-2-sps                 10.3.2.111:8301  alive   client  0.8.1  2         →
 dc1
svc-2-sso                 10.3.2.113:8301  alive   client  0.8.1  2         →
 dc1
```

2. SSH to the VM you want.

   To access the VM, use the cloud-user user ID and include the path to the VM private key. The VM can be accessed using either the node identifier or it's IP address. For example:

```
[cloud-user@ostk003-emp-0 ~]$ ssh -i /var/tmp/vm_private_key cloud-user@10.3 →
.2.31
The authenticity of host 'haproxy (10.3.2.31)' can't be established.
RSA key fingerprint is b9:4f:ca:4f:bc:55:00:de:a8:77:e5:08:56:7c:db:98.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'haproxy,10.3.2.31' (RSA) to the list of known ho →
sts.
[cloud-user@haproxy ~]$
```

2.3 View Log Files and Dump Locations on a Virtual Machine

The following are details of log files available within each service in ENM.

Logs

All logs are configured to be forwarded to the Central Log Service. As such they are visible in Log Viewer using the ENM Launcher.

JBOSS Logs

All JBOSS logs are stored locally in `/ericsson/3pp/jboss/standalone/log`

3PP & System Logs

As standard, most 3PP and system logs are available locally in `/var/log`

Dumps

All application memory and core dump files are located in `/ericsson/enm/dumps`

3 Restarting a Service

3.1 Restart a Service on a Physical ENM Deployment

Prerequisites

— Root access to MS.

Steps

1. Establish the service instances installed on the ENM deployment using `grep` for a particular service instance:

```
[root@<MS> ~]# /opt/ericsson/enminst/bin/vcs.bsh --groups | grep <service_name>
```

Example

```
[root@ieatlms4352 ~]#  /opt/ericsson/enminst/bin/vcs.bsh --groups | grep msp
m

svc_cluster Grp_CS_svc_cluster_mspm ieatrcxb2539-1 parallel vm ONLINE OK -
svc_cluster Grp_CS_svc_cluster_mspm ieatrcxb4373    parallel vm ONLINE OK -
svc_cluster Grp_CS_svc_cluster_mspm ieatrcxb4374    parallel vm ONLINE OK -
```

2. Restart the VCS service group:

```
/opt/ericsson/enminst/bin/vcs.bsh --restart -g <service_group> -s <system>
```

Note: The `-s` command restarts only one service at a time. To restart multiple services, repeat the command and modify the system name.

It is not recommended (unless specifically instructed) to restart more than one instance of a service at the same time. Restarting more than one instance of a service at the same time impacts the service availability and also results in some application specific consequences.

Example

```
/opt/ericsson/enminst/bin/vcs.bsh --restart -g Grp_CS_svc_cluster_mspm -s ieatrcxb4373

[root@ms-1 bin]# bash vcs.bsh --restart -g Grp_CS_svc_cluster_mspm -s ieatrcxb4373
2020-07-23 12:02:04.481 INFO hagrp_offline : Offlining 1 group(s)
2020-07-23 12:02:04.515 INFO hagrp_offline : Offlining Grp_CS_svc_cluster_mspm on ieatrcxb4373
2020-07-23 12:02:04.807 INFO wait_vcs_state : Waiting for Grp_CS_svc_cluster_mspm to go OFFLINE on ieatrcxb4373 (timeout=1800)
2020-07-23 12:05:43.185 INFO wait_vcs_state : Group Grp_CS_svc_cluster_mspm now OFFLINE on ieatrcxb4373 (3m:39s)
```

```
2020-07-23 12:05:43.817 INFO hagrp_online : Onlining 1 group(s)
2020-07-23 12:05:43.822 INFO online_services : Onlining Grp_CS_svc_cluster_m →
spm on ieatrcxb4373
2020-07-23 12:05:44.057 INFO wait_vcs_state : Waiting for Grp_CS_svc_cluster →
_mspm to go ONLINE on ieatrcxb4373 (timeout=4500)
2020-07-23 12:09:03.400 INFO wait_vcs_state : Group Grp_CS_svc_cluster_mspm  →
now ONLINE on ieatrcxb4373 (3m:19s)
[root@ms-1 bin]#
```

3. Verify if the service instance is `ONLINE`:

```
/opt/ericsson/enminst/bin/vcs.bsh --groups | grep mspm
```

Example

```
[root@ieatlms4352 ~]#  /opt/ericsson/enminst/bin/vcs.bsh --groups | grep msp →
m

svc_cluster Grp_CS_svc_cluster_mspm ieatrcxb2539-1 parallel vm ONLINE OK -
svc_cluster Grp_CS_svc_cluster_mspm ieatrcxb4373    parallel vm ONLINE OK -
svc_cluster Grp_CS_svc_cluster_mspm ieatrcxb4374    parallel vm ONLINE OK -
```

4. After the service restarted in *Step 2* is `ONLINE`, you can repeat *Step 2* and *Step 3* to restart further instances of the service as per your requirement.

3.2 Restart a Service on an ENM on Cloud Deployment

Prerequisites

— User connected to EMP server.

Steps

1. Establish the service instances installed on the vENM deployment using grep for a particular service instance.

```
#consul members | grep <service name>
```

Example

```
#consul members | grep mscm
```

2. Connect to the VM of the service group by following *section 3.2* and trigger a healthcheck failure of the VM by killing consul.

```
#pkill consul
```

3. Verify if the service instance is ONLINE.

4. After the restarted service is ONLINE, repeat the preceding two steps to restart further instances of the service as per your requirement.

4 Configuring PIB Parameters

> To configure a Platform Integration Bridge (PIB) parameter, it is necessary to determine what environment you are working on and follow the task relevant to your environment.

4.1 Configuring PIB Parameters on a Physical ENM Deployment

To access the PIB (Platform Integration Bridge) script to update parameters for ENM applications, it is necessary to determine what environment you are working on and follow the task relevant to your environment. This task outlines the steps to read and configure PIB parameters on a physical ENM Deployment.

Prerequisites

— A command window is open and you have super user privileges.

— You are connected to the ENM MS as per the Connect to a Virtual Machine on a Physical ENM Deployment on page 2.

Steps

1. Find the hostname for the service instance:

```
grep <service_name> /etc/hosts
```

2. Choose one of the returned hostnames for the next steps.
3. Navigate to the following directory:

```
[root @ms-1 ~]# cd /ericsson/pib-scripts/etc/
```

4. Check a configuration parameter on sample VM:

```
./config.py read --app_server_address=<service VM hostname>:8080 --service_i →
dentifier=<service_identifier_name> --name=<parameter_name>
```

Note: `--service_identifier=<service_identifier_name>` is optional for this command.

Example

To check value of the `SMRS_ERBS_NoOf_BACKUP_FILES` parameter:

```
./config.py read --app_server_address=svc-1-smrsserv:8080 --name=SMRS_ERBS_N
oOf_BACKUP_FILES
```

5. Update a configuration parameter on a deployed VM:

```
./config.py update --app_server_address=<service VM hostname>:8080 --service
_identifier=<service_identifier_name> --name=<parameter_name> --value=<new_v
alue>
```

Note: `--service_identifier=<service_identifier_name>` is optional for this command.

Example

To update the `SMRS_ERBS_NoOf_BACKUP_FILES` value to 4:

```
./config.py update --app_server_address=svc-1-smrsserv:8080 --name=SMRS_ERBS
_NoOf_BACKUP_FILES --value=4
```

Results

You have updated an application parameter using the PIB script.

4.2 Configuring PIB Parameters on ENM on Cloud Deployment

To access the PIB (Platform Integration Bridge) script to update parameters for ENM applications, it is necessary to determine what environment you are working on and follow the task relevant to your environment. This task outlines the steps to read and configure PIB parameters on an ENM on Cloud Deployment.

Note: ENM concepts are explained in the *ENM Product Description*.

Prerequisites

— A command window is open and you have super user privileges.

— You are connected to an EMP VM using Connect to a Virtual Machine on an ENM on Cloud Deployment on page 3.

Steps

1. As cloud-user change to root:

```
[cloud-user@emp ~]$ sudo su -
[root@emp ~]#
```

2. Find the hostname for the service instance:

```
consul members|grep <service_name>
```

3. Choose one of the returned hostnames for the next steps.
4. Change directory to where the config.py script is located:

```
[root@emp ~]# cd /ericsson/pib-scripts/etc/
[root@gat-emp-0 etc]#
```

5. Read the current parameter value:

```
./config.py read --app_server_address=<service VM hostname>:8080 --service_i →
dentifier=<service_identifier_name> --name=<parameter_name>
```

6. Set the parameter to the required value:

```
./config.py update --app_server_address=<service VM hostname>:8080 --service →
_identifier=<service_identifier_name> --name=<parameter_name> --value=<new_v →
alue>
```

Results

You have updated an application parameter using the PIB script.

5 Fault Management Administration Tasks

This section contains the routine operation and maintenance tasks related to the administration of Fault Management applications.

5.1 Configure Alarm Severity Colors

This section describes how to configure PIB parameters for alarm severity color code in Alarm Monitor application.

The alarm severity colors, that are used in Alarm Monitor application, can be configured as per user needs. The same value (same color) can be set to multiple severities at the same time.

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

Note: After updating the configurable PIB values, FM users must refresh Alarm Monitor application to reflect the changes.

5.1.1 Alarm Severity Color Parameters

Table 1 Parameters Description

Parameter Name	Default Value	Parameter Description	Value Range
criticalColorCode	2	It represents the color code for severity CRITICAL	1 to 8
majorColorCode	3	It represents the color code for severity MAJOR	1 to 8
minorColorCode	4	It represents the color code for severity MINOR	1 to 8
clearedColorCode	5	It represents the color code for severity CLEARED	1 to 8
warningColorCode	7	It represents the color code for severity WARNING	1 to 8
indeterminateColorCode	8	It represents the color code for severity INDETERMINATE	1 to 8

Colors

Index	Image
1	Purple
2	Red
3	Orange
4	Yellow
5	Green
6	Pale Blue
7	Dark Blue
8	Grey

5.2 Configure the Alarm History

By default, historical alarms are purged automatically based on two conditions:

Size based The index size of historical alarms in SOLR database exceeds `SOLR_INDEX_SIZE_LIMIT` of the configured disk space.

Time based The historical alarms in SOLR are older than the `PURGE_TRIGGER_TIME`.

In either of these cases, calculated number of days of data is purged from the database.

For any other reason, historical alarms can be purged manually using the following script:

Manual Cleanup When the user wants to delete manually historical alarms from SOLR, the script called `purgeSolrData.sh` can be used.

To use `purgeSolrData.sh`, the user must have root access to the ENM server.

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

5.2.1 Alarm History Management Configuration Parameters

Parameter Name	Default Value	Parameter Description	Value Range	Units
MAX_SOLR_INDEX_SIZE	300	Maximum allowed SOLR database size for historical alarms.	100–300	GBytes
SOLR_INDEX_SIZE_LIMIT	70	Threshold of the SOLR database size (MAX_SOLR_INDEX_SIZE) after which the automatic housekeeping is executed at regularly intervals.	20–95	Percentage
SIZE_BASED_PURGE_TRIGGERING_TIME	3	Number of oldest days historical alarms that are deleted when SOLR reaches the SOLR_INDEX_SIZE_LIMIT value.	2–5	Days
PURGE_TRIGGER_TIME	90	Maximum number of days (excluding the current one) historical alarms are kept into the SOLR database. Historical alarms older than PURGE_TRIGGER_TIME days are deleted.	90–365	Days
PURGE_FREQUENCY	24	Time interval used to check if historical alarms are older than PURGE_TRIGGER_TIME.	12–24	Hours
SIZE_BASED_PURGE_RETRY_ATTEMPTS	25	Number of attempts to get the SOLR connection while performing size-based purge(even when the SOLR is down).	10–25	Number

Note: For 5K, Small ENM, and Small ENM on VIO, MAX_SOLR_INDEX_SIZE value is 104. The value must be changed manually.

5.2.2 Clean Manually Historical Alarms

There are two ways for purging historical alarms from the database using the purgeSolrData.sh script: delete all records and delete records on time basis.

Prerequisites

User has access to the ENM server and solr instance.

Steps

1. Connect to `solr` instance.

   For more information, see Connect to a Service on page 2.

2. Delete the records using the following syntax:

Option	Description
Delete All Records	`/ericsson/solr/install/purgeSolrData.sh collection1`

Delete Records on Time Basis	`/ericsson/solr/install/purgeSolrData.sh collection1  <Latest Number of days of data to hold>`
	Example of command to retain the history data of last 20 days and remove the history data older than 20 days:
	`/ericsson/solr/install/purgeSolrData.sh  collection1 20`

Results

The historical alarms are removed from the database.

5.2.3 Configure the Parameters to Purge Historical Alarms on Size Basis

Historical alarms are purged automatically when the index size of SOLR database reaches to the `SOLR_INDEX_SIZE_LIMIT` threshold value.

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

1. The historical alarms purging is triggered if the index size of SOLR database exceeds the threshold. the threshold is, by default, 80% of SOLR disk space and can be configured using the `SOLR_INDEX_SIZE_LIMIT` parameter.
2. The records that are older than the `SIZE_BASED_PURGE_TRIGGERING_TIME` value are removed (the default is three days).
3. The historical alarms purging is retried until it is successful or for 25 times. This can be configured using the `SIZE_BASED_PURGE_RETRY_ATTEMPTS` parameter.
4. If the size of the database still exceeds the threshold limit even after removal of oldest data, Steps 2 and 3 are triggered again.
5. This continues until the database size becomes less than the threshold limit.

Configure the Alarm History on page 12

5.2.4 Configure the Parameters to Purge Historical Alarms on Time Basis

> This section describes the procedure to change the automatic cleanup of the alarm history in terms of frequency and time. The alarm history is periodically purged by deleting historical alarms that are older than a defined number of days.

The alarm history is cleaned up regularly based on the values of the `PURGE_TRIGGER_TIME` and `PURGE_FREQUENCY` parameters.

For example, with the default `PURGE_TRIGGER_TIME` of 90 days and `PURGE_FREQUENCY` of 24 hours, the purging operation starts on day 91.

After 91 days, the data associated with day one is removed from the history.

At any time, the data store contains alarm history data of `(PURGE_TRIGGER_TIME + PURGE_FREQUENCY)` days.

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

5.3 Configure Housekeeping of Historical Alarms from Geo Replication Location

This section describes the configurable parameters for housekeeping of historical alarm data from Geo Replication location.

Historical alarms data needed for geographical replication is copied into `historyAlarmsDirectoryForGeoReplication` directory along with historical alarm database when `exportHistoryAlarmsForGeoReplication` parameter is enabled.

After every `purgeFrequencyForHistoryAlarmsForGeoReplication` interval, the purge mechanism is triggered if the size of the directory reaches the `thresholdForHistoryAlarmsForGeoReplication`.

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

5.3.1 Housekeeping Parameters

Parameter Name	Default Value	Parameter Description	Value Range
`exportHistoryAlarmsForGeoReplication`	false	This parameter defines if history alarms are exported to Geo Replication location or not.	NA

Parameter Name	Default Value	Parameter Description	Value Range
historyAlarmsDirectoryForGeoReplication	/ericsson/georeplication/fm/data	This parameter defines the directory where historical alarms are exported.	NA
maxSizeForHistoryAlarmsForGeoReplication	6.25	This parameter defines the maximum allowed space (GB) for historical alarms to be exported to Geo Replication location.	1–6.25
thresholdForHistoryAlarmsForGeoReplication	90	This parameter defines the threshold (percentage) of the directory after which housekeeping performed automatically.	70–100
purgeFrequencyForHistoryAlarmsForGeoReplication	60	This parameter defines the frequency (Minutes) at which automatic purge mechanism is triggered, if the threshold is crossed.	15–1440

5.4 Configure the Connectivity Information SNMP Parameters

> SNMP connectivity information used to support Fault Management of SNMP-based Network Elements.

For details on how to view and modify PIB parameters, seeConfiguring PIB Parameters on page 8.

5.4.1 SNMP Connectivity Attributes

Attribute Name	Attribute Description	Type	Mandatory	Default Values	Notes
snmpAgentPort	The port on which the SNMP agent (for example the node) listens for SNMP requests.	Integer	Yes	161 Set to: — 161 for Baseband RadioNodes — 25161 for SGSN-MME	
snmpVersion	The SNMP protocol version.	Enum: — SNMP_V1 — SNMP_V2C — SNMP_V3	Yes	SNMP_V2C Set to: — SNMP_V3 for CISCO-ASR9000, CISCO-ASR900, JUNIPER-MX, JUNIPER-vMX, JUNIPER-SRX, JUNIPER-vSRX, JUNIPER-PTX	Baseband RadioNodes and SGSN-MME nodes support SNMPv2C or SNMPv3.

Attribute Name	Attribute Description	Type	Mandatory	Default Values	Notes
snmpReadCommunity	The SNMPv1 and SNMP_V2c read community.	String		enm-public Set to: — public for CISCO-ASR9000, CISCO-ASR900, JUNIPER-MX,JUNIPER-vMX, JUNIPER-SRX, JUNIPER-vSRX, JUNIPER-PTX,SSR, vBNG, Router8800	Mandatory for SNMPv1 or SNMPv2C.
snmpWriteCommunity	The SNMPv1 and SNMP_V2c write community.	String		enm-public Set to: — public for CISCO-ASR9000, CISCO-ASR900, JUNIPER-MX, JUNIPER-vMX, JUNIPER-SRX, JUNIPER-vSRX, JUNIPER-PTX, SSR, vBNG, Router8800	Mandatory for SNMPv1 or SNMPv2C. For ECIM-based nodes (for example Baseband RadioNodes & SGSN-MME), the value must be identical to the snmpReadCommunity value.
snmpSecurityName	SNMP Security name for v3 protocol.	String		<null>	Mandatory for SNMPv3.
snmpSecurityLevel	The SNMP_V3 Security Level.	Enum: — NO_AUTH_NO_PRIV — AUTH_NO_PRIV — AUTH_PRIV		<null>	Mandatory for SNMPv3.

5.4.2 NetworkElementSecurity Parameters

Parameter Name	Parameter Description	Type	Default Values	Notes
snmpAuthKey	SNMPv3 Authentication Key	String	<null>	Mandatory for AUTH_NO_PRIV and AUTH_PRIV.
snmpAuthProtocol	SNMPv3 Authentication Protocol	Enum: — SHA1 — MD5 — NONE	NONE	Mandatory for AUTH_NO_PRIV and AUTH_PRIV.
snmpPrivKey	SNMPv3 Privacy (encryption) Key	String	<null>	Mandatory for AUTH_PRIV.
snmpPrivProtocol	SNMPv3 Authentication Protocol	Enum: — NONE — DES — AES128	NONE	Mandatory for AUTH_PRIV.

The security parameters are configured using the `secadm snmp` ENM CLI command.

5.5 Configure Alarm Overload Protection

> This section describes how to configure alarm overload protection to protect the system against alarm storms.

ENM have a self-protection function against alarm storms. An alarm event filter is implemented in order to secure that required alarm processing in ENM continuously works so that monitoring of important alarms is without interruption. An inbuilt default filtering rule is available in ENM.

When the rate of alarms received reaches the alarm threshold limit during the alarm rate check interval, an alert is raised by FM alarm processor containing the specific problem 'Alarm Overload detected'. The subsequent alarms coming from the network elements are discarded until the alarm rate is reduced by 70% of the alarm rate threshold count.

Network elements whose alarms are discarded are set to Alarm Suppressed state.

When the alarm count is decreased by 70% of the AlarmRateThreshold, then the 'alarm overload detected' alert is automatically acknowledged and processing of new alarms restarts.

For details on how to view and modify PIB parameters , see Configuring PIB Parameters on page 8.

5.5.1 Fault Management Alarm Overload Protection Parameters

Parameter Name	Default Value	Parameter Description	Value Range	Units
`alarmOverloadProtectionOn`	true	Indicates whether the alarm protection functionality is enabled.	true-false	N/A
`alarmOverloadProtectionThreshold`	`55200`	Indicates the maximum number of alarms allowed within the alarm rate check interval of 5 minutes.	`10000 – 200000`	Integer
`alarmOverloadProtectionLowerThreshold`	`70`	Indicates the percentage of alarm protection threshold for ceasing alarm protection on the whole network.	`50–90`	Integer

Table 2 Suggested alarmOverloadProtectionThreshold Values per Deployment Type

ENM Deployment Type	alarmOverloadProtectionThreshold Suggested Value
Extra Large ENM (ENM@60k)	55200
Large ENM (ENM@40k)	52500
Medium ENM (ENM@15k)	49700
Small ENM on Openstack Cloud	48600
Extra Small ENM on Openstack Cloud	48100

5.6 Enable and Disable Fault Management Alarm Rate Threshold Crossed Functionality

> When the number of alarms received on any network element reaches the alarm threshold limit during the alarm rate check interval, an internal alarm is raised by FM mediation containing the details 'Alarm Rate Threshold Crossed'.

The subsequent alarms coming from the network element are discarded until the alarm rate is reduced by 50% of the alarm rate threshold count.

Access to the management server.

When the number of alarms received on a particular network element reaches alarm threshold limit, FM mediation raises the 'Alarm Rate Threshold Crossed' alarm.

The network element goes to 'Suspended' state and new alarms are discarded.

When the alarm count is decreased by 50% of the AlarmRateThreshold, then a clear for 'Alarm Rate Threshold Crossed' alarm is generated and processing of new alarms starts.

For details on how to view and modify PIB parameters , see Configuring PIB Parameters on page 8.

5.6.1 Fault Management Alarm Rate Parameters

Parameter Name	Default Value	Parameter Description	Value Range	Units
ALARMRATE_FLOW_CONTROL	true	Indicates whether the alarm threshold functionality is enabled.	N/A	N/A
ALARMRATE_THRESHOLD	180	Indicates the maximum number of alarms allowed within the alarm rate check interval.	100–1000	Integer

Parameter Name	Default Value	Parameter Description	Value Range	Units
ALARMRATE_NORMAL_THRESHOLD	50	Indicates the percentage of alarm rate threshold for clearing the 'Alarm Rate Threshold Crossed' alarm and start processing alarms on the network element.	30–50	Integer
ALARMRATE_CHECK_INTERVAL	3	Indicates the amount of time, in minutes, to check the alarm count on each network element for raising alarm threshold crossed alarm on network element.	1–10	Minutes

5.7 Enable and Disable Fault Management Delayed Acknowledgment Functionality

> Delayed acknowledgment feature allows FM to acknowledge errors and ceased (cleared-unacknowledged) alarms after the configured period at regular intervals.

It is strongly recommended to enable the feature in case FM operators don't clear alarms manually and the number of cleared alarms grows continuously.

User has root access to the ENM server.

For details on how to view and modify PIB parameters using , see Configuring PIB Parameters on page 8.

5.7.1 Fault Management Delayed Acknowledgment Parameters

Parameter Name	Default Value	Parameter Description	Value Range	Units
FMA_DELAYED_ACK_OF_ALARMS_ON	TRUE	Indicates if the delayed acknowledgment functionality is enabled for alarms or not.	N/A	N/A
FMA_TIME_TO_DELAYED_ACK_ALARMS	24 hours	Cleared Alarms are automatically acknowledged after FMA_TIME_TO_DELAYED_ACK_ALARM hours (if the feature is enabled)	1–168	Hours
FMA_DELAYED_ACK_OF_EVENTS_ON	TRUE	Indicates if the delayed acknowledgment functionality is enabled for events (error messages) or not.	N/A	N/A
FMA_TIME_TO_DELAYED_ACK_EVENTS	24 hours	Events (error messages) are automatically acknowledged after FMA_TIME_TO_DELAYED_ACK_ALARM hours (if the feature is enabled)	1–168	Hours

Parameter Name	Default Value	Parameter Description	Value Range	Units
FMA_DELAYED_ACK_CHECK_INTERVAL	10 minutes	Time interval to check if either ceased alarms and events (error messages) have to be auto-acknowledge.	5–60	Minutes

5.8 Enable and Disable Fault Management Downward Acknowledgement Functionality

> Downward Acknowledgement functionality allows Ericsson Network Manager (ENM) to communicate acknowledge and unacknowledge operation information to network elements when performed on the ENM side. It provides flexibility to enable and disable this feature at runtime.

Access to the management server.

FM sends an acknowledgement request down to the network element if the parameter `downwardAck` is set to true.

For details on how to view and modify PIB parameters using , see Configuring PIB Parameters on page 8.

5.8.1 FM Alarm Downward Acknowledgement Parameters

Parameter Name	Default Value	Parameter Description	Value Range
downwardAck	false	Indicates if the downward acknowledgement functionality is enabled or not.	N/A

5.9 Enable and Disable Fault Management Oscillation-Correlation Functionality

> The Oscillation-Correlation feature is used to process the alarms and to avoid unnecessary alarms to appear in the Alarm Monitor application when they are generated and cleared on a Network Element and communicated to Ericsson Network Manager (ENM).

Access to the Management Server.

FM correlates the alarms and their relative clear actions when the same alarm is generated repeatedly on Network Element.

For details on how to view and modify PIB parameters , see Configuring PIB Parameters on page 8.

5.9.1 FM Oscillation-Correlation Parameters

Parameter Name	Default Value	Parameter Description	Value Range
`oscillationAlarmCorrelation`	true	Indicates if the oscillation-alarm-correlation functionality for alarms is enabled or not.	N/A

5.10 Enable and Disable Fault Management updateInsertTime Parameter

When a repeated alarm or event arrives at the Ericsson Network Manager (ENM) system, FM updates the insert time of the repeated alarm or event if the `updateInsertTime` parameter is set to true.

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

5.10.1 Update Insert Time Parameters

Parameter Name	Default value	Parameter Description	Value Range
`updateInsertTime`	false	Indicates if the `updateInsertTime` parameter is enabled or not.	N/A

5.11 Fault Management Heartbeat Timeout and Automatic Synchronization Configuration

> This section describes FM Heartbeat Timeout and Automatic Synchronization Configuration for Fault Management (FM).

For the updated values to be effective, toggle the alarm supervision on the network elements.

In some scenarios, the time taken to raise an alarm can be longer than the expected time due to network issues.

Prerequisites

You must have selected the FM_Administrator role while creating a user profile in the ENM system.

Steps

1. Modify heartbeat time-out, using the command alarm enable or alarm disable with the attribute `HeartbeatTimeout` in the ENM CLI.

For more information, see ENM CLI online help.

2. Change Automatic Synchronization, using the command alarm enable or alarm disable with the `automaticSynchronization` attribute in the ENM CLI.

   For more information, see ENM CLI online help.

5.12 Fault Management Auto-acknowlegment of Cleared Alarms

> This section describes how to update enableAutoAckOnManualClearedAlarms configuration parameter for auto-acknowledge route in Fault Management (FM).

When the feature is enabled, alarms that have been manually cleared via UI, CLI or NBI,FMX are acknowledged by any auto acknowledge rule if the criteria matches with alarms.

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

5.12.1 EnableAutoAckOnManualClearedAlarms Parameter

Table 3 enableAutoAckOnManualClearedAlarms

Parameter Name	Default Value	Parameter Description	Value Range
`enableAutoAckOnManualClearedAlarms`	false	It enables the auto acknowledge for alarms cleared from UI/CLI and NBI/FMX	N/A

5.13 Enable and Disable Fault Management Email Alarm Routing Functionality

If `ENABLE_OUTBOUND_EMAILS` configuration parameter is enabled, `Email Alarm Routing` feature allows the user to create the email alarm routes. All the alarm information is routed to designated email address, if any spontaneous alarm matches the created email alarm route criteria.

If `ENABLE_OUTBOUND_EMAILS` configuration parameter is disabled, `Email Alarm Route` cannot be created and emails are not sent to the designated email address.

Access to the management server.

FM sends the alarm information to the designated email address when the alarm with the matching criteria is received.

For details on how to view and modify PIB parameters using, see Configuring PIB Parameters on page 8.

5.13.1 Enable Outbound Emails Parameters

Parameter Name	Default Value	Parameter Description	Value Range
ENABLE_OUTBOUND_EMAILS	false	Indicates if the ENABLE_OUTBOUND_EMAILS for Email Alarm Routing is enabled or not.	N/A

5.14 Alarm Routing Save to File Configurable Parameters

> This section describes the Alarm Routing Save To File configurable parameters.

The alarm files are saved in the ENM server under the location `/ericsson/netlog/fm/alarmroute`.

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

5.14.1 Save to File Parameters

Parameter Name	Default Value	Parameter Description	Value Range
FILE_ROUTE_ROTATION_PERCENTAGE	75	The parameter represents the maximum percentage of route file size before the worker zip and rotate it.	50–90
MAX_NUMBER_OF_ZIPPED_ROUTE_FILES	2	The parameter value represents the maximum number of rotated zipped route files that have to be retained on the system.	1–4

5.15 Install Certificates to Verify the Digital Signature of an Email

To verify the digital signature of an email received from ENM FM, certificates need to be downloaded from ENM.

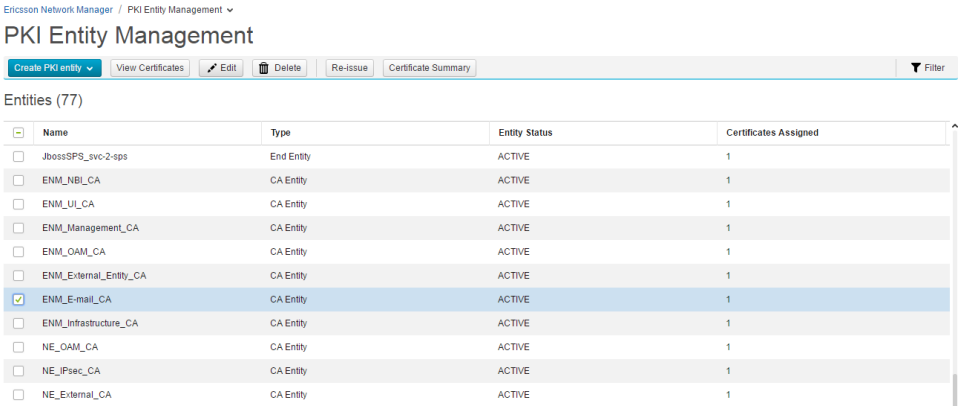
These certificates must be imported to mail client (for example: Microsoft Outlook) trusted publishers.

Steps

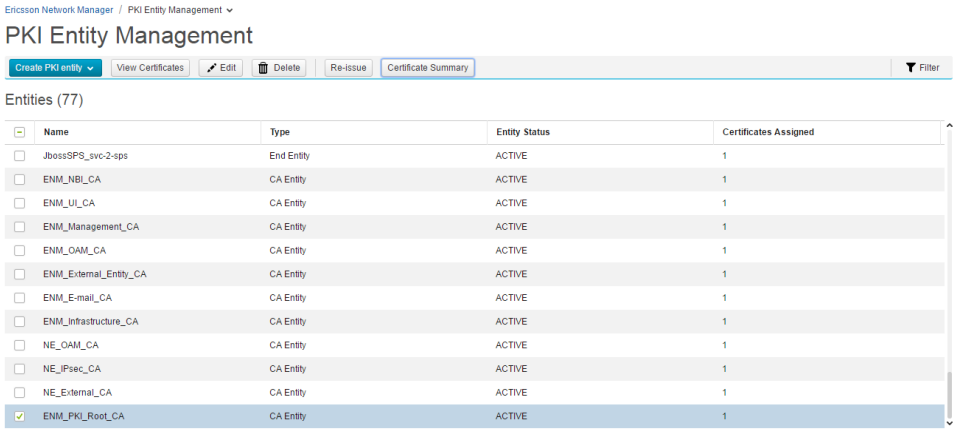
1. Import the certificates from ENM.
2. Install the imported certificates to Outlook trusted publishers.

5.15.1 Import the Certificates from ENM

1. Log on to the ENM Launcher application using valid credentials.
2. In ENM Launcher application, click **PKI Entity Management** link to go to PKI Entity Management application.
3. Select the check box besides ENM_E_Mail_CA and click **Certificate Summary** button.



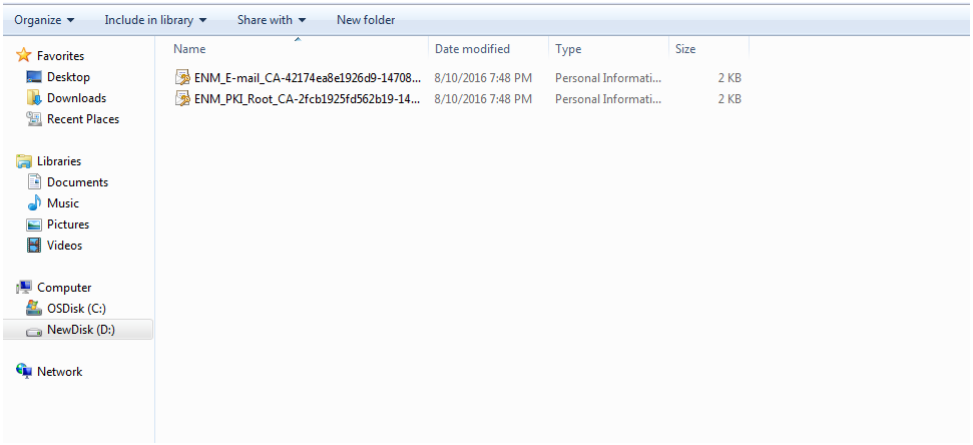
4. A fly out panel is opened as shown below. Save the certificate to local system by clicking **Download** button and select **PKCS 12** (p12) format.
5. Select the check box besides ENM_PKI_Root_CA and click **Certificate Summary** button.



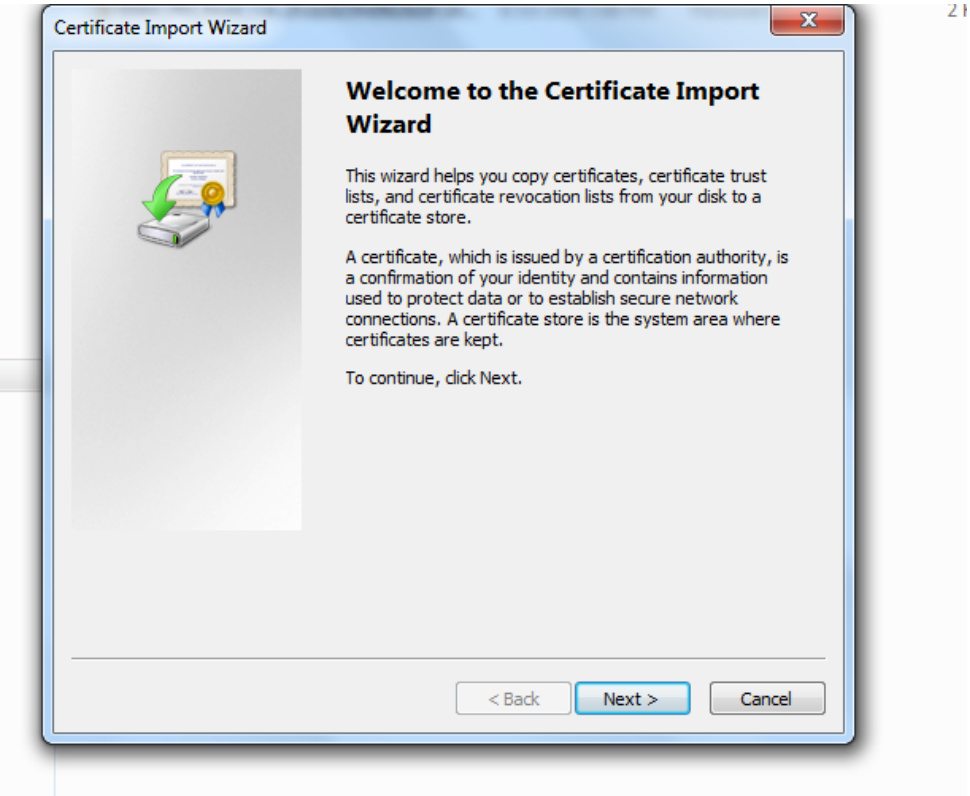
6. A fly out panel is opened as shown below. Save the certificate to local system by clicking **Download** button and select **PKCS 12** (p12) format.

5.15.2 Install the Certificates to Outlook

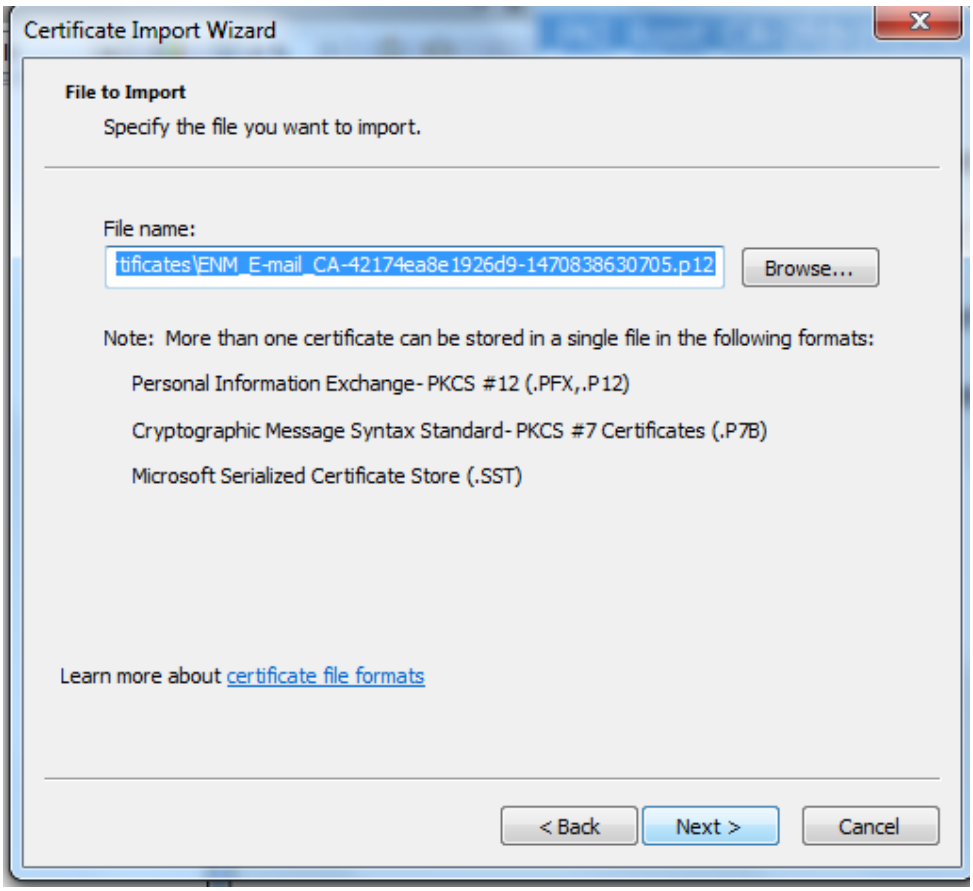
1. Double click on the **downloaded** certificate to start installation.



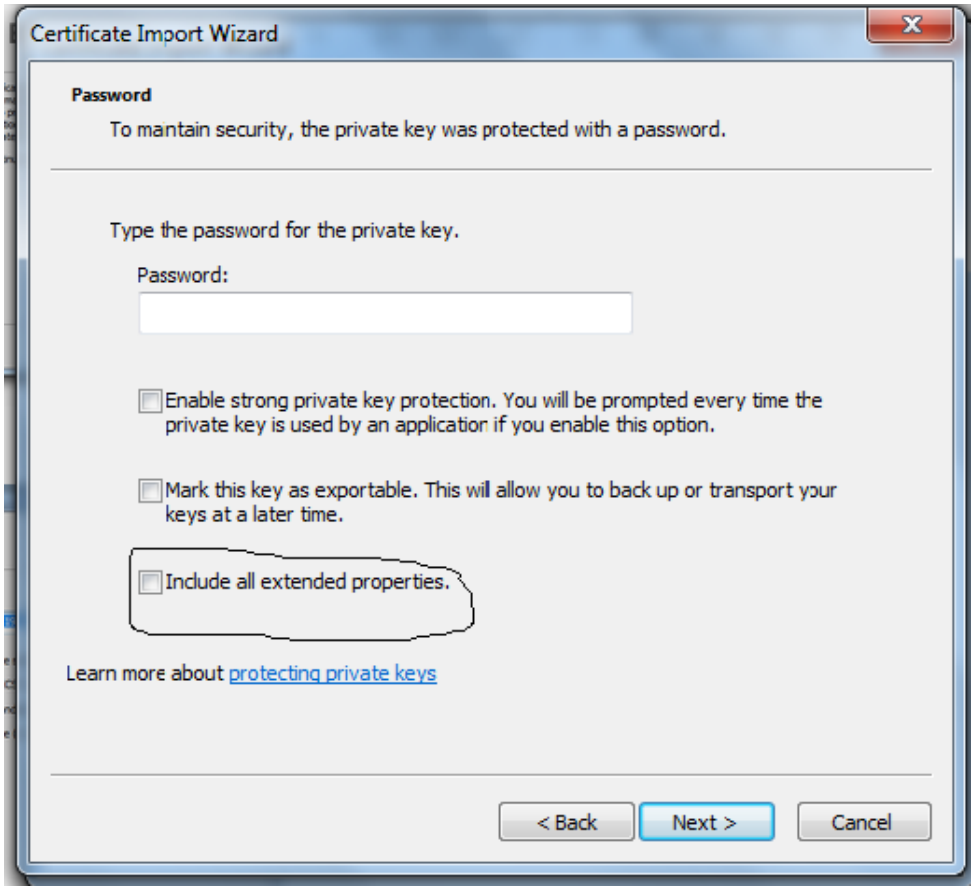
2. A window is opened.



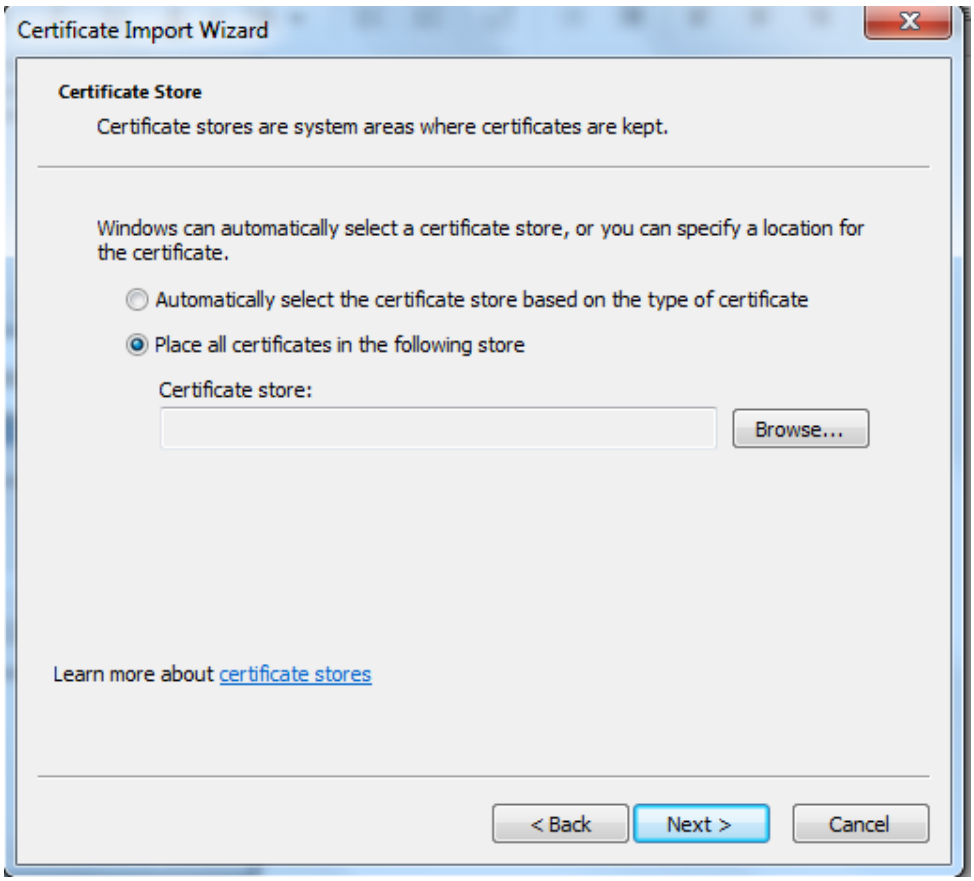
3. Click **Next** button.



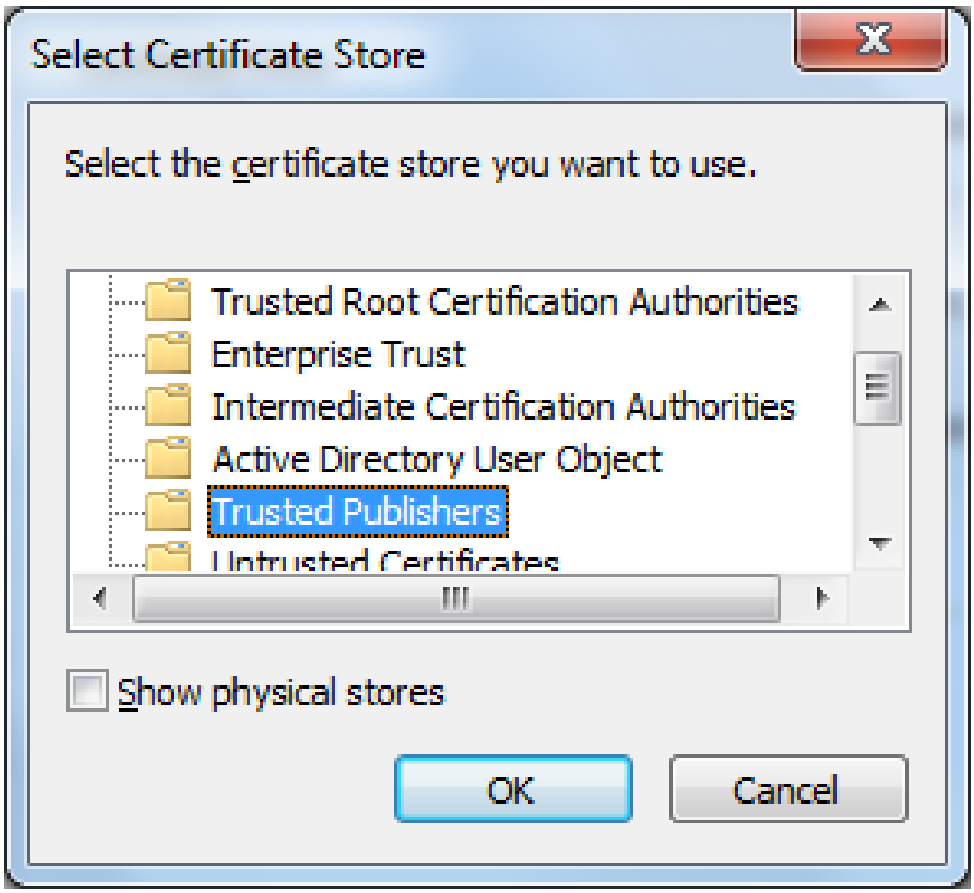
4. Do not provide any password and uncheck **Include all extended properties** checkbox and click **Next** button.



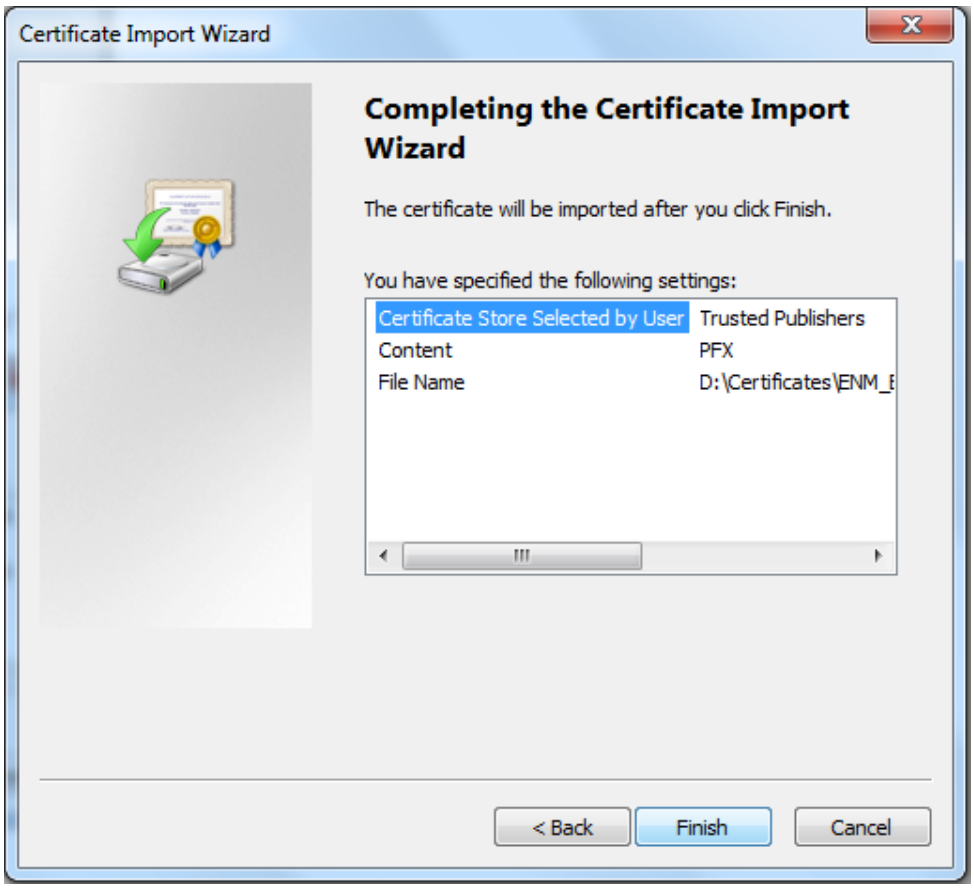
5. Select the **Place all certificates in the following store radio** button and click **Browse** button.



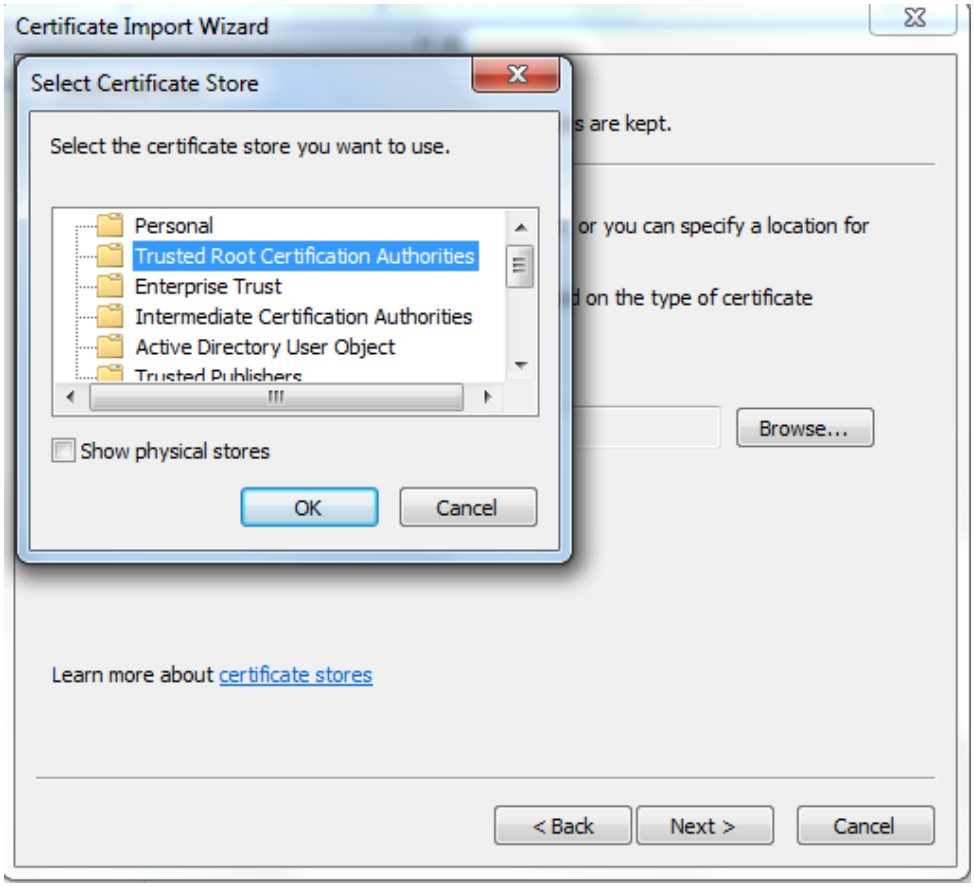
6. Select the folder **Trusted Publishers** and click **OK**.



7. A window is opened, click **Finish** button.



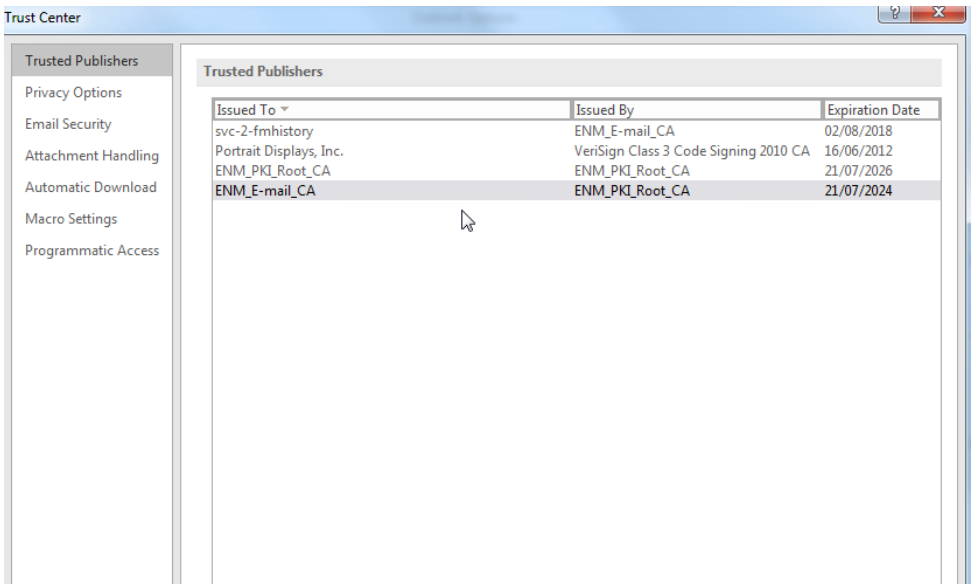
8. Repeat the steps 1,2,3,4 on the same certificate to install it in **Trusted Root Certification Authorities** folder; at step 5, after clicking **Browse** button, select the folder **Trusted Root Certification Authorities**.



Follow the same procedure provided in Install the Certificates to Outlook to install the second certificate.

5.15.3 Verify the Certificates Installation in Outlook

1. Click **File** option in outlook, a window is opened, click **Options** tab.
2. Click **Trust Centre** option.
3. Click **Trust Centre Settings**.
4. Click **Trusted Publishers** and check for ENM_PKI_Root_CA and ENM_E-mail_CA certificates in it.



5.16 Enable and Disable Event Communication To Fault Management From ENIQ

> This section describes how to enable and disable event communication from ENIQ to Fault Management (FM).

FM receives every ENIQ event and processes it if the configured parameter enableEniqs is true. FM discards the ENIQ event if the value of the parameter is false. By default the value of enableEniqs parameter is true.

Access to the management server (MS).

FM processes or discards ENIQ events based on the value of configured parameter enableEniqs.

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

5.16.1 Enable Eniqs Parameter

Parameter Name	Default Value	Parameter Description	Value Range
`enableEniqs`	true	This parameter decides whether FM needs to process ENIQ events or not. By default FM processes ENIQ events.	N/A

5.17 Update From Address Field for Email Route

> This section describes how to update EMAIL_FROM_ADDRESS configuration parameter for email route in Fault Management (FM).

FM sends the alarm information from the configured EMAIL_FROM_ADDRESS to designated email address when the alarm with matching criteria is generated on Network Element.

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

5.17.1 Email From Address Parameter

Parameter Name	Default Value	Parameter Description	Value Range
EMAIL_FROM_ADDRESS	enmfaultmanagement@ericsson.com	Indicates from address field of an email route	N/A

5.18 Update Alarm Routing Email Domains

> This section describes how to update ALARM_ROUTING_EMAIL_DOMAINS configuration parameter for email route in Fault Management (FM). This parameter supports multiple domains that are separated by a comma.

For each alarm that is generated on a Network Element and that matches the route criteria, an email related to the alarm is sent from the configured value of EMAIL_FROM_ADDRESS parameter to the destination email addresses, whose domain values match with the configured domain values in the ALARM_ROUTING_EMAIL_DOMAINS parameter.

When the ALARM_ROUTING_EMAIL_DOMAINS configuration parameter is updated, the previous parameter value is overridden.

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

5.18.1 Alarm Routing Email Domains Parameter

Parameter Name	Default Value	Parameter Description	Value Range
ALARM_ROUTING_EMAIL_DOMAINS	ericsson.com	Indicates the email domains configured for the email route.	N/A

To update the value of parameter without overriding the previously configured values, append the new values by using Comma (,) as a separetor.

For example:

```
--value=domain1.com,domain2.com,domain3.com
```

5.19 Threshold-Ack Functionality

FM raises an internal alarm with severity critical, when the number of open alarms in ENM crosses the specified configured parameter `alarmThresholdForNotification`. If the number of open alarms in ENM still continues and reaches the specified configured parameter `alarmThresholdForForceAck`, FM acknowledges all the cleared alarms after a configured period of time, at regular intervals if `FM_THRESHOLD_ACK_OF_ALARMS_ON` parameter is enabled. By default, Threshold-Ack functionality is enabled in the ENM system.

FM clears the critical alarm raised, if number of open alarms in ENM comes down to a value less than the value of `alarmThresholdForNotification` parameter.

— Threshold alarm with critical severity is generated if number of open alarms crosses the specified configured parameter `alarmThresholdForNotification`. If number of open alarms comes down to a value less than `alarmThresholdForNotification`, then FM clears the critical alarm generated for threshold crossed.

— If number of open alarms in ENM crosses the specified configured parameter `alarmThresholdForForceAck` and `FM_THRESHOLD_ACK_OF_ALARMS_ON` is true, then FM acknowledges all the cleared alarms in ENM.

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

5.19.1 Threshold-Ack Parameters

Table 4 Threshold-Ack Parameters for Small Integrated ENM Transport Only

Parameter Name	Default Value	Parameter Description	Value Range	Units	Comments
`FM_THRESHOLD_ACK_OF_ALARMS_ON`	true	Indicates whether the threshold ack of alarms is enabled or not.	N/A	N/A	-
`alarmThresholdForNotification`	`40000`	Indicates the threshold for alarm limit in the database to generate an internal alarm.	`1000` – `40000`	Integer	The setting of `alarmThresholdForNotification` value less than `1000`, updates the parameter value to the default value or the setting of `alarmThresholdForNotification` value greater than `40000`, updates the

Parameter Name	Default Value	Parameter Description	Value Range	Units	Comments
					parameter value to default value.
alarmThresholdForForceAck	60000	Indicates the threshold for the alarm limit in the database to start force acknowledging cleared alarms.	5000 – 60000	Integer	The setting of alarmThresholdForForceAck value less than 5000, updates the parameter value to default value or the setting of alarmThresholdForForceAck value greater than 60000, updates the parameter value to default value.
timerIntervalToCheckAlarms	5	Indicates the time in minutes to check if alarmLimitForNotification is reached.	5 – 10	Minutes	-

Table 5 Threshold-Ack Parameters for all Deployments Except Small Integrated ENM Transport

Parameter Name	Default Value	Parameter Description	Value Range	Units	Comments
FM_THRESHOLD_ACK_OF_ALARMS_ON	true	Indicates whether the threshold ack of alarms is enabled or not.	N/A	N/A	-
alarmThresholdForNotification	80000	Indicates the threshold for alarm limit in the database to generate an internal alarm.	1000 – 80000	Integer	the setting of alarmThresholdForNotification value less than 1000, updates the parameter value to default value or the setting of alarmThresholdForNotification value greater than 80000, updates the parameter value to default value.
alarmThresholdForForceAck	120000	Indicates the threshold for the alarm limit in the database to start force acknowledging cleared alarms.	5000 – 180000	Integer	The setting of alarmThresholdForForceAck value less than 5000, updates the parameter value to default value or the setting alarmThresholdForForceAck value greater than 180000, updates the parameter value to default value.
timerIntervalToCheckAlarms	5	Indicates the time in minutes to check if alarmLimitForNotification is reached.	5 – 10	Minutes	-

Table 6 Threshold-Ack Parameters for Extra Small ENM Deployments Only

Parameter Name	Default Value	Parameter Description	Value Range	Units	Comments
FM_THRESHOLD_ACK_OF_ALARMS_ON	true	Indicates whether the threshold ack of alarms is enabled or not.	N/A	N/A	-
alarmThresholdForNotification	7000	Indicates the threshold for alarm limit in the database to generate an internal alarm.	1000 – 7000	Integer	The setting of alarmThresholdForNotification value less than 1000, updates the parameter value to the default value.
alarmThresholdForForceAck	10000	Indicates the threshold for the alarm limit in the database to start force acknowledging cleared alarms.	5000 – 10000	Integer	The setting of alarmThresholdForForceAck value less than 5000, updates the parameter value to default value.
timerIntervalToCheckAlarms	5	Indicates the time in minutes to check if alarmLimitForNotification is reached.	5 – 10	Minutes	-

Note: The parameter alarmThresholdForNotification must be updated before alarmThresholdForForceAck.

The value of the parameter alarmThresholdForNotification must be less than the value of the parameter alarmThresholdForForceAck.

5.20 Configure the Alarm Limits in ENM CLI

This section describes how to update maxNumberOfAlarmsInCli. The maxNumberOfAlarmsInCli configuration parameter is the maximum number of alarms that can be exported by ENM CLI commands alarm get and alarm hist.

The maximum number of alarms, which can be exported by ENM CLI commands alarm get and alarm hist, is updated.

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

5.20.1 Alarm ENM CLI Limits Parameters

Parameter Name	Default Value	Parameter Description	Value Range
maxNumberOfAlarmsInCli	10000	Indicates the maximum number of alarms that can be exported by ENM CLI command alarm get	1–120000

5.21 Configure the Transient Alarms

> This section describes the parameters to read, activate, or deactivate the Transient Alarm Staging Algorithm with which a minimum delay (in milliseconds) is guaranteed between an alarm raise and the corresponding clear, update, or repeat alarms This configuration minimizes the potential race condition with FMX processing alarms.

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

5.21.1 Transient Alarms Parameters

Parameter Name	Default Value	Parameter Description	Value Range
`transientAlarmStaging`	FALSE	The parameter represents the Enabled/Disabled state of the Transient Alarm Staging Algorithm.	N/A
`transientAlarmStagingThresholdTime`	`300`	The parameter value represents the miniumum time gap(in milliseconds) that is granted between each of the transient alarms by staging the same.	`50–1000`

5.22 Configure the FM NBI Common Parameters

This section describes how to update configuration parameters applied by FM NBI.

Access to the management server (MS).

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

5.22.1 Common Parameters for FM NBI

Alarms or events coming from Network Elements that support ERICSSONALARM-MIB 2.1 can be enriched with an extra attribute named `targetAdditionalInformation`.

ENM enables or disables the forwarding of the `targetAdditionalInformation` to NMS system through FM NBI with the `additionalInformationBlocker` parameter.

FM NBI forwards `targetAdditionalInformation` attribute to NMS system if the `additionalInformationBlocker` parameter is set to false.

Note: `additionalInformationBlocker` parameter is applied for all FM NBIs: CORBA, BNSI, and SNMP.

Prerequisites

Alarms or events coming from Network Elements that support ERICSSONALARM- MIB 2.1

Parameter Name	Default Value	Parameter Description	Value Range
additionalInformationBlocker	false	Indicates if targetAdditionalInformation is forwarded or not to NMS systems through FM NBI.	N/A

5.23 Configure FM BNSI NBI Configurable Parameters

> To configure FM BNSI NBI configurable parameters.

FM BNSI NBI provides the FM interface between the Ericsson Network Manager (ENM) FM and Network Management System (NMS).

NMS systems use the BNSI NBI interface to receive spontaneous alarms, events, and perform alarm operations such as acknowledging, terminating, and reading active alarms from ENM Fault Management.

The following types of configurable parameters are available:

— BNSI NBI Enable parameters

— BNSI NBI Application parameters

— BNSI NBI SSH Parameters

— BNSI NBI Profile parameters

— BNSI NBI Additional Information Order parameters

ENM BNSI Agent supports BNSI connection over SSH, RSH, and REXEC transport mechanism as per BNSI Protocol Specification.

The use of the RSH and REXEC protocols for connecting to ENM BNSI NBI is supported for backward compatibility towards the legacy product. However, this is discouraged. The RSH and REXEC protocols are considered highly vulnerable and open for exploitation. They can expose the systems as authentication data is sent in clear text which means that someone could gain unauthorized access to ENM.

This risk does not apply to BNSI sessions started over the SSH protocol.

5.23.1 bnsiNbiEnabled Parameters

The bnsiNbiEnabled parameters enable or disable the BNSI Agent.

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

Parameter Name	Parameter Description	Default Values
`bnsiNbiEnabled`	The parameter controls if the BNSI Agent is enabled or disabled. Set the parameter to `true` to enable the BNSI Agent (by default it is disabled).	false

The `bnsiNbiEnabled` parameters are two parameters that can be edited respectively by a configuration file and by the PIB.

To enable the BNSI NBI application only one `bnsiNbiEnabled` parameter (the parameter in the configuration file or the PIB parameter) can be configured to `true`.

To disable the BNSI NBI application both the `bnsiNbiEnabled` parameters in the configuration file and in the PIB must be configured to `false`.

5.23.1.1 Edit bnsiNbiEnabled Parameter in the Configuration File

To update or read the `bnsiNbiEnabled` parameter present in the configuration file.

The parameter is read every 10 seconds at runtime.

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

5.23.2 BNSI NBI Application Parameters

BNSI NBI application parameters control the BNSI Agent behavior.

Parameter Name	Parameter Description	Default Values
`AckAndTermSendResponse`	The parameter controls whether or not the response message is sent (by default it is not sent) on acknowledge and termination actions for all BNSI sessions that have been started without a valid `-actResp` (for more information, see FM BNSI Northbound Interface Integration Programmers Guide in CPI, Reference [2]. For BNSI sessions started with `-actResp`, this parameter is ignored. The parameter is reread every 10 seconds at runtime and	no

Parameter Name	Parameter Description	Default Values
	applied (according to the criteria above) on newly started BNSI sessions.	
AsyncRetrievedOnFirstBunch	The parameter controls the number of alarm records to be retrieved in the first bunch during any alarm synchronization (all network, single NE, extended or not). The parameter is reread every 10 second at runtime and is applied on newly started BNSI sessions.	10
MaxWaitingTimeForAlarmBunch	The parameter controls the interval time in seconds between two bunches of alarms during alarm synchronization (for all types of synchronization). The BNSI Agent uses this time for estimating the size of the alarm bunches after the first bunch so that the interval time is less than the specified parameter. The parameter is reread every 10 seconds at runtime and applied on newly started BNSI sessions.	5
AgentProfileCheckInterval	The parameter controls the time in seconds after which the directory containing the BNSI Agent Profiles is checked for updates (added, removed or modified profiles). The parameter is reread every 10 seconds at runtime and applied on newly started BNSI sessions.	10

5.23.2.1 Edit BNSI NBI Application Parameters

> To update or read the BNSI NBI Application parameters.

1. Log on to the relevant service, see Connect to a Service on page 2.
2. Go to the directory `/ericsson/tor/data/nbi/fm/conf`.
3. Edit the `BNSINBIConfiguration.properties` file for updating and reading the parameters described in the BNSI NBI Application Parameters on page 41

5.23.3 BNSI NBI SSH Parameters

> To configure the ssh parameters that define the Key re-exchange mechanism.

The BNSI NBI application must be re-started to apply the parameters changes.

Key Re-Exchange Mechanism

Key re-exchange mechanism is described into https://tools.ietf.org/html/rfc4253 and it is recommended that the keys are changed after each gigabyte of transmitted data or after each hour of connection time, whichever comes sooner.

Since the re-exchange is a public key operation, it requires a fair amount of processing power and must not be performed too often.

Parameter Name	Parameter Description	Default Values
`MaxBytes`	SSH Keys are changed after each `MaxBytes` GBytes of transmitted data as described in https://tools.ietf.org/html/rfc4253. `MaxBytes` value must be between 1 and `100000000`	1
`MaxKeyInterval`	SSH Keys are changed after `MaxKeyInterval` hours of connection time as described into https://tools.ietf.org/html/rfc4253. `MaxKeyInterval` value must be between `0` and `10`; if `MaxKeyInterval` is set to `0` the feature is disabled.	1
`idleTimeoutMs`	The connection is shut-down by BNSI after `idleTimeoutMs` minutes in case the client does not reply to the key re-exchange message. `idleTimeoutMs` value must be between 5 and 10.	10

5.23.3.1 Edit BNSI NBI SSH Parameters

To update or read the BNSI NBI Application parameters:

Steps

1. Log on to the relevant service, see Connect to a Service on page 2.
2. Go to the directory `/ericsson/tor/data/nbi/fm/conf`.
3. Edit the `BNSISSHConfiguration.properties` file for updating and reading the parameters described in the BNSI NBI SSH Parameters on page 42.
4. Restart the BNSI agent.

5.23.4 BNSI NBI Profile Parameters

BNSI NBI Profile parameters control the alarm filtering rules of the BNSI Agent profiles files. See FM BNSI Northbound Interface Integration Programmers Guide in CPI, Reference [2] and BNSI, Basic Network Surveillance Interface Protocol Specification in CPI, Reference [1].

Each file in the directory `/ericsson/tor/data/nbi/fm/conf`:

— is considered by BNSI Agent as a possible BNSI Agent profile for alarm filtering.

— is expected to be filled with the parameters reported in the following table.

— is applied on any BNSI session specifying that file as BNSI Agent profile n line with BNSI protocol specification. See BNSI, Basic Network Surveillance Interface Protocol Specification in CPI, Reference [1].

During the BNSI session start, if the BNSI Agent profile is not specified or not found, the default values are applied for alarm filtering.

The profile parameters are:

— re-read at runtime at an interval specified in the `AgentProfileCheckInterval` parameter (see BNSI NBI application parameters for more information).

— applied both on the existing and newly started BNSI sessions that have set the specific BNSI Agent profile at session start time.

5.23.4.1 FM BNSI NBI Profile Parameters

Parameter Name	Parameter Description	Default Values	Possible Values
`Perceived Severity`	The parameter controls the perceived severity levels of the alarms to be forwarded to the NMS: so alarms whose perceived severity is different from those specified are not forwarded. The parameter is a character string (no parameter name, just parameter value) that must be positioned in the row of the profile file. The perceived severity levels to forward are represented by single character codes. All character codes are listed in Possible Values	CcMmIW	**Indeterminate** I **Critical** C **Major** M **Minor** m **Warning** W **Cleared** c By default, alarms of all perceived severity levels are forwarded.
`Contained Object`	The `contained_object` parameter (name-value pair) controls alarm forwarding only from Managed Elements or also from Equipment and other types of objects that are children of the network elements specified in the configuration file.	contained_object=true	If the `contained_object` parameter is set to `true`, and no network elements are specified in the configuration file, alarms from the children of all network If the `contained_object` parameter is set to `false`, alarms from the children of the specified network element are not forwarded. If the `contained_object` parameter is set to `true`, alarms from the children of the

Parameter Name	Parameter Description	Default Values	Possible Values
			specified network element are forwarded.
Network Element List	The Network Element List parameter controls the network element (NE) from which alarms are forwarded. The syntax is a list of Full Distinguished Name (FDNs) of the NEs, one per row. All the alarms from the NEs that are listed in the Network Element List parameter are forwarded to BNSI NMS.	empty meaning all network elements	If alarms fulfill the Perceived Severity and Contained Object criteria specified in the BNSI Agent profile file but no valid profile is specified on the BNSI session, the alarms from all ENM NEs are forwarded. If alarms fulfill the Perceived Severity and Contained Object criteria specified in the BNSI Agent profile file but no NE FDN is specified in the BNSI Agent profile file, the alarms from all ENM NEs are forwarded.

Example of Perceived Severity cCmMIW

```
cCmMIW
contained_object=true
NetworkElement=LTE01ERBS00050
```

Any BNSI Session started, specifying this profile and with filtering enabled, receives all the fault notifications:

— raised by node LTE01ERBS00050

— regardless of the severity of the notification (all severity values are specified)

— regardless of the object affected by the fault (so even if it is an object contained by the network element LTE01ERBS00050)

Example of Perceived Severity cCMI

```
cCMI
contained_object=false
```

Any BNSI Session started, specifying this profile and with filtering enabled, receives all the fault notifications:

— raised by any node in ENM (no network element specified => all network elements)

— having a perceived severity either cleared or critical or major or indeterminate (minor and warning are not specified in the first row)

— and not raised by objects contained by the node, that is only fault raised at network element level

Example of Perceived Severity cC

```
cC
contained_object=true
```

```
NetworkElement=LTE01ERBS00050
NetworkElement=LTE01ERBS00051
NetworkElement=LTE01ERBS00052
```

Any BNSI Session started, specifying this profile and with filtering enabled, receives all the fault notifications:

— raised by one of the following nodes: LTE01ERBS00050, LTE01ERBS00051 and LTE01ERBS00052

— having a perceived severity cleared or critical (the only severity values specified in the first row)

— regardless of the object affected by the fault (so both faults at network element level and faults raised by objects that are contained by the network element)

5.23.4.2 Edit BNSI NBI Profile Parameters

To update or read the BNSI NBI profile parameters:

Steps

1. Log on to the relevant service, see Connect to a Service on page 2.
2. Go to the directory `/ericsson/tor/data/nbi/fm/bnsi`.
3. To update or read the parameter for an existing BNSI Agent profile, edit the existing `<bnsi-agent-profile-name>.config` file.
4. To add a new BNSI Agent profile, create a `<bnsi-agent-profile-name>.config` file containing the parameter of the table.
5. To remove an existing BNSI Agent profile, delete the related `<bnsi-agent-profile-name>.config` file.

5.23.5 BNSI NBI Additional Information Parameters Order

BNSI NBI Additional Information Order file provides the filter and the order for the BNSI additional attributes flow.

The `BNSIAdditionalInformationOrder.properties` file is initially empty and the BNSI alarm record shows all the additional attributes.

The BNSI Manager is enabled to fill the `BNSIAdditionalInformationOrder.properties` file with the names and the order of the required additional attributes to receive only the specified additional attributes in the required position.

After starting a new BNSI session, the BNSI alarm record shows the additional attributes accordingly to the new setting.

When the `BNSIAdditionalInformationOrder.properties` file is accidentally removed, the BNSI alarm record in the next started BNSI session shows all the additional attributes.

5.23.5.1 Edit BNSI NBI Additional Information Parameters Order

To update or read the BNSI NBI additional information order:

Steps

1. Log on to the relevant service, see Connect to a Service on page 2.
2. Go to the directory `/ericsson/tor/data/nbi/fm/conf`.
3. Edit the `BNSIAdditionalInformationOrder` file to read or update the attributes using the described filling scheme.

```
<AttributeName> "=" <id-position>
```

5.24 Configure CORBA NBI Configurable Parameters

This section describes the Fault Management (FM) Common Object Request Broker Architecture (CORBA) Northbound Interface (NBI) configurable parameters.

Provides an FM interface between Ericsson Network Manager (ENM) FM and the Network Management System (NMS). NMS systems use this interface to receive spontaneous alarms, events, and perform alarm operations such as acknowledge or unacknowledge and read active alarms from ENM FM.

Access to the Management Server.

The required parameters are modified and NBI reads the updated values while performing the operations.

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

5.24.1 FM CORBA NBI Configurable Parameters

Parameter Name	Default Value	Parameter Description	Value Range	Units	Possible Values
`alarmIteratorSize`	50	The maximum number of alarms retrieved in a getAlarmList before the Iterator method is invoked. If the number of alarms is less than this value, all alarms are returned in an array.	50–150	Integer	N/A
`alarmIteratorTimeout`	20000	Indicates that this timeout is approximately the time an NMS gets to handle an iterator of the size alarmIteratorSize.	20000–30000	Milliseconds	N/A

Parameter Name	Default Value	Parameter Description	Value Range	Units	Possible Values
DNPrefix	N/A	The DNPrefix is added to the alarm's full distinguished name in the following way to map the object to the NMS information structure: <DNPrefix>,SubNetwork=<subNetworkName>, ManagedElement=<managedElementName>	N/A	N/A	N/A
eventBatchSize	10	The number of structured events in the EventBatch before the batch is sent with the NotificationService. The events are only sent if the batch is filled up before the eventBatchTimer is reached.	10–50	Integer	N/A
eventBatchTimer	3	The time for which the structured events remain in the EventBatch before the batch is sent with the NotificationService. The events are only sent if the EventBatchSize has not been reached within the defined interval. The value is given in seconds.	1–10	Seconds	N/A
eventSubscription Category	1z1	Specifies the subscription category that is used for Event IRP structured events. This is the category that is used to forward FM error messages.	N/A	N/A	N/A
GMTOffset	0	The displacement from GMT in hours.	-12 to 14	Hours	N/A
GMTOffsetInMins	0	The displacement from GMT in minutes in addition to hours (GMTOffset).	0 - 45	Minutes	N/A
IRPVersion	0	The IRP version that is currently supported.	N/A	N/A	N/A
maxEventQueueSize	10000	The maximum number of alarms that the agent can buffer internally. This value affects the memory consumed by the CORBA Alarm IRP Agent. The queue is used temporarily to store alarms and error messages received from FM Kernel before they are processed by the CORBA Alarm IRP Agent.	1000–10000	Integer	N/A
notificationIDTypeLong	false	Specifies the format in which the notificationID is sent. The notification ID attribute is always a number but for backward compatibility reasons it can be coded either as a long or string value.	N/A	N/A	The value of the configuration parameter specifies how it is coded in the system: **True** Coded as a long value **False** Coded as a string value
notificationMode	1	Defines how the structured events are sent to the NotificationAgent.	1–2	Integer	If the value of the parameter is: **1** Structured events are sent one by one **2** Structured events are sent in a batch When the notificationMode is set to 1, the events are sent immediately and the value of eventBatchSize is ignored. When the notificationMode is set to 2, the parameters eventBatchSize and eventBatchTimer control the behavior of the batches.
routeFMEventsToNMS	true	Specifies the default handling of FM alarms.	N/A	N/A	If the parameter value is set to: **True** All alarms are forwarded to NMS

Parameter Name	Default Value	Parameter Description	Value Range	Units	Possible Values
					False No alarms are forwarded This parameter together with the alarm attribute RouteToNMS handles filtering of unwanted alarms to NMS.
`rejectErrorNotifications`	true	Defines whether FM error messages should be forwarded to NMS or not.	N/A	N/A	If the value of the parameter is: **True** Acknowledgements on error messages are not forwarded. **False** The original error messages and acknowledgments on error messages are forwarded as alarms. In either of the cases the original error messages are forwarded as it is. In alarm synchronisation from NMS by default no error messages are sent. However if this parameter is set to false, the error messages are forwarded as alarms.
`systemDN`	Ericsson OSS	The SystemDN parameter is the unique identifier of the OSS system. It must be the full distinguished name. This value must be updated for easy identification of the agent.	N/A	N/A	N/A

5.25 Configure CORBA NBI User Defined Attributes

This section describes how to update `userDefinedAttributesToSend` configuration parameter for configuring user defined attributes to be sent from Corba to NBI.

This parameter supports multiple attributes that are separated by a colon.

While sending alarm from CORBA to NBI, user defined attributes configured using `userDefinedAttributesToSend` parameter are sent along with the alarm if that attribute is either present in main alarm attributes or additional attributes.

When the `userDefinedAttributesToSend` configuration parameter is updated, the previous parameter value is overridden.

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

5.25.1 User Defined Attributes To Send Parameters

Parameter Name	Default Value	Parameter Description	Value Range
userDefinedAttributesToSend	""(empty string)	User defined attributes to be sent from CORBA to NBI	N/A

To configure more than one attribute, use ":" as a separator.

Example:

```
"value="attributeName1:attributeName2""
```

To remove all the configured values use the NONE value:

Example:

```
"value="NONE""
```

5.26 Configure FM SNMP NBI User Defined Attributes

The parameter nbiSnmpUserAdditionalInfo specifies a list of additional attributes to be sent to the SNMP Manager through the FM SNMP NBI in addition to the default attributes already sent.

The value of the parameter is a list of additional attributes names separated by semicolon with the following format:

```
AdditionalAttributeName1[:SnmpNbiAttributeName1];AdditionalAttributeName2[:SnmpNbiAttributeName2];...
```

The part indicated between [ ] is optional and specifies the names exported through the SNMP NBI that can differ from the additional attribute names, as reported in the ENM User Interfaces.

The SnmpNbi attribute name can be specified appending it to the additional attribute name separated by a colon (":").

If the SnmpNbiAttributeName is omitted, then the attribute name is the AdditionalAttributeName.

It is suggested to use short names for the attribute names to save bytes.

When the configurable parameter is set, any FM SNMP NBI Manager can receive the data related to the Additional Information.

The default content of the additional information can be removed using the configurable parameter additionalInformationBlocker. For more information, see Configure the FM NBI Common Parameters on page 39.

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

5.26.1 FM SNMP NBI User Defined Attribute

Parameter Name	Default Value	Parameter Description	Value Range	Possible Values
nbiSnmpUserAdditionalInfo	Empty string	This parameter defines a configurable parameter to specify the additional attribute name	All characters except: semicolon ";" and colon ":"	String

To update using shortname to save bytes.

Example:

```
--value="translateResult:TR;sequenceNumber:SN"
```

To remove all the configured values use the NONE value.

Example:

```
"value="NONE""
```

5.27 Extend ENM to Support MME and ECM Sending Alarms to Multiple ENMs

> This section describes the configurable parameter used for enabling ENM extension to support MME and ECM nodes sending alarms to multiple ENMs.

The ENM extension is available only for the nodes with the following NE Type:

— SGSN-MME

— vSGSN-MME

— ECM

5.27.1 Multiple ENM FM Support Parameter

Parameter Name	Default Value	Parameter Description	Value Range	Possible Values
multipleENMFMSupport	false	This parameter enables the FM supervision on the same Network Element on multiple ENMs as follows: — false:one network element must be supervised from a single ENM — true: one network element can be supervised from multiple ENMs.	Boolean	true or false

5.27.2 Add Multiple ENM FM Support

Prerequisites

Access to the ENM Servers.

Steps

1. Disable alarm supervision on the involved nodes on all the ENMs.
2. Update the `multipleENMFMSupport` configuration parameter. See Configuring PIB Parameters on page 8.
3. Read the `multipleENMFMSupport` configuration parameter. See Configuring PIB Parameters on page 8.
4. Enable the alarm supervision on the Network Elements to let the updated values effective.

Results

The ENM extension is enabled to support MME and ECM nodes sending alarms to multiple ENMs.

5.28 Configure the Alert Synchronization for ECM

During FM synchronization, ENM retrieves all alarms and all alerts from ECM. Alerts do not have any cease state and it is not possible to figure out if a

synchronized alert is still useful. After an FM synchronization, removed alerts, by means of manual or automatic acknowledge, can re-appear

This section describes the procedure to disable and enable the alert synchronization during the FM synchronization operation of ECM.

When alert synchronization is disabled only alarms are collected during the FM synchronization operation of ECM.

Prerequisites

User has root access to the ENM server.

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

5.28.1 Alert Synchronization for ECM Parameter

Parameter Name	Default Value	Parameter Description	Value Range	Possible Values
ecmDisableAlertSynchronization	false	This parameter disables alert synchronization for ECM: — true=disable alert synchronization — false=enable alert synchronization	Boolean	true or false

6 Automatic Alarm Handling (FMX) Administration Tasks

This section contains the routine operation and maintenance tasks related to the administration of Automatic Alarm Handling (FMX) applications.

6.1 FMX Server

6.1.1 FMX Framework

The FMX Framework is composed of:

— **Runtime environment** for event processing, rule logic execution, external actions, and module management.

— **Management environment** for module administration, configuration, and monitoring.

— **Development environment** for creating and verifying rules.

FMX Runtime Environment

The FMX Runtime Environment is composed of the following FMX Stack services:

FMX Module Server
Controls the Module Management (for example CLI and web UI), FMX Rule Engine, and FMX Rule Editor operations towards the FMX Module Repository.

FMX Inference Engine Server
Complex event stream processing and policy engine. The Rule and Utility Modules are loaded and configured in runtime.

FMX Action Server
Executes actions for interaction with external nodes or other external data sources on behalf of the FMX Engine Server.

Redis Cluster Shared memory between FMX services for events, FMX rule objects, FMX rule context and FMX rule policy parameters.

RabbitMQ Cluster

Message Queue for internal communication between FMX services.

FMX Management Environment

The FMX Management Environment is provided by the following UI applications:

Module Management
Facilitates rule and utility modules administration, modification of module runtime rule policy parameters, viewing active rule module triggers and viewing rule execution statistics.

Time Periods
Supports view and modification of time events and/or recurring time events (for example weekends, holidays, maintenance time, rush hour, and so on) for usage within the rule modules as triggers or rule logic conditions.

Monitor
Dashboard for monitoring the amount of processed events and executed rules.

FMX Development Environment

The FMX Development Environment is provided by the following UI applications:

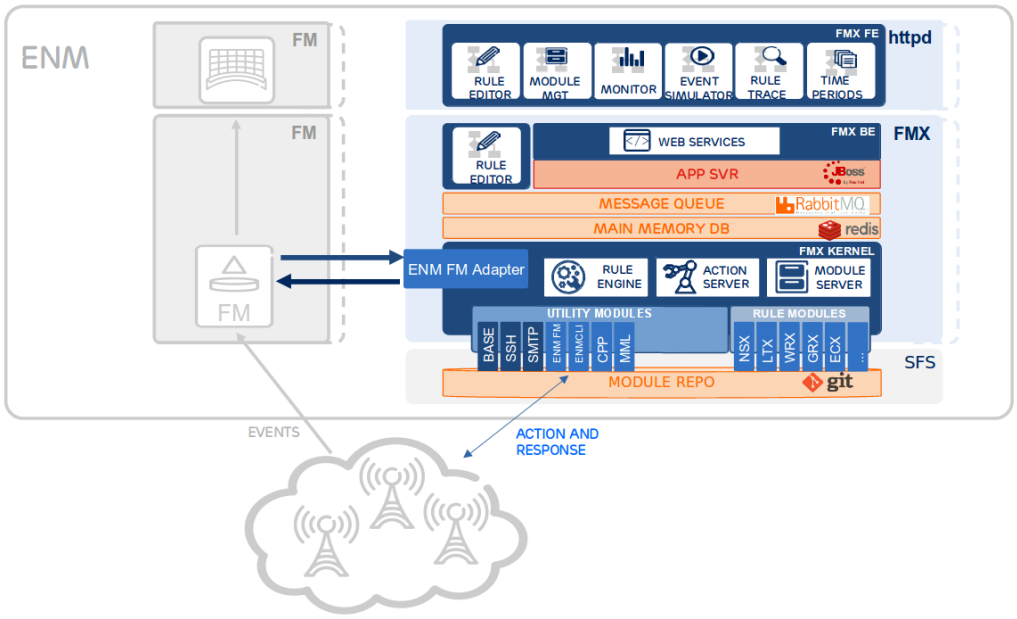
FMX Rule Editor
Graphical Rule IDE (Integrated Development Environment) for creation of event processing rules in form of logical flows with pluggable and configurable utility blocks.

FMX Rule Trace
Supports subscription to real-time block-by-block rule execution trace messages. Used for rule development and troubleshooting purposes.

FMX Event Simulator
Allows creation and injection of pre-defined alarm and error sequences into the FM system to facilitate Rule Modules validation.

Note: FMX web UI applications run on the JBoss EAP Application Server.

FMX Rule Editor is a thick client application and is launched in the FMX SG VM but using Cendio Thinlinc as Remote Desktop Server for displaying the X application.



6.1.1.1 High Availability

RabbitMQ

The FMX services use RabbitMQ for internal messaging within the FMX services.

RabbitMQ is deployed in cluster configuration along the FMX servers for High Availability and scalability purposes.

Redis

The FMX services use Redis as shared Main Memory Database (MMDB).

Redis is deployed in cluster configuration along the FMX servers for High Availability and scalability purposes.

6.1.1.2 Persistent Versus Stateless

FMX is NOT strictly "stateless".

FMX keeps runtime information in the Redis Cluster. This information does not persist if all FMX servers are stopped.

FMX uses shared file system as storage for persistent information:

User Space The user rule modules created from the Rule Editor are stored in the user space under `$HOME/.fmx/`. The $HOME is mounted and shared between all FMX servers.

Shared Space Loaded Modules, Event Simulator sequences, Time Periods, Rule, and Utility Modules configuration, and so on, are stored in the mounted shared file system under `/ericsson/tor/data/fmx/`.

FMX Module Repository
The Archived Modules are stored in a Control Version System within the shared file system.

6.1.2 FMX Services

The FMX SG contains several services that can be grouped as follows:

— FMX Stack Services: Provide FMX Runtime Environment.

— FMX Back-End Services: Provide Management and Development Environment.

— Supporting Services: Support common VM deployment, monitoring, and administration.

FMX Stack Services

The FMX Stack services are:

— `fmxas` - FMX Action Server LSB init service.

— `fmxie` - FMX Inference Engine LSB init service.

— `fmxms` - FMX Module Server LSB init service.

— `rabbitmq-server-cluster` - AMQP service provided by RabbitMQ broker.

— `redis-cluster` - Manages Redis Cluster Server local nodes.

The FMX Stack services have dependencies on the following RHEL standard services:

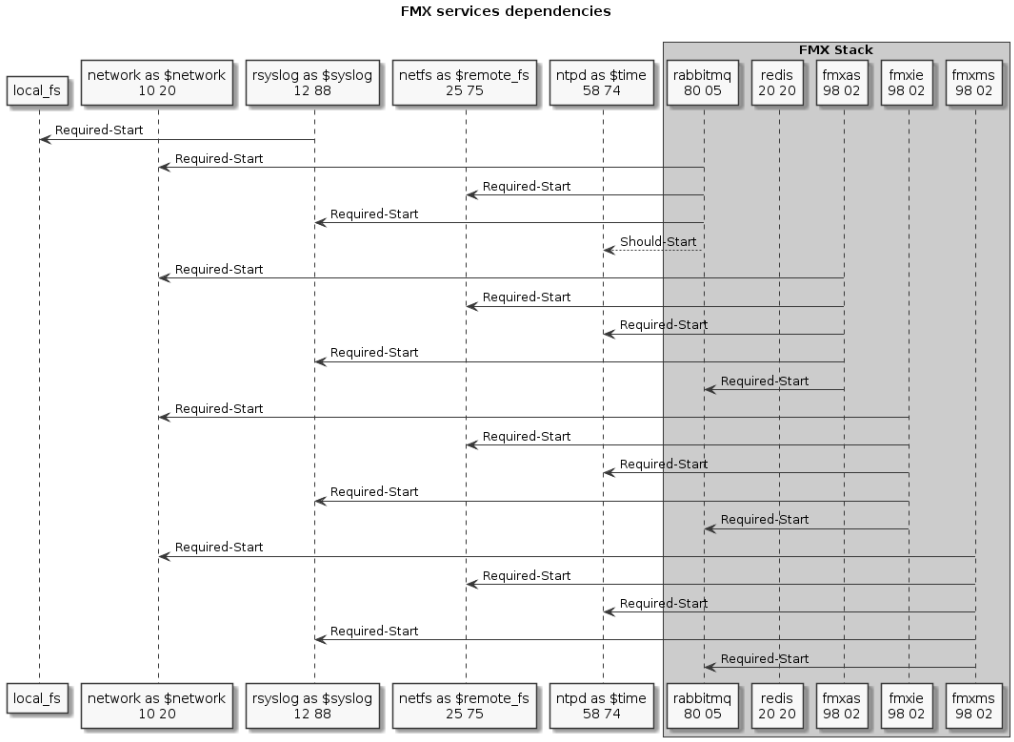
— `ntpd` - Network Time Protocol (NTP) daemon. The `ntpd` program is an operating system daemon that synchronizes the system clock with remote NTP time servers or local reference clocks.

— `netfs` - Network File Systems daemon. The `netfs` daemon mounts and unmounts Network File Systems.

— `rsyslogd` - Reliable and extended `syslogd`. The `rsyslogd` is a system utility that supports message logging.

— `network` - Activates or deactivates all the network interfaces configured to start at boot time.

The FMX Stack services dependencies are shown in the following graph:



FMX Back-End Services

The FMX back-end (`fmxadminws`) applications run on top of JBoss:

— jboss - JBoss LSB service

Supporting Services

As part of standard deployment, the following RHEL standard services are enabled:

— `crond` daemon to execute scheduled commands.

— `sshd` OpenSSH SSH daemon. The `sshd` provide secure encrypted communications between two untrusted hosts over an insecure network.

Also, for cloud deployment:

— `consul` `Consul` service. `Consul` is a highly available and distributed service discovery and KV store designed with support for the modern data center to make distributed systems and configuration easy.

6.1.3 FMX Directory Structure

This section describes the main directories used by FMX SG in FMX servers local file system and in the Shared File System (SFS).

6.1.3.1 FMX Servers Directory Structure

Services

The `FMX LSB System V init script` services are located in the `/etc/init.d` directory:

- `fmxas` - FMX Action Server LSB service
- `fmxie` - FMX Inference Engine LSB service
- `fmxms` - FMX Module Server LSB service
- `rabbitmq-server-cluster` - RabbitMQ LSB service
- `redis-cluster` - Redis LSB service
- `jboss` - JBoss LSB service

Configuration

The configuration files are provided with default configurations on deployment but can be modified if needed.

The directory `/etc/opt/ericsson/fmx` holds configuration for FMX services.

The FMX LSB services configuration directories are:

- `/etc/opt/ericsson/fmx/actionserver`
- `/etc/opt/ericsson/fmx/engine`
- `/etc/opt/ericsson/fmx/moduleserver`

The `rabbitmq`, `redis-cluster`, and `jboss` services configuration directories are:

- `/etc/rabbitmq/`
- `/etc/redis-cluster/`
- `/etc/jboss-as/`

Other FMX configuration directories are:

- `/etc/opt/ericsson/fmx/adapters` used by installed adapters.

— `/etc/opt/ericsson/fmx/cli` used by the FMX CLI.

— `/etc/opt/ericsson/fmx/editor` used by the FMX Rule Editor application.

Binaries, Scripts, and Libraries

The directory `/opt/ericsson/fmx` holds directories with static files, such as binaries, jar files, and scripts.

Note: Do not modify these files after installation.

FMX subdirectories:

— `actionserver` used by the FMX Action Server LSB service (`fmxas`).

— `cli` used by the FMX CLI (`fmxcli`).

— `editor` used by the FMX Rule Editor application.

— `engine` used by the FMX Inference Engine LSB service (`fmxie`).

— `moduleserver` used by the FMX Module Server LSB service (`fmxms`).

— `tools` diverse FMX utilities.

Logs

The directory `/var/log` contains logs from `LSB init` services.

FMX subdirectories:

— `/var/log/fmx`

— `/var/log/rabbitmq-server-cluster`

— `/var/log/redis`

— `/var/log/jboss-as/`

PIDs

The directory `/var/run` holds PID files for running `LSB init` services.

FMX subdirectories:

— `/var/run/fmx`

— `/var/run/rabbitmq-server-cluster/`

— `/var/run/redis/`

— `/var/run/jboss-as/`

6.1.3.2 Shared File System Directory Structure

FMX Persistent Storage

The mounted directory `/ericsson/tor/data/` and the subdirectories are shared between the servers.

FMX uses `/ericsson/tor/data/fmx` for keeping shared persistent data among the cluster nodes. Any changes made in the directory and in the subdirectories are reflected on each FMX server in the FMX cluster.

The historical folder `/var/opt/ericsson/fmx` is linked on each FMX server to the shared directory `/ericsson/tor/data/fmx` for OSS-RC backwards compatibility.

Subdirectories within `/ericsson/tor/data/fmx`:

- `dat` is the location for temporal data used by rules and utility modules.
- `etc` used for global configuration for the FMX services and modules.
- `export` is the temporal location for imported and exported modules.
- `log` is the location for custom modules log files.
- `modules` contains loaded modules. Each loaded module has the corresponding repository directory.
- `moduleserver/repos` contains the module archive (Git repositories). Each archived module has the corresponding Git repository directory.
- `sequences` contains Event Simulator sequences (.seq files).
- `timeperiods` contains for Time Periods containers (iCalendar .ics files).
- `var` is the location for temporal data used by rules and utility modules.

6.1.4 FMX Stack OCF Resource Agent

The FMX Stack services are managed by the FMX Stack Open Cluster Framework (OCF) resource agent.

An OCF-compliant cluster resource agent is an executable that manages a cluster resource.

The `fmxstack` OCF script supports the following subset of standard operations:

- `start` starts the FMX stack services in the server.
- `stop` starts the FMX stack services in the server.

— `monitor` monitors the health of the FMX stack services in the server.

Also, the `fmxstack` OCF script supports the following custom operation:

— `pacemaker` configures and restores the FMX stack services in the server.

The FMX stack OCF script is located in `/usr/lib/ocf/resource.d/fmxstack`.

For more information on OCF resource agents, see http://www.linux-ha.org/wiki/OCFResourceAgents.

FMX Stack Pacemaker Service

The `fmxstack-pacemaker` service provides simple continuous self-configuration and self-healing for FMX stack services.

As long as the `fmxstack-pacemaker` service is started, the `fmxstack pacemaker` command is executed every 30 seconds.

On every `fmxstack pacemaker` execution, the file `/var/log/fmx/fmxstack.status` is updated with the FMX stack health status.

The `fmxstack-pacemaker` service is started by `/etc/rc.local` on FMX server start.

FMX Stack Services Start

The FMX stack services are configured and started on FMX server start.

The script `/etc/rc.local` is executed after all the normal system services are started, at the end of the process of switching to a multiuser run level. It is traditionally used to start custom services.

The following snippets from `/etc/rc.local` take care of FMX stack services start:

```
# Enable fmxstack pacemaker
/sbin/service fmxstack-pacemaker start

# Run fmxstack pacemaker
/opt/ericsson/fmx/tools/bin/fmxstack pacemaker
```

FMX Stack Services Stop

The FMX stack services are stopped on FMX server stop.

Any scripts within `/usr/lib/ocf/pre_shutdown/` are executed before stopping an ENM SG.

The file `/usr/lib/ocf/pre_shutdown/fmxstack-stop` is a symbolic link to `/opt/ericsson/fmx/tools/bin/fmxstack-stop`.

The `fmxstack-stop` script simplifies the graceful stop of FMX stack services:

```
# Disable fmxstack pacemaker
/sbin/service fmxstack-pacemaker stop

# Stop the fmxstack services
/opt/ericsson/fmx/tools/bin/fmxstack stop
```

FMX Stack Monitor

Any scripts within `/usr/lib/ocf/resource.d/` are executed at regular intervals for monitoring the server health status.

The FMX stack services are continuously monitored by using `fmxstack` monitor. The `fmxstack` monitor returns the exit code stored in the `/var/log/fmx/fmxstack.status` file from the latest `fmxstack` pacemaker execution. The `fmxstack` monitor returns error if no file or if the file is not recently updated.

During troubleshooting, it can be necessary to stop and start some of the FMX stack services. Restarting the FMX stack processes in one server does not impact event processing because of N-active FMX service group configuration.

The `fmxstack` monitor reports that the FMX server is offline as long as any of the FMX stack services health-check fails.

6.2 Service Management

To start and stop the services on FMX, see Restarting a Service on page 6.

6.2.1 Manage FMX Services

The FMX services are managed by the following `LSB System V init` services in the FMX servers:

- `fmxie` - FMX Inference Engine
- `fmxms` - FMX Module Server
- `fmxas` - FMX Action Server

The FMX services are configured and started automatically on FMX server deployment.

Syntax

The `fmxie` service syntax is:

```
Usage: /etc/init.d/fmxie {start|stop|restart|condrestart|try-res →
tart|reload|status}
```

The fmxie, fmxms, and fmxas services share the syntax.

The services must be managed with root privileges.

The services run under nmxadm credentials.

Commands

Start an FMX service:

```
[root@<fmxserver>]# service <service> start
```

Show the status of an FMX service:

```
[root@<fmxserver>]# service <service> status
```

Stop an FMX service:

```
[root@<fmxserver>]# service <service> stop
```

Restart an FMX service:

```
[root@<fmxserver>]# service <service> restart
```

Examples

Start fmxie service:

```
[root@svc-1-fmx ~]# service fmxie start
Starting fmxie: [  OK  ]
```

Check fmxie service status:

```
[root@svc-1-fmx ~]# service fmxie status
fmxie (pid  12345) is running...
```

Stop fmxie service:

```
[root@svc-1-fmx ~]# service fmxie stop
Stopping fmxie:[  OK  ]
```

Check fmxie service status after stopped:

```
[root@svc-1-fmx ~]# service fmxie status
fmxie is stopped
```

Restart fmxie service:

```
[root@svc-1-fmx ~]# service fmxie restart
Stopping fmxie: [  OK  ]
Starting fmxie: [  OK  ]
```

6.2.2 Manage FMX RabbitMQ Cluster Service

The FMX RabbitMQ Cluster is managed by the custom `rabbitmq-server LSB System V init` service in the FMX servers.

The configuration and creation of the FMX RabbitMQ Cluster is performed automatically on FMX servers deployment.

Syntax

The `rabbitmq-server` service syntax is:

```
Usage: /etc/init.d/rabbitmq-server {start|stop|status|rotate-logs|restart|condre →
start|try-restart|reload|force-reload| cluster-status|status-cluster|pacemaker}
```

The service must be managed with `root` privileges.

The service runs under `rabbitmq` credentials.

Commands

Start the `rabbitmq-server` service:

```
[root@<fmxserver>]# service rabbitmq-server start
```

Show a detailed status of the `rabbitmq-server` service:

```
[root@<fmxserver>]# service rabbitmq-server status
```

Create a `rabbitmq-server` cluster with existing node in current server or join existing nodes in current server to existing `rabbitmq-server` cluster on remote server:

```
[root@<fmxserver>]# service rabbitmq-server pacemaker
```

Show the status of the `rabbitmq-server` cluster and its members:

```
[root@<fmxserver>]# service rabbitmq-server status-cluster
```

Stop the `rabbitmq-server` service:

```
[root@<fmxserver>]# service rabbitmq-server stop
```

Restart the `rabbitmq-server` service:

```
[root@<fmxserver>]# service rabbitmq-server restart
```

Examples

Check `rabbitmq-server` service status:

```
[root@svc-1-fmx /]# service rabbitmq-server status
Status of node 'rabbit@10.0.0.21' ...
[{pid,4969},
 {running_applications,
     [{rabbitmq_management,"RabbitMQ Management Console","3.6.6"},
      {rabbitmq_web_dispatch,"RabbitMQ Web Dispatcher","3.6.6"},
      {webmachine,"webmachine","1.10.3"},
      {mochiweb,"MochiMedia Web Server","2.13.1"},
      {rabbitmq_management_agent,"RabbitMQ Management Agent","3.6.6"},
      {rabbit,"RabbitMQ","3.6.6"},
      {mnesia,"MNESIA  CXC 138 12","4.12.5"},
      {os_mon,"CPO  CXC 138 46","2.3.1"},
      {ssl,"Erlang/OTP SSL application","6.0"},
      {public_key,"Public key infrastructure","0.23"},
      {crypto,"CRYPTO","3.5"},
      {amqp_client,"RabbitMQ AMQP Client","3.6.6"},
      {rabbit_common,[],"3.6.6"},
      {inets,"INETS  CXC 138 49","5.10.6"},
      {compiler,"ERTS  CXC 138 10","5.0.4"},
      {xmerl,"XML parser","1.3.7"},
      {syntax_tools,"Syntax tools","1.6.18"},
      {asn1,"The Erlang ASN1 compiler version 3.0.4","3.0.4"},
      {ranch,"Socket acceptor pool for TCP protocols.","1.2.1"},
      {rabbitmq_clusterer,"Declarative RabbitMQ clustering",[]},
      {sasl,"SASL  CXC 138 11","2.4.1"},
      {stdlib,"ERTS  CXC 138 10","2.4"},
      {kernel,"ERTS  CXC 138 10","3.2"}]},
 {os,{unix,linux}},
 {erlang_version,
     "Erlang/OTP 17 [erts-6.4] [source] [64-bit] [smp:2:2] [async-threads:64] [h →
ipe] [kernel-poll:true]\n"},
 {memory,
     [{total,57832064},
      {connection_readers,0},
      {connection_writers,0},
      {connection_channels,0},
      {connection_other,2808},
      {queue_procs,2808},
      {queue_slave_procs,0},
      {plugins,628848},
      {other_proc,18563480},
      {mnesia,71888},
      {mgmt_db,573800},
      {msg_index,51984},
      {other_ets,1445072},
      {binary,40408},
      {code,27860281},
      {atom,992409},
      {other_system,7598278}]},
 {alarms,[]},
 {listeners,[{clustering,25672,"::"},{amqp,5672,"::"}]},
 {vm_memory_high_watermark,0.4},
 {vm_memory_limit,2012980838},
 {disk_free_limit,50000000},
 {disk_free,9309081600},
 {file_descriptors,
     [{total_limit,31900},
      {total_used,2},
      {sockets_limit,28708},
      {sockets_used,0}]},
 {processes,[{limit,1048576},{used,233}]},
 {run_queue,0},
 {uptime,4039},
 {kernel,{net_ticktime,60}}]
```

Check `rabbitmq-server` cluster status:

```
[root@svc-1-fmx /]# service rabbitmq-server status-cluster
Configured nodes : rabbit@10.0.0.21, rabbit@10.0.0.22
Running nodes    : rabbit@10.0.0.21, rabbit@10.0.0.22
Gospel           : rabbit@10.0.0.21
```

Stop `rabbitmq-server` service on FMX server 2:

```
[root@svc-2-fmx /]# service rabbitmq-server stop
Stopping rabbitmq-server: rabbitmq-server.
```

Check `rabbitmq-server` cluster status on FMX server 2:

```
[root@svc-2-fmx /]# service rabbitmq-server status-cluster
Error when executing /usr/sbin/rabbitmqctl eval 'rabbit_clusterer:status().', co →
de=17664, output=Error: unable to connect to node 'rabbit@10.0.0.22': nodedown

DIAGNOSTICS
===========

attempted to contact: ['rabbit@10.0.0.22']

rabbit@10.0.0.22:
  * connected to epmd (port 4369) on 10.0.0.22
  * epmd reports: node 'rabbit' not running at all
                  other nodes on 10.0.0.22: ['rabbitmq-cli-73']
  * suggestion: start the node

current node details:
- node name: 'rabbitmq-cli-73@svc-2-fmx.no-domain'
- home dir: .
- cookie hash: Vtc6Em4xwfeV/vq5Sq3gtA==
```

Check `rabbitmq-server` cluster status from FMX server 1:

```
[root@svc-1-fmx /]# service rabbitmq-server status-cluster
Configured nodes : rabbit@10.0.0.21
Running nodes    : rabbit@10.0.0.21
Gospel           : rabbit@10.0.0.21
```

Start back the `rabbitmq-server`

```
 service in FMX server[root@svc-2-fmx /]# service rabbitmq-server start
Starting rabbitmq-server: rabbitmq-server.
```

Check again `rabbitmq-server` cluster status:

```
[root@svc-2-fmx /]#  service rabbitmq-server cluster-status
Configured nodes : rabbit@10.0.0.21, rabbit@10.0.0.22
Running nodes    : rabbit@10.0.0.21, rabbit@10.0.0.22
Gospel           : rabbit@10.0.0.21 service in FMX server
                2:
```

6.2.3 Manage FMX Redis Cluster Service

The FMX Redis Cluster is managed by the custom `redis-cluster LSB System V init` service in the FMX servers.

The configuration and creation of the FMX Redis Cluster is performed automatically on FMX servers deployment.

Syntax

The `redis-cluster` service syntax is:

```
Usage: /etc/init.d/redis-cluster {start|stop|restart|reload|condrestart|try-rest
art|status| cluster-status|status-cluster|pacemaker}
```

The service must be managed with `root` privileges.

The service runs under `redis` credentials.

Commands

Start the `redis-cluster` service:

```
[root@<fmxserver>]# service redis-cluster start
```

Note: The `redis-cluster` service starts a pre-configured number of Redis nodes on each FMX server.

Show the status of the `redis-cluster` service nodes on the current FMX server (lists Redis nodes processes status in the current server):

```
[root@<fmxserver>]# service redis-cluster status
```

Show a detailed status of the `redis-cluster` and the nodes members:

```
[root@<fmxserver>]# service redis-cluster status-cluster
```

Create an FMX Redis Cluster with existing nodes in current server or join existing nodes in current server to existing FMX Redis Cluster on remote server:

```
[root@<fmxserver>]# service redis-cluster pacemaker
```

Stop the `redis-cluster` nodes on the current server:

```
[root@<fmxserver>]# service redis-cluster stop
```

Restart the `redis-cluster` nodes on the current server:

```
[root@<fmxserver>]# service redis-cluster restart
```

Examples

Start the Redis service in an FMX server (`svc-1-fmx`):

```
[root@svc-1-fmx ~]# service redis-cluster start
Starting redis-server instance 10.0.0.21:7001...           [  OK  ]
Starting redis-server instance 10.0.0.21:7002...           [  OK  ]
Starting redis-server instance 10.0.0.21:7003...           [  OK  ]
Starting redis-server instance 10.0.0.21:7004...           [  OK  ]
Starting redis-server instance 10.0.0.21:7005...           [  OK  ]
Starting redis-server instance 10.0.0.21:7006...           [  OK  ]
```

Check `redis-cluster` service nodes process status on an FMX server (`svc-1-fmx`):

```
[root@svc-1-fmx ~]# service redis-cluster status
redis-server (pid  299) is running...
redis-server (pid  311) is running...
redis-server (pid  327) is running...
redis-server (pid  341) is running...
redis-server (pid  355) is running...
redis-server (pid  369) is running...
```

Check the status of the FMX Redis Cluster from an FMX server (`svc-1-fmx`):

```
[root@svc-1-fmx ~]# service redis-cluster status-cluster
470d44b8df4dac63da59fddb4af05a7ef73ef369 10.0.0.21:7001 myself,master - 0 0 15 c →
onnected 5461-8192
0ca0e5e9e1f7249ac6c5590a2550efc54b893a67 10.0.0.21:7002 master - 0 1519315621742 →
 16 connected 10923-13653
58451d6b95d77ae9710e13c4904b75915b33bdfd 10.0.0.21:7003 master - 0 1519315621081 →
 17 connected 0-2730
be6758c60c57934f7a1f33142d12957050097c43 10.0.0.21:7004 master - 0 1519315620168 →
 18 connected 2731-5460
40adba32f8ac5bf9757848500437d99bc7ac5768 10.0.0.21:7005 master - 0 1519315620675 →
 19 connected 8193-10922
651b91d7f0ecc9189ca25f682df31fc44894c4a5 10.0.0.21:7006 master - 0 1519315620168 →
 20 connected 13654-16383
3272598eefd8b8d52b34e11509781899618c48dd 10.0.0.22:7001 slave 470d44b8df4dac63da →
59fddb4af05a7ef73ef369 0 1519315621193 15 connected
b2bdbbdd583954e5a34ee8882f1b7f5aedfe1649 10.0.0.22:7002 slave 0ca0e5e9e1f7249ac6 →
c5590a2550efc54b893a67 0 1519315621081 16 connected
5907d88e1a0c51621e8cb12bd8e0477b36074ec4 10.0.0.22:7003 slave 58451d6b95d77ae971 →
0e13c4904b75915b33bdfd 0 1519315621192 17 connected
de234350ed15220d92e7ba87a0ff65146365de1c 10.0.0.22:7004 slave be6758c60c57934f7a →
1f33142d12957050097c43 0 1519315621742 18 connected
8338dad29120268ca9ecf359e794ac2a4004a430 10.0.0.22:7005 slave 40adba32f8ac5bf975 →
7848500437d99bc7ac5768 0 1519315620575 19 connected
ee5efa950ad27daa854f2aa994e337ff3ac1fe5e 10.0.0.22:7006 slave 651b91d7f0ecc9189c →
a25f682df31fc44894c4a5 0 1519315621742 20 connected
cluster_current_epoch:20
cluster_known_nodes:18
cluster_my_epoch:15
cluster_size:6
cluster_slots_assigned:16384
cluster_slots_fail:0
cluster_slots_ok:16384
cluster_slots_pfail:0
cluster_state:ok
cluster_stats_messages_received:654140
cluster_stats_messages_sent:654742
exit_status:0
```

Stop the `redis-cluster` service in an FMX server (`svc-1-fmx`):

```
[root@svc-1-fmx ~]# service redis-cluster stop
Stopping redis-server instance 10.0.0.21:7001...           [  OK  ]
Stopping redis-server instance 10.0.0.21:7002...           [  OK  ]
```

```
Stopping redis-server instance 10.0.0.21:7003...           [  OK  ]
Stopping redis-server instance 10.0.0.21:7004...           [  OK  ]
Stopping redis-server instance 10.0.0.21:7005...           [  OK  ]
Stopping redis-server instance 10.0.0.21:7006...           [  OK  ]
```

6.3 Service Configuration

6.3.1 Configure a User and Cendio Thinlinc to Launch FMX Rule Editor

> The FMX Rule Editor is a thick client application that is launched from Automated Alarm Handling (FMX) web user interface (UI).

The thick application is executed in the FMX VM for the logged user and the X display is exported into `Cendio Thinlinc` session in a browser tab.

6.3.1.1 Configure a User to Launch FMX Rule Editor

To launch the FMX Rule Editor, the user needs to have the FMX_Administrator role and an extra user role with POSIX credentials (such as Element_Manager_Operator).

Prerequisites

User has access to the ENM User Management application.

Steps

1. Click **User Management** under Security in ENM Application Launcher.
2. Select the user account that you need to modify and click **Edit Profile**.
3. Select a user role configured with POSIX credentials (Element_Manager_Operator).

   Element_Manager_Operator is required for sticky sessions but other POSIX roles can be used to connect to `Cendio Thinlinc`.
4. Make sure the role FMX_Administrator is selected. Select it now if it is not selected.
5. Click **Save Changes** and exit User Management.

Results

The user can launch the FMX Rule Editor from the Automated Alarm Handling (FMX) web UI.

6.3.1.2 Configure a Cendio Thinlinc to Launch FMX Rule Editor

The user has to configure and enable the Cendio Thinlinc screen saver functionality to all the remote desktop users.

Note: If the FMX Rule Editor is opened but the ENM user is inactive for more than the configured idle time, then the rule editor session disconnects automatically and any not saved changes are discarded.

Prerequisites

User has access to the ENM User Management application.

Steps

1. Configure the Cendio thinlinc screen server enable functionality with the following parameters values.

Parameter	Value
Enabled	True
CountdownTime	30
IdleTime	3600

See Configure Timeout on Cendio Thinlinc Sessions in [7].

Results

The Cendio Thinlinc screen saver is enabled and configured.

6.3.2 Create System User for Automatic Alarm Handling (FMX) Interaction with ENM CLI

Automatic Alarm Handling (FMX) interacts with the ENM CLI. To interact with the ENM CLI, the FMX service uses a configurable ENM user for authentication and authorization.

The system administrator must create a specialized FMX ENM CLI system user and authentication credentials as a post installation step.

Prerequisites

— You can log in to ENM Launcher with Administrator role.

The username and password are fully configurable. The following procedure uses example values.

Steps

1. Log in to the ENM Launcher as a user with an Administrator role.

2. Open the User Management application.
3. Create a user (for example, `fmxenmcli`), with the following properties:
   - Name: `fmxenmcli`
   - Roles for `cmedit get`: `Cmedit_Operator`
   - Roles for alarm operations: `FM_Operator`
   - Roles for `cmedit get`, `create`, `delete`, `set`, `action`, and so on: `Cmedit_Administrator`
   - Password: Set a temporary password since this must be changed immediately.
   - Enabled: Set the user status to `Enabled` to allow it to log in.
4. Log out of the ENM Launcher.
5. Log in again as the previously created user (`fmxenmcli` in this example).

   When the password change request appears, set a new password.
6. Log in to one of the FMX instance. For more information, see Connect to a Service on page 2.
7. Switch to the user `nmxadm`:

   ```
   sudo su nmxadm
   ```
8. Run the following command:

   ```
   /opt/ericsson/fmx/configEnmCliCredentials/bin/configEnmCliCredentials
   ```

   The tool requests a username and password, these need to be lined up with the user created in Step 1, Step 2, Step 3, Step 4, and Step 5.

   Note: When the tool requests the password, the characters are not printed to screen.

   The tool encrypts the password and then writes both the username and password to the file `/ericsson/tor/data/fmx/etc/enm_adapter/fmxenmcli.properties`. If the file exists, the tool prompts you whether you want to overwrite the existing file.
9. Run the following command to force an immediate topology refresh (optional):

   ```
   /opt/ericsson/fmx/cli/bin/fmxcli -c flush_topology
   ```

Note: If this procedure is not followed, the only impact is within the Automated Alarm Handling (FMX) as follows:

- The user is unable to see the ENM node topology information in the Automated Alarm Handling Module Administration application within the "Activate for" module operation until FMX automatically refreshes topology, which happens every 60 minutes. However, the user still has the possibility to manually type one or more nodes for the "Activate for..." operation".
- The FMX ENM CLI block uses the same configured user credentials to automatically execute ENM CLI commands if configured in the FMX rules. Therefore, any FMX rules using the FMX ENM CLI block fails if the credentials are not configured correctly. The allowed operations depend on the configured Roles assigned to that user.
- If the credentials are not configured correctly, FMX fails to log in and the relevant access security logs are filled with errors.

Results

FMX can execute ENM CLI commands. For example, fetch the ENM node list.

6.3.3 Create System WinFIOL MML Users to Support MML Rule Execution

Automatic Alarm Handling (FMX) interacts with the WinFIOL MML mediation server to execute MML commands towards AXE-based switches.

To send MML commands, the block must be configured with a specialized ENM WinFIOL username.

Prerequisites

- You can log in to ENM Launcher with Administrator role.

The username and password are fully configurable. The following procedure uses example values.

Steps

1. Log in to the ENM Launcher as a user with an Administrator role.
2. Open the User Management application.
3. Create a user (for example, `fmxenmmml`), with the following properties:
   - Name: `fmxenmmml`
   - Roles for MML commands: `WinFIOL_Operator`, `Element_Manager_Operator`

- Password: Set a temporary password since this must be changed immediately.
- Enabled: Set the user status to `Enabled` to allow it to log in.

4. Click **Save Changes** and exit User Management.
5. Log out of the ENM Launcher.
6. Log in again as the previously created user (`fmxenmmml` in this example).

   When the password change request appears, set a new password.

Results

FMX is able to run rules executing ENM MML commands with the created MML User.

6.3.4 Configure CPP Utility to Support AMOS Execution

> Automatic Alarm Handling (FMX) has the capability to interact with CPP-capable nodes using AMOS. For that, the FMX service uses a configurable ENM user for authentication and authorization.

As a post installation step, the system administrator must configure a specialized system user, authentication credentials, and host names as follows:

- Load the CPP utility module
- Configure a specialized FMX AMOS system user
- Configure the FMX AMOS authentication credentials
- Configure the FMX AMOS host names

Prerequisites

- You have access to the ENM User Management application.
- You have ssh access to one of the FMX nodes.
- You have ssh access to one of the AMOS nodes.
- The CPP Utility module must be loaded.

Steps

The user name and password are fully configurable. The following procedure uses example values.

1. Log on to the ENM Launcher with a user with Administrator role.
2. Open the User Management application.
3. Create a new user (for example `fmxamos`) with the following properties:

Name	`fmxamos`
Roles	`Amos_Operator`, `Element_Manager_Operator`, `SystemAdministrator`, `BasebandSupportExpert`, `NodeB_Application_Administrator`, and `ENodeB_Application_Administrator`, **Note:** When selecting `COM Roles`, the `Target Groups Assigned` attribute must be set to `COM Target Groups(ALL)` while default value is `COM Target Groups(NONE)`.
Password	`<password>`
Force Password Change	`disabled`

Note: For more information on COM Roles and their description, see the Predefined COM Roles section in ENM identity and Access Management System Administrator Guide in CPI, Reference [9].

4. Log in to ENM Launcher as the new created user and click **Shell Terminal (SSH on AMOS VM)**.
5. Enter:

```
amos
```

   — If getting an error, then the owner and group of the new user directories in the `/ericsson/log/amos/moshell_logfiles/` path must be corrected.

   — If no errors enter:

```
exit
```

   and go to Step 9.

6. Log in to one of the AMOS instance. For more information, see Connect to a Service on page 2.
7. Change ownership and group:

```
chown -R fmxamos:enm_users /ericsson/log/amos/moshell_logfiles/fmxamos
```

8. Verify that the `fmxamos/logs_moshell` directories have the correct owner `fmxamos` and group `enm_users` entering the following command:

```
[fmxamos@scp-1-amos(enmHost) ~]$ ls -lrta /ericsson/log/amos/moshell_logfile →
s/fmxamos
```

Example

```
[fmxamos@scp-1-amos(enmHost) ~]$ ls -lrta /ericsson/log/amos/moshell_logfile →
s/fmxamos
total 16
drwxrwxrwt. 22 fmxamos enm_users 8192 Jun 24 16:35 logs_moshell
drwxrwxrwt.  4 fmxamos enm_users   96 Oct 24 15:28 .
drwxrwxrwt.  4 fmxamos enm_users   96 Oct 28 12:47 logs_mobatch
drwxrwxrwt. 16 root    root      8192 Nov 27 09:47 ..
[fmxamos@scp-1-amos(enmHost) ~]$
```

9. Verify that the new created user can run a command to a node, entering:

```
amos <nodename>
```

10. Verify that the needed passwordless login is possible, entering:

```
lt all
```

— If the `fmxamos` user gets prompted for the nodes username and password, check again Note.

Example

Unsuccessful passwordless command.

```
SELNWE003ENB11> lt all
191128-16:22:23 10.37.144.130 19.0h  stopfile=/tmp/1496
Trying temporary_amos_fmxamos_571's password from amos lookupAccount ... Not →
 OK
Trying temporary_amos_fmxamos_571's password from amos lookupAccount ... Not →
 OK
Please enter Username: labuser
Please enter labuser's Node Password:
```

Example

Successful passwordless command.

```
[fmxamos@scp-2-amos(enmHost) ~]$ amos selnwe003enb11
..
SELNWE003ENB11> lt all
191128-15:43:19 10.37.144.130 19.0h  stopfile=/tmp/23945
Trying temporary_amos_fmxamos_571's password from amos lookupAccount ... OK
$ssh_pid = 1821
Connected to 10.37.144.130 (SubNetwork=5G,ManagedElement=selnwe003enb11)
Checking MOM version...MSRBS_NODE_MODEL_19.Q4_423.28075.62_679d
```

```
Parsing MOM (cached): /var/tmp/20191128-153026_23884/MSRBS_NODE_MODEL_19.Q4_
423.28075.62_679d.xml.cache.gz .................Done.

Using paramfile /opt/ericsson/amos/moshell/commonjars/pm/PARAM_MSRBS_19.Q3.t
xt

Parsing file /opt/ericsson/amos/moshell/commonjars/pm/PARAM_MSRBS_19.Q3.txt
.....................................................Done.

Using imomdfile /opt/ericsson/amos/moshell/commonjars/pm/IMOMD_MSRBS_19.Q3.t
xt

Using imomfile /opt/ericsson/amos/moshell/commonjars/pm/IMOM_MSRBS_19.Q3.txt

Connected to 10.37.144.130 (SubNetwork=5G,ManagedElement=selnwe003enb11)

Last MO:  8429. Loaded 8429 MOs. Total:  8430 MOs.
```

11. Log in to one of the FMX instance. For more information, see Connect to a Service on page 2.

12. Switch to the user `nmxadm`:

    ```
    sudo su nmxadm
    ```

13. Run the `aes` utility to get an encrypted string containing the previously configured password:

    ```
    [<user>@<fmxnode>]$ /var/opt/ericsson/fmx/modules/cpp-block-module/aes --enc
    rypt '<password>'
    <encrypted password string>
    ```

14. Obtain AMOS server host names.

    — For physical environment execute the following command:

    ```
    [<user>@<fmxserver>]$ cat /etc/hosts | grep amos
    ```

    — For cloud environment execute the following command with root user:

    ```
    [root@<fmxserver>]# consul members | grep amos
    ```

15. Configure the CPP properties file located in: `/var/opt/ericsson/fmx/etc/modules/cpp-block-module/cpp-block-module.cfg`.

    The following items must be configured in the CPP properties file:

    — AMOS user name with the previously configured user name

    — AMOS password with the previously obtained encrypted password string

    — AMOS servers with the previously obtained AMOS server host names.

    ```
    # AMOS server user
    AMOS_USER=fmxamos
    # AMOS server password
    AMOS_PASSWORD=<encrypted password string>
    ```

```
# AMOS servers
AMOS_SERVERS=scp-1-amos,scp-2-amos
```

Results

FMX is able to run rules executing ENM AMOS commands.

6.4 Use the FMX CLI

In addition to the Automatic Alarm Handling (FMX) Module Management GUI, it is possible to manage the FMX modules using the command line interface FMX CLI from any of the FMX nodes.

Syntax

```
$ fmxcli (-h |--help)
```

```
$ fmxcli (-v |--version)
```

```
$ fmxcli [(-n|--dry-run)] [(-f |--file) <FILENAME>] [-c <COMMAND >]
```

where:

-n --dry-run Read and parse commands, but do not perform any actions.

-f <FILENAME> --file <FILENAME>
Read commands from `<FILENAME>`

-c <COMMAND> Execute `<COMMAND>`.

-v --version Output version info.

-h --help Output this help.

-c help List available commands.

-c help <COMMAND>
Output help for `<COMMAND>`.

6.4.1 Available Commands in the FMX CLI

Command	Description	Syntax
`activate`	Activate a loaded module.	`activate <modulename>`

Command	Description	Syntax
activatefor	Activate a loaded module for the list of selected nodes. The module will not be activated if the list is empty.	activatefor <modulename>
backup	Create backup files of the current module states and parameter values	backup
deactivate	Deactivate an activated module.	deactivate <modulename>
delete	Delete a selected version or all versions of a module from the archive. If the module is loaded, the loaded version will not be deleted.	delete <modulename> -v <version>|all
describe	Display detailed description of a loaded module.	describe <modulename>
event	Send an event to a rule engine module. Optionally attribute(s) can be added. If module option is omitted, the event is sent to all modules.	event <eventname> [-m <modulename>] [attr1=<val1> [attr2=<val2> [...]]]
exec	Run commands from a file.	exec <filename1> [<filename2> [...]]
exit	Exit the fmxcli application.	exit
export	Export a module from the archive into an fmx container file. If no path is entered, the file is stored in the /var/opt/ericsson/fmx/export directory.	export <modulename> [-p <path>] [-v <version>]
flush_topology	Clear the cache of the topology. The cache will be recreated automatically.	flush_topology
func	Run an FMX expression.	func <FMX_functional_expression>
help	Show the list of available commands, or details for one command.	help [command]
import	Import module file(s) into the archive.	import <filename1> [<filename2> [...]]
list	List loaded modules.	list
list_archive	List all modules available in the archive.	list_archive
list_elements	List available network model objects.	list_elements
list_params	List parameters for selected module.	list_params <modulename>
load	Load module(s).	load <modulename1> [-v <version1>] [<modulename2> [-v <version2>] [...]]
quit	Exit the fmxcli application.	quit
restore	Restore module states and parameter values from backup files	restore
set_param	Set parameter value in module.	set_param <modulename> <namespace> <parametername> <value>
setobj	Set a list of network model objects a module shall be activated for.	setobj <modulename> [<object1> [<object2> [...]]]
statistics	Show statistics for module.	statistics <modulename>
unload	Unload a loaded module.	unload <modulename>
version	Show the version of fmxcli.	version

6.4.2 Use FMX CLI Commands

1. Import `baseblock-module` utility:

```
[<user>@svc-1-fmx ~]$ cd /var/opt/ericsson/fmx/export
[<user>@svc-1-fmx export]$ fmxcli -c import baseblocks-module-15.31.fmx
```

2. Load two modules (start the fmxcli prompt by typing `fmxcli` and pressing Enter):

```
[<user>@svc-1-fmx ~]$ fmxcli
> load baseblocks-module NSX_Configuration
```

3. List loaded modules:

```
[<user>@svc-1-fmx ~]$ fmxcli -c list
Loaded modules:
name                    type             version  active         simulation    →
    description
baseblocks-module               UTILITYMODULE   15.67            false         →
 FMX Core Base Blocks
cpp-block-module                UTILITYMODULE   15.46            false         →
 Support for node interaction via CPP protocol
enm-blocks-module               UTILITYMODULE   16.11            false         →
 ENM Event Blocks
enmcli-blocks-module            UTILITYMODULE   16.5            false          →
Support for direct usage of ENM CLI feature
eNodeB_Manually_Locked          RULEMODULE      16.0    No        false         →
  RoD for reacting to UNIX event generated when the eNB cell is manually loc →
ked and forgotten
exclusive-region-module         UTILITYMODULE   15.42            false         →
 Flow control exclusive-regions blocks, with overflow and priority control
LTX_eNodeB_down                 RULEMODULE      15.7    No        false         →
  LTX - LTE eXpert Rules Package
mail-block-module               UTILITYMODULE   15.29            false         →
 Support for basic SMTP. Send e-mail with possible attachments
multi-attribute-module          UTILITYMODULE   15.30            false         →
 Set multiple Object attributes
NSX_Configuration               RULEMODULE      16.15    No        false        →
   NSX - Network Surveillance eXpert Rules Package
NSX_Event_to_Alarm              RULEMODULE      16.10    Yes        false       →
    NSX - Network Surveillance eXpert Rules Package
NSX_Hard_filter                 RULEMODULE      16.7    No        false         →
  NSX - Network Surveillance eXpert Rules Package
NSX_Maintenance_filter          RULEMODULE      16.7    No        false         →
  NSX - Network Surveillance eXpert Rules Package
NSX_Shortlived_and_Frequent     RULEMODULE      16.14   No        false         →
  NSX - Network Surveillance eXpert Rules Package
semaphore                       UTILITYMODULE   15.35            false         →
 Flow control semaphore rule blocks
ssh-block-module                UTILITYMODULE   15.35            false         →
 Support for node interaction via SSH protocol
```

4. Activate a rule module:

```
[<user>@svc-1-fmx ~]$ fmxcli -c activate NSX_Configuration
```

5. Display information about a loaded module:

```
[<user>@svc-1-fmx ~]$ fmxcli -c describe NSX_Shortlived_and_Frequent
    Name:        NSX_Shortlived_and_Frequent
    Version:     16.14
    Type:        Rule
    State:       Inactive
    Active:      No
    Locking mode:     R
    Parameters:
        SL_FRQ.Filter_Time = 3
        SL_FRQ.Frequent_Interval = 60
        SL_FRQ.Frequent_Limit = 5
        SL_FRQ.Frequent_Poll_Time = 10
```

```
        SL_FRQ.Hide_During_Frequent = false
    Active for:     [None]
    Deactive for:     [All]
    Requires:     BASEBLOCKS-MODULE; ENM-BLOCKS-MODULE; EXCLUSIVE-REGION-MOD →
ULE; NSX_CONFIGURATION; NSX_MAINTENANCE_FILTER
```

6. Display module statistics:

```
[<user>@svc-1-fmx ~]$ mxcli -c statistics NSX_Shortlived_and_Frequent
Rule Name;Block Type;Block ID;Event/Name;Counter
MODULE_DEACTIVATED;MODULE-ACTIVATION-TRIGGER;2;NSX_Shortlived_and_Frequent A →
CTIVATED;0
Short-lived alarm filtering and Frequent alarm creation;OSSRC-EVENT-TRIGGER- →
FILE;11;ShortLivedEvent;0
Frequent alarm handling;OSSRC-EVENT-TRIGGER-FILE;19;GenericFrqEvent;0
MODULE_DEACTIVATED;MODULE-ACTIVATION-TRIGGER;16;NSX_Shortlived_and_Frequent  →
DEACTIVATED;0
Short-lived alarm filtering and Frequent alarm creation;UPDATE-TRIGGER;38;Sh →
ortLivedEvent;0
MODULE_DEACTIVATED;UNIX-TRIGGER;26;updateSLF;0
```

7. Backup loaded modules, states, and parameters:

```
[<user>@svc-1-fmx ~]$ fmxcli -c backup
```

8. Restore loaded modules, states, and parameters::

```
[<user>@svc-1-fmx ~]$ fmxcli -c restore
```

6.4.3 FMX Backup and Restore with GeoRedundancy

In GeoRedundancy there are two independent ENM systems:

— Primary ENM site that is in active state and runs ENM services.

— Secondary ENM site that is in passive state and is active when the Primary ENM site is OFF.

The backup and restore commands provide:

— the scheduled export on the Primary ENM system on regular basis.

— the data copy from the Primary ENM system to the Secondary ENM system.

— the data import on regular basis.

As part of the GeoRedundancy Solution setup, a separate `GeoR NFS` directory (around 40GB) is created for the export of the data.

By default backup and restore files are in:

`/ericsson/georeplication/.workingDir/FMX/` directory.

The `nmxadm` user must have the permissions to write contents.

Directory and file name can be changed modifying the `backup.filename` value inside the `enm_utilities_compabilitytable.properties` configuration file, located in `/etc/opt/ericsson/fmx/moduleserver`.

All the required applications configuration data are exported to the respective application directory in the `GeoR NFS` directory. The replication data on the Secondary ENM is made transferring a created single tar file..

The current implementation collects the following data:

— information on all loaded modules and parameters on the Primary ENM

— configuration and data files

— archived versions of the loaded modules

The information is compressed in a tar file in the geo replication shared file system. The tar file is manually transferred to the same location on the Secondary ENM system.

The restore command extracts the tar file, restores parameter values, and imports modules if necessary. All the active modules are deactivated, and all rule modules, not part of the backup, are unloaded. All modules are reloaded.

Note: For backwards compatibility, the highest revision of the same utility module present on the secondary ENM is used instead.

All modules restore the same state of activation and simulation as on the Primary ENM. For example if a module was active for only a selection of the topology, the module is active for the same selection of the topology on the Secondary ENM. If the backup is restored on an ENM with a different topology, the module needs manual actions to select a new set of topology, to activate the module for the whole network, or to deactivate the module.

Usage:

```
fmxcli -c backup
```

Manually transfer backup file to the same location on the Secondary ENM system:

```
fmxcli -c restore
```

6.5 Log Management

Each of the FMX stack services creates its own log file(s) on each FMX node.

service	description	logging file
fmxie	FMX Engine Server	`/var/log/fmx/fmxie.log`

service	description	logging file
	FMX Engine Server (rule trace)	`/var/log/fmx/fmxie_ruletrace.log`
fmxms	FMX Module Server	`/var/log/fmx/fmxms.log`
fmxas	FMX Action Server	`/var/log/fmx/fmxas.log`
fmxadminws	FMX Admin Web Services	`/var/log/fmx/fmxws.log`
rabbitmq-service-cluster	RabbitMQ server's Erlang log file	`/var/log/rabbitmq/rabbit@<hostname>.log`
	RabbitMQ server's Erlang SASL (System Application Support Libraries) log file.	`/var/log/rabbitmq/rabbit@<hostname>-sasl.log`
redis-cluster	Redis instance running on <instance_port> (for example, if redis-cluster is configured to run with 6 instances per cluster node, <instance_port> is 7001-7006)	`/var/log/redis/redis-<instance_port>.log`

Configuration

The log level and retention policy for the each FMX stack service are configurable.

The provided defaults are optimal for most of the cases but specific procedures are provided here so the customer can adapt the log level and retention policy to its specific needs or modify it for troubleshooting purposes.

Forwarding

For central logging management, the FMX stack services logging messages (except rule trace) are forwarded into rsyslog service on each node.

The rsyslog daemon is configured to send log messages to the central log database.

6.5.1 Modify Log Level for FMX Log Files

The logging level for the FMX services can be modified for troubleshooting purposes.

FMX logging level is managed by log4j2 configuration on each of the FMX services and defined in the `/etc/opt/ericsson/fmx/<service>/log4j2.xml` configuration file, created by FMX packages (ERICfmx<service>-<version>.noarch.rpm).

The configuration snippet below shows the predefined configuration for the `fmxms` service logfiles. For space reasons, only the `loggers` section from the configuration file is shown.

```
    <loggers>
        <Logger name="com.ericsson" level="info" additivity="false">
            <AppenderRef ref="allLogRollingFileAppender"/>
        </Logger>
        <root level="fatal">
            <AppenderRef ref="console" level="fatal"/>
```

```
            <AppenderRef ref="sysLogAppender" level="info"/>
        </root>
    </loggers>
```

This is, there are three log appenders (AppenderRef) defined: `logfile`, `console` and `syslog`. Each of them is configured with independent log levels.

You can change this default log level configuration to apply to specific customer policies.

For more details on specific log4j2 configuration syntax and options, see *https://logging.apache.org/log4j/2.x/manual/configuration.html*.

6.5.1.1 FMX Logging Configuration Files

service	description	logging configuration file
fmxie (LSB)	engine server	`/etc/opt/ericsson/fmx/engine/log4j2.xml`
fmxms (LSB)	module server	`/etc/opt/ericsson/fmx/moduleserver/log4j2.xml`
fmxas (LSB)	action server	`/etc/opt/ericsson/fmx/actionserver/log4j2.xml`
fmxadminws (jboss)	admin web services	`/ericsson/tor/data/fmx/etc/fmxadminws/log4j2.xml`

6.5.1.2 Level Filtering

The following table illustrates how level filtering works in log4j2. In the table, the vertical header shows the level of the `LogEvent`, while the horizontal header shows the level associated with the appropriate `LoggerConfig`. The intersection identifies whether the `LogEvent` is allowed to pass for further processing (YES) or discarded (NO).

Event Level	LoggerConfig Level						
	TRACE	DEBUG	INFO	WARN	ERROR	FATAL	OFF
ALL	YES	YES	YES	YES	YES	YES	NO
TRACE	YES	NO	NO	NO	NO	NO	NO
DEBUG	YES	YES	NO	NO	NO	NO	NO
INFO	YES	YES	YES	NO	NO	NO	NO
WARN	YES	YES	YES	YES	NO	NO	NO
ERROR	YES	YES	YES	YES	YES	NO	NO
FATAL	YES	YES	YES	YES	YES	YES	NO
OFF	NO	NO	NO	NO	NO	NO	NO

6.5.1.3 Modify FMX Log Files Log Level

The configuration changes are stored in the local file system. Therefore:

— the changes apply only to the modified node(s).

— the changes are not persistent in the case of a node reinstall or a node upgrade.

Prerequisites

User is logged on as `root`.

Steps

1. Open the logging configuration file (for example for `fmxms`) with a file editor (for example, `vi`):

   ```
   [root@<fmxnode>]# vi /etc/opt/ericsson/fmx/moduleserver/log4j2.xml
   ```

2. Modify the log level.
3. Save the changes and exit the file editor.
4. Repeat for the other FMX nodes.

6.5.2 Modify Log Level for RabbitMQ Log Files

The logging level for the RabbitMQ service can be modified for

troubleshooting purposes.

RabbitMQ logging level is configured in the `/etc/rabbitmq/rabbitmq.config` configuration file.

The variable `log_levels` controls the granularity of logging. The value is a list of log event category and log level pairs.

The level can be one of

— None

— Error

— Warning

— Info

— Debug

At present there are four categories defined (other, currently uncategorised, events are always logged):

channel for all events relating to AMQP channels

connection for all events relating to network connections

federation for all events relating to federation

mirroring for all events relating to mirrored queues

In the following RabbitMQ configuration snippet, the categories `connection` and `channel` are specified with their default settings. Each of them can be configured with independent log levels.

```
%% -*- mode: erlang -*-
[
  {rabbitmq_management,[{load_definitions, "/etc/rabbitmq/definitions.json"}]},
  {rabbitmq_clusterer,[{config, "/var/opt/ericsson/fmx/etc/rabbitmq/rabbitmq-clu →
ster.config"}]},
  {rabbit, [ {log_levels, [{connection, info}, {channel, info}]} ] }
].
```

You can change this default log level configuration to apply to specific customer policies.

For more details on specific RabbitMQ configuration syntax and options, see *https://www.rabbitmq.com/configure.html and https://www.rabbitmq.com/relocate.html*.

6.5.2.1 RabbitMQ Log Files

RabbitMQ creates the following log files:

Name	Description	Location
RABBITMQ_LOGS	RabbitMQ servers Erlang log file	/var/log/rabbitmq/rabbit@<hostname>.log
RABBITMQ_SASL_LOGS	RabbitMQ servers Erlang SASL (System Application Support Libraries) log file.	/var/log/rabbitmq/rabbit@<hostname>-sasl.log

6.5.2.2 Modify RabbitMQ Log Files Log Level

The configuration changes are stored in the local file system. Therefore:

— the changes apply only to the modified node(s).

— the changes are not persistent in the case of a node reinstall or a node upgrade.

Prerequisites

User is logged on as `root`.

Steps

1. Open the logging configuration file with a file editor (for example, `vi`):

```
[root@<fmxnode>]# vi /etc/rabbitmq/rabbitmq.config
```

Note: To connect a service, see Connect to a Service on page 2.

2. Modify the log level.
3. Save the changes and exit the file editor.
4. Restart the `rabbitmq-server-cluster` LSB service in that node.

```
[root@<fmxnode>]# service rabbitmq-server-cluster restart
```

Note: To start and stop the services on FMX, see Restarting a Service on page 6.

5. Repeat for the other FMX nodes.

Results

After modifying the log level configuration, the changes do not take effect immediately. It is necessary to restart the service.

6.5.3 Modify Log Level for Redis Log Files

The logging level for the Redis services can be modified for troubleshooting purposes.

Redis logfile location and logging level are configured by default in the `/etc/redis.conf` configuration file.

The redis instances are managed from the custom `redis-cluster` LSB script. Each redis instance has a separate configuration file under `/etc/redis-cluster/redis-<instance>.conf`.

The following is a snippet of the default configuration file for redis instance running on port `7001`, `/etc/redis-cluster/redis-7001.conf`:

```
# Specify the server verbosity level.
# This can be one of:
# debug (a lot of information, useful for development/testing)
# verbose (many rarely useful info, but not a mess like the debug level)
# notice (moderately verbose, what you want in production probably)
# warning (only very important / critical messages are logged)
loglevel notice

# Specify the log file name. Also the empty string can be used to force
# Redis to log on the standard output. Note that if you use standard
# output for logging but daemonize, logs will be sent to /dev/null
logfile /var/log/redis/redis-7001.log
```

Each redis instance creates its own logfile under `/var/log/redis` directory:

```
/var/log/redis/redis-<instance>.log
```

where <instance> is the port the redis instance is listening on.

Redis has four log levels:

— debug

— verbose

— notice

— warning

The default log level is set to notice.

You can change the default log level configuration to apply to specific customer policies.

6.5.3.1 Log Level with VI

The configuration changes are stored in the local file system. Therefore:

— the changes apply only to the modified node(s)

— the changes are not persistent in the case of a node reinstall or a node upgrade.

Prerequisites

You are logged on as `root`.

Steps

1. Open the redis instance logging configuration file with a file editor (for example, `vi`):

   ```
   [<user>@<fmxnode>]$ vi /etc/redis-cluster/redis-7001.conf
   ```

   Note: To connect a service, see Connect to a Service on page 2.

2. Modify the log level.
3. Save the changes and exit the file editor.
4. Repeat for the other configuration files for each Redis instance in that node.
5. Restart the `redis-cluster` LSB service in that node. To start and stop the services on FMX, see Restarting a Service on page 6.
6. Repeat for the other FMX nodes.

6.5.3.2 Log Level on Runtime

The configuration changes are applied directly to the running instance and no configuration is saved. Therefore:

— the changes apply only to the modified node(s)

— the changes are not persistent in the case of a node reinstall or a node upgrade.

Steps

1. use the `CONFIG SET` command from the `redis-cli` to set the loglevel on runtime:

```
[<user>@<fmxnode>]$ redis-cli -c -p 7001 -h localhost CONFIG SET loglevel de
bug
OK
```

2. Repeat for the other configuration files for each Redis instances in that node.

   In this case the changes take effect immediately and it is not necessary to restart the service.

3. Repeat for the other FMX nodes.

6.5.4 Modify Log Retention Policy for FMX Log Files

The logging rotation and deletion for the FMX services can be configured to meet specific customer retention policy requirements.

Logging rotation and deletion are managed by log4j2 configuration on each of the FMX services and defined in the `/etc/opt/ericsson/fmx/<service>/log4j2.xml` configuration file, created by FMX packages (ERICfmx<service>-<version>.noarch.rpm).

FMX Logging Configuration Files on page 84

The configuration snippet below shows the predefined configuration for the `fmxms` service logfiles. For space reasons, only the `RollingRandomAccessFile` section from the configuration file is shown.

```
        <RollingRandomAccessFile name="allLogRollingFileAppender"
                                 fileName="${logFolder}/fmxms.log"
                                 filePattern="${logFolder}/fmxms-%d{yyyy-MM-dd}T%
d{HH-mm}_%i.log.gz"
                                 immediateFlush="false">
            <PatternLayout pattern="%d{ISO8601};%p;[%t];%logger;%replace{%msg%th
rowable}{\n}{_ENDL_}\n"/>
            <Policies>
                <CronTriggeringPolicy schedule="0 0 0 * * ?"/>
                <SizeBasedTriggeringPolicy size="20 MB" />
            </Policies>
            <DefaultRolloverStrategy>
                <Delete basePath="${logFolder}" maxDepth="1">
```

```
                    <IfFileName glob="fmxms-*.gz">
                        <IfAccumulatedFileSize exceeds="100 MB" />
                    </IfFileName>
                </Delete>
            </DefaultRolloverStrategy>
        </RollingRandomAccessFile>
```

This is, by default, the `fmxms` logfile is rotated every day (managed by `CronTriggeringPolicy`) or if log files size exceeds 20 MB (managed by `SizeBasedTriggeringPolicy`), whichever happens first. The rotated files are renamed based on `filePattern`. If the accumulated rotated files size matching the given delete pattern is more than 100 MB (managed by `IfAccumulatedFileSize`), the older files are deleted and only the most recent 100 MB is kept.

You can change this default log rotation and retention configuration to apply to specific customer policies.

For more details on specific log4j2 configuration syntax and options, see *https://logging.apache.org/log4j/2.x/manual/appenders.html#RollingRandomAccessFileAppender*.

6.5.4.1 Modify FMX Log Files Log Retention Policy

The configuration changes for the LSB services are stored in the local file system. Therefore:

— the changes apply only to the modified node(s).

— the changes are not persistent in the case of a node reinstall or a node upgrade.

Prerequisites

User is logged on as `root`.

Steps

1. Open the logging configuration file (for example for `fmxms`) with a file editor (for example, `vi`):

   ```
   [root@<fmxnode>]# vi /etc/opt/ericsson/fmx/moduleserver/log4j2.xml
   ```

   To connect a service, see Connect to a Service on page 2.

2. Modify the log retention policy.
3. Save the changes and exit the file editor.
4. Repeat for the other FMX nodes.

Results

After modifying the log file configuration, the changes take effect immediately. It is not necessary to restart the service.

6.5.5 Modify Log Retention Policy for RabbitMQ Log Files

The logging rotation and deletion for the RabbitMQ service can be configured to meet specific customer retention policy requirements.

Logging rotation and deletion policy are managed by logrotate and defined in the `/etc/logrotate.d/rabbitmq-server` configuration file, created by RabbitMQ package (rabbitmq-server-<version>.noarch.rpm).

The configuration snippet below shows the predefined configuration for the RabbitMQ service logfiles.

```
/var/log/rabbitmq/*.log {
        weekly
        missingok
        rotate 20
        compress
        delaycompress
        notifempty
        sharedscripts
        postrotate
            /sbin/service rabbitmq-server rotate-logs > /dev/null
        endscript
}
```

This is, by default, the RabbitMQ log files are rotated weekly, log files are rotated 20 copies before being removed and rotated files are compressed.

You can change this default log rotation and retention configuration to apply to specific customer policies.

For more details on specific logrotate configuration syntax and options, see *https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Deployment_Guide/s2-log_rotation.html*, *http://www.linuxcommand.org/man_pages/logrotate8.html* and *https://linuxconfig.org/logrotate*.

6.5.5.1 Modify RabbitMQ Log Files Log Retention Policy

The configuration changes for logrotate are stored in the local file system. Therefore:

— the changes apply only to the modified node(s).

— the changes are not persistent in the case of a node reinstall or a node upgrade.

Prerequisites

User is logged on as `root`.

Steps

1. Open the logging configuration file with a file editor (for example, `vi`):

   ```
   [root@<fmxnode>]# vi /etc/logrotate.d/rabbitmq-server
   ```

   Note: To connect a service, see Connect to a Service on page 2.

2. Modify the log rotation and retention policy.
3. Save the changes and exit the file editor.
4. Repeat for the other FMX nodes.

Results

After modifying the log file configuration, the changes take effect immediately. It is not necessary to restart the service.

6.5.6 Modify Log Retention Policy for Redis Log Files

The logging rotation and deletion for the Redis service can be configured to meet specific customer retention policy requirements.

Logfile rotation and deletion policy are managed by logrotate and defined in the `/etc/logrotate.d/redis` configuration file, created by Redis package (redis-<version>.el6.x86_64.rpm).

The configuration snippet below shows the predefined configuration for the Redis service logfiles.

```
/var/log/redis/*.log {
    weekly
    rotate 10
    copytruncate
    delaycompress
    compress
    notifempty
    missingok
}
```

This is, by default, the Redis log files are rotated weekly, log files are rotated 10 copies before being removed and rotated files are compressed.

You can change this default log rotation and retention configuration to apply to specific customer policies.

For more details on specific logrotate configuration syntax and options, see *https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Deployment_Guide/s2-log_rotation.html*, *http://www.linuxcommand.org/man_pages/logrotate8.html* and *https://linuxconfig.org/logrotate*.

6.5.6.1 Modify Redis Log Files Log Retention Policy

The configuration changes for logrotate are stored in the local file system. Therefore:

— the changes apply only to the modified node(s).

— the changes are not persistent in the case of a node reinstall or a node upgrade.

Prerequisites

User is logged on as `root`.

Steps

1. Open the logging configuration file with a file editor (for example, `vi`):

   ```
   [root@<fmxnode>]# vi /etc/logrotate.d/redis
   ```

2. Modify the log rotation and retention policy.
3. Save the changes and exit the file editor.
4. Repeat for the other FMX nodes.

Results

After modifying the log file configuration, the changes take effect immediately. It is not necessary to restart the service.

6.6 Security

6.6.1 FMX Redis Cluster Security

This section contains a general description and the configuration of the Redis 3PP in FMX, as follows:

— *Redis Authentication Feature* section describes the background how the authentication feature works in Redis 3PP. The purpose is to provide the system administrator a deeper understanding for how the authentication feature is implemented and works in Redis.

— *FMX Redis Cluster Authentication Configuration* section describes how the Redis authentication feature is configured in the FMX SG. This is what the system administrator needs to follow to configure the feature.

6.6.1.1 Redis Authentication Feature

While Redis does not try to implement Access Control, it provides a tiny layer of authentication that can be turned on editing the `redis.conf` file.

When the authorization layer is enabled, Redis refuses any query by unauthenticated clients. A client can authenticate itself by sending the AUTH command followed by the password.

The password is set by the system administrator in clear text inside the `redis.conf` file. It must be long enough to prevent brute-force attacks for two reasons:

— Redis is very fast at serving queries. Many passwords per second can be tested by an external client.

— The Redis password is stored inside the `redis.conf` file and inside the client configuration, so it does not need to be remembered by the system administrator, and thus it can be very long.

The goal of the authentication layer is to optionally provide a layer of redundancy. If firewalling or any other system implemented to protect Redis from external attackers fails, an external client is not able to access the Redis instance without knowledge of the authentication password.

The `AUTH` command, like every other Redis command, is sent unencrypted, so it does not protect against an attacker that has enough access to the network to perform eavesdropping.

6.6.1.2 FMX Redis Cluster Authentication Configuration

The FMX Redis Cluster is formed by a pre-configured number of Redis nodes per FMX server. The Redis authentication feature is enabled by default in FMX Redis Cluster deployment.

The FMX Redis Cluster has no default authentication password configured on initial installation (as per security guidelines).

The FMX Redis Cluster is accessed by:

— Previous existing and newly added Redis cluster nodes.

— The customized `redis-cluster LSB init` service script, used to manage the set of FMX Redis Cluster nodes within each FMX host.

— The FMX services (`fmxie`, `fmxms`, `fmxas`, `fmxws`), which use the FMX shared cache memory within the FMX Redis Cluster.

— The `redis-cluster-auth` tool, used for setting Redis cluster authentication password.

All the listed entities need to access FMX Redis Cluster authorization configuration.

For seamless deployment and upgrade of FMX VMs, the FMX Redis Cluster authentication password requires to be stored in shared storage.

For compatibility between cloud and physical deployments, the authentication password shared storage is a file in the shared file system. Authentication is only enabled if the file is not empty.

The shared file contents are encrypted using AES. The encryption key and initialization vector are stored in FMX servers local file system for security reasons. A default key and IV (initialization vector) are provided on installation.

Change Authentication Password

The FMX Redis Cluster is formed by a pre-configured number of Redis nodes per FMX server. The authentication password is changed centrally for all the FMX Redis Cluster nodes with the `/bin/redis-cluster-auth` tool:

```
[root@fmxengine1 /]# /bin/redis-cluster-auth
    Changing password for Redis Cluster
    New password: secreto
    Retype new password: secreto
    Apply to all Redis Cluster nodes ? (y/[n]): y
    Redis Cluster password changed.
```

The tool stores the new password encrypted in the shared file system in the `/ericsson/tor/data/fmx/redis/.authpass`:

```
[root@fmxengine1 /]# cat /ericsson/tor/data/fmx/redis/.authpass
    4VHiWftCu6elAw1b1vqolc/B7ekyRpMP0YNVwJnKHwYnbMmsfa97AUZylF9Y →
GKhm
    FzhJp64ys4aLCeUm+QM1/Zr9XpN5+l+ZxYxR0T1HcsA=
```

The tool also creates a local copy of that file under `/tmp/ericsson/tor/data/fmx/redis/.authpass`. This way the FMX clients are resilient upon shared file system temporary unavailability.

The details of the password change operation are logged to `syslog` and can be read from `/var/log/messages` directory.

FMX Redis Cluster Clients

The FMX Redis Cluster clients consume the FMX Redis Cluster authentication configuration:

— The `redis-cluster LSB init` service script reads if authentication is enabled and an authentication password is configured and operates the FMX Redis Cluster nodes using that authentication password. On start operation, it applies the existing authentication password in the `/etc/redis-cluster/redis-<port>.config` files.

— The FMX services (`fmxie`, `fmxms`, `fmxas`, `fmxws`) read on process start if authentication is enabled and an authentication password is configured and use it for establishing connection with FMX Redis Cluster nodes. The FMX services watch also for any changes in the `$AUTHPASS_FILE` for re-establishing the connection with the updated authentication password.

— The `redis-cluster-auth` tool reads if authentication is enabled and an authentication password is configured and operates the FMX Redis Cluster nodes using that authentication password.

If using the `redis-cli`, the authentication password needs to be provided within the command line or tool interactive prompt.

FMX Redis Cluster Authentication Configuration Notes

— FMX application uses Redis as event processing cache, not for sensitive data storage.

— Redis is fast at serving queries. Many passwords per second (~150k/sec) can be tested by an external client. This means that strong password must be used otherwise it is easy to break by brute-force attack.

— Redis do not log authentication failures so there is not an easy way to detect a brute-force attack.

— Redis authentication password is sent in clear text by the clients so with network access it is easy to snoop.

— Redis authentication is not supported by official Redis cluster tools like `redis-trib.rb` for cluster keys redistribution. Even more, some Redis cluster operations (for example migrate slots, used on cluster rebalance) do not support Redis authentication.

6.6.2 FMX RabbitMQ Cluster Security

This section contains a general description and the configuration of the RabbitMQ SSL/TLS feature in FMX, as follows:

— *RabbitMQ SSL/TLS Feature* section describes the background how the TLS works in RabbitMQ 3PP. The purpose is to provide the system administrator a deeper understanding for how the TLS feature is implemented and works in RabbitMQ.

— *FMX RabbitMQ Cluster SSL/TLS Configuration* section describes how the RabbitMQ TLS feature is configured in the FMX SG. This is what the system administrator needs to follow in order to configure the feature.

Note: During RabbitMQ outage the FMX services are not available, all the clients connections are recreated when services are restarted.

6.6.2.1 RabbitMQ SSL/TLS Feature

Using SSL/TLS, the communication channels are encrypted to provide "confidentiality" so that an eavesdropper cannot get and understand data in transit between the peers. This also means AMQP (Advanced Message Queuing Protocol) is secured and messages are not exchanged in clear text.

RabbitMQ has in built support for TLS and SSL/TLS support can be enabled for:

— communication channels

— node communication between nodes in a cluster

— available plugins

The TLS Handshake Protocol is responsible for the authentication and key exchange necessary to establish secure sessions when used on RabbitMQ broker.

The TLS Handshake Protocol involves the following steps:

1. The client sends a `Client hello` message to the server with the client random value and supported cipher suites.
2. The server responds by sending a `Server hello` message to the client with the server random value.
3. The server sends its certificate to the client for authentication and can request a certificate from the client. The server sends the `Server hello done` message.
4. If the server has requested a certificate from the client, the client sends it.
5. The client creates a random premaster secret and encrypts it with the public key from the server certificate, sending the encrypted Pre-Master Secret to the server.
6. The server receives the premaster secret. The server and client each generate the Master Secret and session keys based on the premaster secret.
7. The client sends `Change cipher spec` notification to the server to indicate that the client starts using the new session keys for hashing and encrypting messages. Client also sends `Client finished` message.
8. The server receives `Change cipher spec` and switches its record layer Security state to symmetric encryption using the session keys. The server sends `Server finished` message to the client.
9. The client and the server can now exchange application data over the secured channel they have established. All messages, sent from the client to the server and from the server to the client, are encrypted using session key.

To complete all steps required by the TLS Handshake protocol, certificate chains must be available.

Certificate chains are a key feature of the entire SSL/TLS concept. A root Certificate Authority (CA) issues certificates for sub CAs that, in turn, issue certificates for other sub CAs or end systems such as servers.

RabbitMQ SSL/TLS Server Configuration

To enable the SSL/TLS support in RabbitMQ on Server side, this information is provided:

— Root certificate, server certificate file, and the server key.

— Socket that is going to be used for TLS connections.

— Options to drive the handshake protocol:

  • ask for clients certificates

  • behavior for client certificate validation

All those options are available on RabbitMQ editing the configuration file `rabbitmq.config`.

See snippet from `/etc/rabbitmq/rabbitmq-tls.config`:

```
 %% -*- mode: erlang -*-
[
   {rabbit, [
      {tcp_listeners, []},
      {ssl_listeners, [5671]},
      {ssl_options, [{cacertfile, /path/to/ca/cacert.pem},
                     {certfile, /path/to/server/cert.pem},
                     {keyfile, /path/to/server/key.pem},
                     {versions, ['tlsv1.2', 'tlsv1.1']},
                     {ciphers,[<available ciphers>]},
                     {honor_cipher_order,true},
                     {honor_ecc_order,true},
                     {secure_renegotiate,true},
                     {verify,verify_peer},
                     {fail_if_no_peer_cert,true}]}
 ]},
].
```

RabbitMQ SSL/TLS RabbitMQ Management Plug-in Configuration

The RabbitMQ management plug-in can be configured with SSL/TLS support. In this case, the plug-in is configured providing an HTTPS-based API for managing and monitoring the RabbitMQ server with a browser-based UI and a command line tool `rabbitmqadmin`.

The configuration is available by editing the configuration file `rabbitmq.config` and in general this information is provided:

— Root certificate, server certificate file, and the server key.

— Socket that is going to be used for TLS connections.

See snippet from `/etc/rabbitmq/rabbitmq-tls.config`:

```
{rabbitmq_management, [
     {load_definitions, "/etc/rabbitmq/definitions.json"},
     {listener, [{port, 15672},
                 {ssl, true},
                 {ssl_opts, [{cacertfile, "/path/to/ca/cacert.pem"},
                 {certfile, /path/to/server/cert.pem},
                 {keyfile, /path/to/server/key.pem}]}
                 ]}
  ]},
].
```

RabbitMQ SSL/TLS RabbitMQ Inter Cluster Configuration

Sometimes it is desirable to make the Erlang nodes talk to each other using TLS (SSL), and thus make the whole RabbitMQ cluster communication through TLS.

To configure the RabbitMQ inter cluster communication, concatenate some server certificate and key (`rabbit.pem`) and define some variables used to startup RabbitMQ to support the communication through TLS between cluster nodes.

See snippet from `/etc/rabbitmq/rabbitmq-env.conf`:

```
ERL_SSL_PATH="/usr/lib64/erlang/lib/ssl-8.2.4/ebin"

SERVER_ADDITIONAL_ERL_ARGS="-pa $ERL_SSL_PATH \
-proto_dist inet_tls \
-ssl_dist_opt server_certfile /path/to/rabbit.pem \
-ssl_dist_opt server_secure_renegotiate true client_secure_renegotiate true"


CTL_ERL_ARGS="-pa $ERL_SSL_PATH \
-proto_dist inet_tls \
-ssl_dist_opt server_certfile /path/to/rabbit.pem \
-ssl_dist_opt server_secure_renegotiate true client_secure_renegotiate true"
```

6.6.2.2 FMX Rabbit Cluster SSL/TLS Configuration

File, directories and commands must be executed on FMX VMs. For more information, see the *Connect to a Virtual Machine* section in ENM System Administrator Guide in CPI, Reference [6].

The FMX Rabbit Cluster uses server and client certificates stored into `/etc/rabbitmq/ssl`.

— Certificates are self-signed and the end date is 20 years.

— Certificates are renewed every upgrade, so the end is 20 years from the latest upgrade.

— In case certificates expire, FMX VMs are recreated automatically with new certificate by means of clean-starts.

— FMX certificates can be manually renewed by means of offline, undefine and online FMX VMs.

To check certificates expiration date, execute the following commands:

— Root CA Certificate

```
openssl x509 -enddate -noout -in /etc/rabbitmq/ssl/fmxca/cacert.pem
```

— Server Certificate

```
openssl x509 -enddate -noout -in /etc/rabbitmq/ssl/server/cert.pem
```

— Client Certificate

```
 openssl x509 -enddate -noout -in /etc/rabbitmq/ssl/client/cert.pem
```

See snippet from `/etc/rabbitmq/rabbitmq-notls.config` for unsecure RabbitMQ configuration:

```
%% -*- mode: erlang -*-
[
   {rabbitmq_management,[{load_definitions, "/etc/rabbitmq/definitions.json"}]},
   {rabbitmq_clusterer,[{config, "/etc/rabbitmq/rabbitmq-cluster.config"}]}
].
```

See snippet from `/etc/rabbitmq/rabbitmq-tls.config` for secure RabbitMQ configuration:

```
%% -*- mode: erlang -*-
[
   {rabbit, [
       {tcp_listeners, []},
       {ssl_listeners, [5671]},
       {ssl_options, [{cacertfile,"/etc/rabbitmq/ssl/fmxca/cacert.pem"},
                              {certfile,"/etc/rabbitmq/ssl/server/cert.pem"},
                              {keyfile,"/etc/rabbitmq/ssl/server/key.pem"},
                              {versions, ['tlsv1.2', 'tlsv1.1']},
                              {ciphers,[{dhe_rsa,aes_256_cbc,sha256},
                                        {dhe_dss,aes_256_cbc,sha256},
                                        {rsa,aes_256_cbc,sha256},
                                        {rsa,aes_128_cbc,sha256},
                                        {dhe_rsa,aes_256_cbc,sha},
                                        {dhe_dss,aes_256_cbc,sha},
                                        {rsa,aes_256_cbc,sha},
                                        {dhe_rsa,'3des_ede_cbc',sha},
                                        {dhe_dss,'3des_ede_cbc',sha},
                                        {rsa,'3des_ede_cbc',sha}]},
                             {honor_cipher_order,true},
```

```
                            {honor_ecc_order,true},
                            {secure_renegotiate,true},
                            {verify,verify_peer},
                            {fail_if_no_peer_cert,true}]}
]},
{rabbitmq_management, [
      {load_definitions, "/etc/rabbitmq/definitions.json"},
      {listener, [{port, 15672},
                  {ssl, true},
                  {ssl_opts, [{cacertfile, "/etc/rabbitmq/ssl/fmxca/cacert.pem"},
                  {certfile, "/etc/rabbitmq/ssl/server/cert.pem"},
                  {keyfile, "/etc/rabbitmq/ssl/server/key.pem"
}]}
                      ]}
]},
{rabbitmq_clusterer,[{config, "/etc/rabbitmq/rabbitmq-cluster.config"}]}
].
```

Enable SSL/TLS Configuration on RabbitMQ

The SSL/TLS configuration must be applied once on an FMX Rabbit Cluster node with the `/usr/sbin/rabbitmq-cluster-tls` tool.

Do the following steps:

1. Execute the `/usr/sbin/rabbitmq-cluster-tls` tool on one FMX cluster node to enable the TLS feature.

   For example, if the FMX cluster consists of two nodes,`svc-3-fmx` and `svc-4-fmx`:

   Log on `svc-3-fmx` as root:

```
[root@svc-3-fmx ~]# /usr/sbin/rabbitmq-cluster-tls

Do you want enabling or disabling TLS configuration for RabbitMq Cluster (en →
able/disable):enable

Apply TLS configuration to RabbitMq Cluster nodes? (Y/[N]): Y

[DBUG] [5201] (+0.190) Found PHYSICAL deployment. ENM_ON_CLOUD=

[DBUG] [5201] (+0.249) Not existing SFS lock path: /ericsson/tor/data/fmx/da →
t/fmxrabbit-lock

[INFO] [5201] (+0.379) Raise SFS lock (2018-07-06T22:33:46+0000) on '/ericss →
on/tor/data/fmx/dat/fmxrabbit-lock' with session_id=5201 succeeded

[INFO] [5201] (+0.423) Writing tls status to shared fs...

[INFO] [5201] (+0.576) Wrote tlsstatus to /ericsson/tor/data/fmx/etc/rabbitm →
q/.tlsstatus.

[INFO] [5201] (+0.692) Try to cache tls status file from SFS to local fs...

[INFO] [5201] (+0.902) Booking RabbitMq restart via shared fs...

[INFO] [5201] (+0.959) Booked RabbitMq restart.

[DBUG] [5201] (+0.986) Found PHYSICAL deployment. ENM_ON_CLOUD=

[INFO] [52015201] (+1.104) Release SFS lock (2018-07-06T22:33:46+0000) on '/ →
ericsson/tor/data/fmx/dat/fmxrabbit-lock' with session_id=5201 succeeded
```

```
RabbitMq Cluster TLS configuration changed.
```

RabbitMQ Services restart automatically on all FMX Rabbit Cluster nodes (see `/var/log/messages`).

```
[root@svc-3-fmx /]# cat /var/log/messages | grep "rabbitmq:"
...
… svc-3-fmx rabbitmq: … Stopping RabbitMQ on 10.0.0.21...
… svc-3-fmx rabbitmq: … Attempting to stop daemon /usr/sbin/rabbitmq-server
… svc-3-fmx rabbitmq: … Stopped RabbitMQ
...
… svc-3-fmx rabbitmq: … RabbitMQ stopped on 10.0.0.21...
… svc-3-fmx rabbitmq: … All RabbitMQ instances are stopped, preparing to sta →
rt ...
...
… svc-3-fmx rabbitmq: … Try to cache TLS status file from SFS to local FS...
… svc-3-fmx rabbitmq: … Reading cached TLS status...
… svc-3-fmx rabbitmq: … Updating TLS variable to RabbitMQ configuration file →
 [/etc/rabbitmq/rabbitmq-env.conf]..
… svc-3-fmx rabbitmq: … Updating TLS status [true] to RabbitMQ configuration →
 file [/etc/rabbitmq/rabbitmq-env.conf]..
… svc-3-fmx rabbitmq: … Try to update RabbitMQ configuration link...
… svc-3-fmx rabbitmq: … Attempting to start daemon /usr/sbin/rabbitmq-server
… svc-3-fmx rabbitmq: … Successfully started daemon /usr/sbin/rabbitmq-serve →
r.
...
… svc-3-fmx rabbitmq: … RabbitMQ is currently running
```

The tool stores the new TLS status in the shared file system in the `/ericsson/tor/data/fmx/etc/rabbitmq/.tlsstatus` file:

```
[root@svc-3-fmx /]# cat /ericsson/tor/data/fmx/etc/rabbitmq/.tlsstatus>
true
```

The tool creates a local copy of that file in `/tmp/ericsson/tor/data/fmx/etc/rabbitmq/.tlsstatus` file. In this way, the FMX clients are resilient upon shared file system temporary unavailability.

After service restart, the tools adjust links and system variables to finalize the RabbitMQ configuration:

```
[root@svc-3-fmx /]# ls -la /etc/rabbitmq/rabbitmq.config
```

```
lrwxrwxrwx 1 root root 33 Jun 18 14:22 rabbitmq.config -> /etc/rabbitmq/rabb →
itmq-tls.config

[root@svc-3-fmx /]# cat /etc/rabbitmq/rabbitmq-env.conf | grep "RABBITMQ_TLS →
_ENABLED="


RABBITMQ_TLS_ENABLED=true
```

The details of the TLS change operation are logged to syslog and can be read from `/var/log/messages`.

Disable SSL/TLS Configuration on RabbitMQ

The SSL/TLS configuration must be applied once on an FMX Rabbit Cluster node with the `/usr/sbin/rabbitmq-cluster-tls` tool.

Do the following steps:

1. Execute the `/usr/sbin/rabbitmq-cluster-tls` tool on one FMX cluster node to disable the TLS feature.

   For example, if the FMX cluster consists of two nodes, `svc-3-fmx` and `svc-4-fmx`:

   Log on `svc-3-fmx` as root:

```
[root@svc-3-fmx ~]# /usr/sbin/rabbitmq-cluster-tls

Do you want enabling or disabling TLS configuration for RabbitMq Cluster (en →
able/disable):disable

Apply TLS configuration to RabbitMq Cluster nodes? (Y/[N]): Y

[DBUG] [5201] (+0.190) Found PHYSICAL deployment. ENM_ON_CLOUD=

[DBUG] [5201] (+0.249) Not existing SFS lock path: /ericsson/tor/data/fmx/da →
t/fmxrabbit-lock

[INFO] [5201] (+0.379) Raise SFS lock (2018-07-06T22:33:46+0000) on '/ericss →
on/tor/data/fmx/dat/fmxrabbit-lock' with session_id=5201 succeeded

[INFO] [5201] (+0.423) Writing tls status to shared fs...

[INFO] [5201] (+0.576) Wrote tlsstatus to /ericsson/tor/data/fmx/etc/rabbitm →
q/.tlsstatus.

[INFO] [5201] (+0.692) Try to cache tls status file from SFS to local fs...

[INFO] [5201] (+0.902) Booking RabbitMq restart via shared fs...

[INFO] [5201] (+0.959) Booked RabbitMq restart.

[DBUG] [5201] (+0.986) Found PHYSICAL deployment. ENM_ON_CLOUD=

[INFO] [5201] (+1.104) Release SFS lock (2018-07-06T22:33:46+0000) on '/eric →
sson/tor/data/fmx/dat/fmxrabbit-lock' with session_id=5201 succeeded

RabbitMq Cluster TLS configuration changed.
```

RabbitMQ Services restart automatically on all FMX Rabbit Cluster nodes (see `/var/log/messages`).

```
[root@svc-3-fmx /]# cat /var/log/messages | grep "rabbitmq:"

...

… svc-3-fmx rabbitmq: … Stopping RabbitMQ on 10.0.0.21...

… svc-3-fmx rabbitmq: … Attempting to stop daemon /usr/sbin/rabbitmq-server

… svc-3-fmx rabbitmq: … Stopped RabbitMQ

...

… svc-3-fmx rabbitmq: … RabbitMQ stopped on 10.0.0.21...

… svc-3-fmx rabbitmq: … All RabbitMQ instances are stopped, preparing to sta →
rt ...

...

… svc-3-fmx rabbitmq: … Try to cache TLS status file from SFS to local FS...

… svc-3-fmx rabbitmq: … Reading cached TLS status...

… svc-3-fmx rabbitmq: … Updating TLS variable to RabbitMQ configuration file →
 [/etc/rabbitmq/rabbitmq-env.conf]..

… svc-3-fmx rabbitmq: … Updating TLS status [false] to RabbitMQ configuratio →
n file [/etc/rabbitmq/rabbitmq-env.conf]..

… svc-3-fmx rabbitmq: … Try to update RabbitMQ configuration link...

… svc-3-fmx rabbitmq: … Attempting to start daemon /usr/sbin/rabbitmq-server

… svc-3-fmx rabbitmq: … Successfully started daemon /usr/sbin/rabbitmq-serve →
r.

...

… svc-3-fmx rabbitmq: … RabbitMQ is currently running
```

The tool stores the new TLS status in the shared file system in the `/ericsson/tor/data/fmx/etc/rabbitmq/.tlsstatus` file:

```
[root@svc-3-fmx /]# cat /ericsson/tor/data/fmx/etc/rabbitmq/.tlsstatus>

false
```

The tool creates a local copy of that file in `/tmp/ericsson/tor/data/fmx/etc/rabbitmq/.tlsstatus` file. In this way, the FMX clients are resilient upon shared file system temporary unavailability.

After service restart, the tools adjust links and system variables to finalize the RabbitMQ configuration:

```
[root@svc-3-fmx /]# ls -la /etc/rabbitmq/rabbitmq.config

lrwxrwxrwx 1 root root 33 Jun 18 14:22 rabbitmq.config -> /etc/rabbitmq/rabb →
itmq-notls.config

[root@svc-3-fmx /]# cat /etc/rabbitmq/rabbitmq-env.conf | grep "RABBITMQ_TLS →
_ENABLED="
```

```
RABBITMQ_TLS_ENABLED=false
```

The details of the TLS change operation are logged to syslog and can be read from `/var/log/messages`.

Note: During RabbitMQ outage, the FMX services are not available and all the clients connections are recreated when services are restarted.

6.6.3 Configure Ciphers for SSH

Cipher configuration is supported through a configuration file. The SSH block utility uses the 3PP JSCH for SSH connectivity and as such can only support cipher suites supported by JSCH.

The configuration file is located under:`/ericsson/tor/data/fmx/etc/modules/ssh-block-module/ssh-block-module.properties`

When the utility module is loaded, the loadscript delivered in the utility module moves the file `ssh-block-module.properties` from `/var/opt/ericsson/fmx/modules/ssh-block-module/` to `/ericsson/tor/data/fmx/etc/modules/ssh-block-module`

If a file already exists with the same name, it is renamed with a `.old` suffix (for example the name `ssh-block-module.properties` is renamed `ssh-block-module.properties.old`). The `.old` file is automatically overwritten without prompting for input for multiple load or unload sequences.

Upgrade Scenarios

`Unchanged ssh-block-module.properties`

No action is needed as the new cipher suites will be available and used.

`Changed ssh-block-module.properties`

Depending on use case either:

1. Set to false the parameter `automaticallyUpdateThisFileOnLoad`, if any property from `ssh-block-module.properties` is changed, to have the change remained even after the restart of FMX VM. The changes made in `ssh-block-module.properties` are reloaded to defaults when the property `automaticallyUpdateThisFileOnLoad` is set to true. This property must be set to false to retain the changes even after the upgrade of the FMX instance.
2. Audit, compare, and move cipher configuration to the new `ssh-block-module.properties` file.
3. Rename the file `ssh-block-module.properties.old` in `ssh-block-module.properties`. This can be useful when `ssh-block-module.properties` has already gone through strict auditing.

Configuration Reload Mechanism

No-restart configuration changes is supported through a configuration reload mechanism. The file is polled every 15 seconds for changes.

Configuration File Key Mapping

Entry	Short	Explanation
`kex`	Key Exchange	Determines what algorithms must be used for the key exchange between the client and server.
`cipher.s2c`	Cipher Server to Client	Cipher suites supported for Server to Client communication.
`cipher.c2s`	Cipher Client to Server	Cipher suites supported for Client to Server communication.
`CheckCiphers`	Check Ciphers	A list of Ciphers which should be first checked for availability. All ciphers in this list which are not working will be removed from the ciphers.c2s and ciphers.s2c before sending these lists to the server in a KEX_INIT message.
`mac.c2s`	Message Authentication Code Client to Server	Message authentication code algorithms for client-to-server transport.
`mac.s2c`	Message Authentication Code Server to Client	Message authentication code algorithms for server-to-client transport.

7 Network Surveillance eXpert (NSX) Administration Tasks

> This section describes the system administration tasks for Network Surveillance eXpert (NSX) rule modules, including LTE Surveillance eXpert (LTX) and ECM Rule Expert Package (ECX).

The NSX consists of a number of Automatic Alarm Handling (FMX) rule modules designed to reduce the number of presented alarms in ENM alarm list and to enhance the alarm presentation.

Requirements

- You must have root authority to perform some of the tasks described in this chapter.
- You must be familiar with the following:
  - Fault Management
  - Automatic Alarm Handling (FMX)

7.1 Network Surveillance eXpert (NSX)

The purpose of the NSX Rule Modules Package is to assist operators with the management of alarms.

The package focus is on providing customers with generic configurable rules for custom basic alarm filtering and correlation.

This Rule Module requires the following FMX Utility Modules to be loaded:

- baseblocks-module
- enm-blocks-module
- exclusive-region-module
- event-used-in-other-module

7.1.1 Configure Alarm Monitor (FM) for NSX

FMX Rules mainly communicate with the operator through the Alarm Monitor (FM). You can place longer pieces of information, such as complete printouts, after the actual Problem Text and you can then view them by expanding the alarm.

Abbreviated information can, as an alternative, be displayed in dedicated columns in the Alarm Monitor (FM). You can do this by letting the FMX store this information in an Additional Attribute. Since the number of columns that you can view simultaneously is limited, it is strongly recommended that you develop a strategy, or scheme, in which to place such information.

The Maintenance Attribute

The NSX package contains functionality for filtering out alarms from nodes in Maintenance mode.

You can perform this filtering in the following ways:

— Hide the alarms in question from view in both the ENM and NMS systems.

— Hide the alarms in question from view in the NMS system, but present them in the Alarm Monitor (FM) in ENM.

— Do not hide the alarms, but just mark them.

7.1.1.1 Additional attributes in NSX

Additional Attribute	Description
`FMX_Status`	A three-character code explaining the state of the rule execution. The most commonly used codes are the following: — Rdy - Ready; FMX processing has finished. — ERR - Error; a script or MML session failed. — FRQ - Frequent alarm. — MLT - Multiple; a group alarm that is FMX-created. — Inv - Investigating; FMX Processing that might take a longer time is ongoing.
`FMX_Object`	A shorter character string containing the main object related to the alarm - a cell, for instance.
`FMX_Additional`	A longer character string containing more information such as related objects, a summary of the fault information, or anything that makes sense for the rule in question.
`Maintenance`	A string that indicates that the alarm comes from a node that is under maintenance. The possible values are: MARK, NMS, NMSandOSS, FALSE, or it is not set. — MARK indicates that the alarm comes from a node that is under maintenance. — NMS is the same as MARK, but indicates that the alarms are not forwarded toward the NMS. — NMSandOSS indicates that these alarms are hidden in ENM and are not forwarded to the NMS. (You cannot see this value in the Alarm Monitor (FM), since these alarms are not visible; it is listed here for completeness.) — FALSE indicates that the node is back from Maintenance state.

Additional Attribute	Description
	Alarms that come from nodes that are not and have not been in Maintenance state do not have any values set. One purpose of this Additional Attribute is to allow customized filtering in Alarm Lists.

It is not necessary for all the rules to use all the fields. See the Alarm Monitor (FM) online Help in ENM for information about making these columns visible in the Alarm Monitor (FM).

7.1.2 Configure NSX Module Parameters

Description

Define the parameters that control rule behavior in the FMX for each module and sometimes for each module and rule. The default values for the different parameters are set when the module loads. You can change the values in Module Management by selecting a module and choosing Module Parameters. A pane opens in which you can change the values in the Current Value column. For more information, see the *Automatic Alarm Handling (FMX)* online help.

You can set the configuration parameter values for each trigger event in the NSX modules by means of configuration text files, as described in specific module's configuration chapter.

7.1.2.1 Module Parameters for NSX

Module Parameter	Description
`Clear_Time`	Controls the time (in minutes) before the alarm is cleared. The default value is 30. Used in the NSX_Event_to_Alarm module. For example, if the Clear_Time parameter is set to five and the NSX_Event_to_Alarm module is activated, the alarm generated by the module is cleared after five minutes if an event specified in the configuration text file triggers the module. This module parameter can be overwritten by clear time values for each trigger event in the configuration text file.
`DefaultSeverity`	Controls the severity set on alarms generated from error messages. The default value is Critical. Used in the NSX_Event_to_Alarm module. This module parameter can be overwritten by severity values for each trigger event in the configuration text file.
`Filter_Time`	Controls how many minutes to wait before checking whether an alarm is still active. The default value is 3. Used in the NSX_Shortlived_and_Frequent module. This results in a delay on alarm presentation, but it is necessary to recognize short-lived alarms while keeping the ALV clean. Decimal values are allowed. For example, use 2.5 for two and a half minutes.
`Frequent_Interval`	Controls how many minutes back in time the frequent check is made. The default value is 60. Used in the NSX_Shortlived_and_Frequent module. For example, assume that Frequent_Limit is set to 3 and Frequent_Interval is set to 30, and no FREQUENT alarm is active, but two alarms that the FMX is counting have occurred 10 and 20 minutes ago. If one more alarm arrives within 10 minutes, the threshold Frequent_Limit is reached and a FREQUENT alarm is issued. However, if it takes 15 minutes instead for the next alarm to arrive, the threshold is not reached. Then, the threshold counter

Module Parameter	Description
	remains at two, and there are five minutes left for the frequent limit to be reached. The Frequent_Interval parameter is also used for another purpose: The FMX automatically clears the FREQUENT alarm when no new original alarms occur during the Frequent_Interval.
`Frequent_Limit`	Controls how many instances of the alarm need to occur before a FREQUENT alarm is issued. For each situation, the counter is based on the Object of Reference (FDN) of the alarm. The default value is 5. Used in the NSX_Shortlived_and_Frequent module.
`Frequent_Poll_Time`	For Frequent alarms, FMX continuously updates the FMX_Additional attribute with the current number of alarms issued during the Frequent_Interval time period. The counter is adjusted on a polling basis. The time between each update is defined by the Frequent_Poll_Time. The recommended values of Frequent_Poll_Time are between a quarter of- to the whole of the Frequent_Interval. The default value is 10. Used in the NSX_Shortlived_and_Frequent module.
`Hide_During_Frequent`	`TRUE` or `FALSE`: defines whether the original triggering alarms are hidden during a FREQUENT alarm situation. The default value is FALSE. Used in the NSX_Shortlived_and_Frequent module.

7.1.3 Configure Event Selection for the NSX_Event_to_Alarm Module

The triggering events and the configuration for these events, which are handled by the NSX_Event_to_Alarm module, are stored in the `/ericsson/tor/data/fmx/etc/modules/nsx/EventToAlarm_events.txt` file. The file defines which error messages need to be handled by the module, the severity level of the generated alarm, and the time in minutes after which the alarm is cleared.

Steps

Edit `EventToAlarm_events.txt` with any text editor.

The file is located in the `/ericsson/tor/data/fmx/etc/modules/nsx/` directory.

Note: Any configuration change is detected automatically and loaded into the rule.

7.1.3.1 EventToAlarm_events.txt

The file contains one row for each unique system type-specific problem pair. Each row consists of the following four parameters:

`<SourceType>; <SpecificProblem>; <PerceivedSeverity>; <ClearTime>`

Parameter	Description
`SourceType`	The SourceType of the error message. For example, `ERBS` for an eNodeB or `RNC` for an RNC-node. This parameter is mandatory.
`SpecificProblem`	This parameter is optional. If specified, only the specific problem is considered. If not specified, all the error messages are converted to Alarms.
`PerceivedSeverity`	This parameter is optional. The severity of the generated alarm. Default value `Critical` is used if it is not specified.

Parameter	Description
`ClearTime`	This parameter is optional. The time (in minutes) after which the newly created alarm is cleared. Default value `30 minutes` is used if it is not specified.

Example 1

```
# This is the list of error messages that should be handled by the FMX module NS
X_Event_to_Alarm.
# The syntax for each entry is:
# <SourceType>;<SpecificProblem>;<PerceivedSeverity>;<ClearTime>
# where
# SourceType is the SourceType of the incoming event
# SpecificProblem is the event specification
# PerceivedSeverity will be the severity of the generated alarm.
# ClearTime is the time (in minutes) after the newly created alarm gets cleared.
#
# For example:
# RBS;NodeBFunction_NodeRestarted;warning;60
# would convert every NodeBFunction_NodeRestarted event with RBS SourceType (CPP
 based nodeB) to an alarm
# with the same SpecificProblem and 'warning' severity.
# The alarm will be cleared after 60 minutes.
#
#
RBS;NodeBFunction_NodeRestart;warning;5
RBS;TxDeviceGroup_SuccessfulRecoveryActionPerformedBoardRestart;minor;3
RBS;TxDeviceGroup_TemperatureExceptionallyHigh;minor;3
ERBS;TxDeviceGroup_TemperatureExceptionallyHigh;warning;1
```

7.1.4 Configure Event Selection for the NSX_Hard_Filter Module

The configuration that defines how the specific alarms need to be handled by the NSX_Hard_filter module, is stored in the file:

`/ericsson/tor/data/fmx/etc/modules/nsx/HardFilter.txt`

The file defines which source types and specific problems need to be handled by the module.

When a Module is activated, open alarms present in database that match with the rule are acted.

Steps

Edit the `/ericsson/tor/data/fmx/etc/modules/nsx/HardFilter.txt` with any text editor.

Note: Any configuration change, applied by saving the file, is automatically detected, loaded into the rule, and updated in FM without impacting the already active configuration. The best practice is to save the `HardFilter.txt` file just once to avoid multiple subscription updates because each update causes a synchronization of alarms with the database.

7.1.4.1 HardFilter.txt

The file contains one row for each specific problem. Each row consists of the following two parameters:

`<SourceType>; <SpecificProblem>`

Parameter	Description
`SourceType`	The SourceType of the alarm or error message. For example, ERBS for an eNodeB or RNC for an RNC node. This parameter is mandatory .Multiple SourceTypes can be given in the file by separating them with commas.
`SpecificProblem`	The specific problem of the selected alarm or error message.This parameter is optional. When the SpecificProblem for a specific SourceType is not mentioned, every alarm on the SourceType are filtered.

Example 2

```
# This is the list of alarms that should be filtered by the FMX module NSX_Hard_ →
filter
# The syntax for each entry is:
# <SourceType>;<SpecificProblem>
# where
# SourceType is the System Type of the alarm (mandatory)
# SpecificProblem is the alarm specification
#
# For example:
# SGSN-MME;pmLogReady
# would filter out all pmLogReady alarms from SGSN-MME nodes.
# alarm is stored in history database although it is not visible in Alarm Monito →
r
#
ERBS,RadioNode;NssSynchronization_SystemClockStatusChanged
ERBS,RadioNode;Nss Synchronization System Clock Status Change
ERBS,RadioNode;NSS Synchronization System Clock Status Change
ERBS,RadioNode;System Clock in Holdover Mode
ERBS,RadioNode;E1Ttp_NOCRC4MFA
ERBS,RadioNode;Extensive Ethernet frame flooding
ERBS,RadioNode;Ip over ATM Link Supervision Failed
ERBS,RadioNode;Ip over ATM Link Supervision Recovered
ERBS,RadioNode;NssSynchronization_SynchRefChanged
ERBS,RadioNode;SYSTEM_CLOCK_IN_HOLDOVER_MODE

RNC,RBS;AscDeviceGroup_SuccessfulRecoveryActionPerformedBoardRestart
RNC,RBS;E1Ttp_NOCRC4MFA
RNC,RBS;Extensive Ethernet frame flooding
RNC,RBS;Ip over ATM Link Supervision Failed
RNC,RBS;Ip over ATM Link Supervision Recovered
```

7.1.5 Configure Event Selection for the NSX_Shortlived_and_Frequent Module

The configuration that defines how specific alarms are handled by the module dealing with short-lived and frequent alarms is stored in the `/ericsson/tor/data/fmx/etc/modules/nsx/ShortLivedEvents.txt` file. The file serves two main purposes:

— It defines which alarms need to be handled by the module.

— It sets the parameter values for each defined alarm. Error messages do not display any short-lived characteristics and are consequently of no interest.

Steps

Edit the `ShortLivedEvents.txt` with any text editor.

The file is located in the `/ericsson/tor/data/fmx/etc/modules/nsx/` directory.

Note: Any configuration change is detected automatically and loaded into the rule.

7.1.5.1 ShortLivedEvents.txt

The file contains one row for each specific problem. Each row consists of the following nine parameters:

`<SystemType>; <SpecificProblem>; <FilterTime>; <FrequentLimit>; <FrequentInterval>; <FrequentPollTime>; <HideDuringFrequent>; <FDNPart>; <AxeMO>`

Parameter	Description
`SystemType`	The system type of the alarm. For example, LRAN for an eNodeB, or WPP for an MME node.
`SpecificProblem`	The specific problem of the selected alarm.
`FilterTime`	The limit (in minutes) for an alarm to be regarded as short-lived. Decimal fractions are allowed.
`FrequentLimit`	How many instances of the alarm need to occur before a FREQUENT alarm is issued.
`FrequentInterval`	The interval (in minutes) during which the frequent limit is checked. Decimal fractions are allowed.
`FrequentPollTime`	The time (in minutes) between frequency updates or checks to clear FREQUENT alarms. Decimal fractions are allowed.
`HideDuringFrequent`	Whether the original alarm(s) need to be hidden when a FREQUENT alarm is active (TRUE or FALSE).
`FDNPart`	The part of the ObjectOfReferences fully distinguished name which must be presented as the FMX_Object. If not specified, the FMX_Object is attached, by default, to the part after the last comma in the ObjectOfReference.
`AxeMO`	An attribute which is needed to parse the additional alarming MO from the alarm printout in the case of AXE-based alarms. This parameter is only required for AXE alarms.

Example 3

```
# This is the list of alarms that should be filtered by the FMX module NSX_Short →
lived_and_Frequent
# The syntax for each entry is:
# <SystemType>;<SpecificProblem>;<FilterTime>;<FrequentLimit>;<FrequentInterval> →
;<FrequentPollTime>;<HideDuringFrequent>;<FDNPart>;<AxeMO>
# where
# SystemType is the System Type of the alarm
# SpecificProblem is the alarm specification
# FilterTime is the limit (in minutes) for an alarm to be regarded as short-live →
d.
# FrequentLimit is how many instances of the alarm that need to occur before a F →
REQUENT alarm is issued.
# FrequentInterval is the interval (in minutes) during which the frequent limit →
is checked.
# FrequentPollTime is the time (in minutes) between frequency updates or checks →
for clearing of FREQUENT alarms.
# HideDuringFrequent decides if the original alarm should be hidden when FREQUEN →
```

```
T alarms are shown (TRUE or FALSE)
# FDNPart decides which part of the ObjectOfReference Fully Distinguished Name t →
hat should be presented as FMX_Object.
# If not specified, it defaults to the part after the last comma in the ObjectOf →
Reference.
# AxeMO is needed to parse the additional alarming MO from the alarm printout in →
 case of AXE-based alarms.
# Need to be specified only in case of AXE alarms.
#
# For example:
# LRAN;LinkFailure;3;5;60;10;TRUE;RiPort
# would filter out LinkFailure alarms from LRAN nodes with an alarm life-time of →
 less that 3 minutes.
# If 5 such alarms arrive from the same ObjectOfReference within 60 minutes, FMX →
 will create a FREQUENT LinkFailure alarm.
# The original LinkFailure alarms are hidden. The FREQUENT LinkFailure alarm is →
updated with frequency value each 10 minutes.
# The RiPort of the ObjectOfReference name is presented as the FMX_Object in the →
 Alarm List.
#
# Example for AXE based alarms:
# AXE;Digital Path Fault Supervision;1;3;5;0.5;FALSE;ManagedElement;DIP
# DIP information will be parsed from the alarm printout as additional alarm MO →
information. This parameter will be presented as the FMX_Object in the Alarm Lis →
t.
#
####Be aware that LRAN;ServiceUnavailable; har a module of its own already
LRAN;LinkFailure;2;5;60;10;TRUE;RiPort;
LRAN;TU Synch Reference Loss of Signal;1;5;60;10;TRUE;MeContext;
LRAN;Resource Configuration Failure;2;5;60;10;TRUE;
LRAN;ResourceConfigurationFailure;2;5;60;10;TRUE;
LRAN;NumberOfHwEntitiesMismatch;3;5;60;10;TRUE;
LRAN;CircuitBreakerTripped;1;5;60;10;TRUE;HwUnit
LRAN;Gigabit Ethernet Link Fault;1;5;60;10;TRUE;
LRAN;NoContact;1;5;60;10;TRUE;
LRAN;SystemOvervoltage;1;5;60;10;TRUE;
```

7.1.6 Configure Event Selection for the NSX_Maintenance_filter Module

The NSX Rule package is equipped with support for placing selected nodes into Maintenance Mode between a specified start and end date/time value..The effect is that alarms (existing alarms and new alarms) for these nodes are managed according to the select mode (see Syntax on page 115) for the scheduled duration."

You can define more (even overlapping) maintenance windows for the same node with different filtering settings. You can set filtering levels through the `mode` parameter. If a node has overlapping maintenance periods, the highest-priority mode is applied during the overlapping part (MARK = highest, NMSandOSS = lowest), see Additional attributes in NSX on page 108. The module updates the alarm state of the nodes each minute. After a maintenance window is closed, the alarms which are still active are displayed (if they were filtered previously). Alarms from nodes that are not in maintenance mode and all clear alarms are handled as usual.

Steps

To add, remove, and list nodes in the schedule, execute the

```
/ericsson/tor/data/fmx/etc/modules/nsx/ManageNodesInMaintenance.pl
```

file.

Note: Any configuration change is detected automatically.

7.1.6.1 Syntax

Options	Description
`-node`	The name of the `<node>` for the action. The script verifies that the node is defined in ENM before adding it to the list. Short names (for example, eNodeB32) are accepted, but it is recommended that you use a fully distinguished name to avoid confusion.
`-Start`	The date and time for the start of the maintenance period in the format of yyyymmddHHMM, with or without separators. Separators - (hyphen) / (dash) <space> . (full stop) : (colon) _ (underscore) are allowed. Make sure to quote the value if spaces are included. Seconds are truncated to zero. Year values beyond 2038 are not supported.
`-End`	The date and time for the end of the maintenance period in the format of yyyymmddHHMM, with or without separators. Separators - (hyphen) / (dash) <space> . (full stop) : (colon) _ (underscore) are allowed. Make sure to quote the value if spaces are included. Seconds are truncated to zero. Year values beyond 2038 are not supported.
`-l[ist]`	List all entries in the maintenance list. Specify `-n[ode] <node>` for a subsection of entries for only this node. This is useful as input for copy/paste when adding or removing entries.
`-a[dd]`	Add one specific entry in the maintenance list. Overlapping time periods for the same node are supported, see the `-mode` option. Only valid with -n[ode] <node> -Start <yyyymmddHHMM> -End <yyyymmddHHMM>.
`-r[em]`	Remove one specific entry in the maintenance list. Only valid with `-n[ode] <node> -Start <yyyymmddHHMM> -End <yyyymmddHHMM>`.
`-delete`	Delete all entries of a node in the maintenance list. If you use it together with the `-days` parameter, only entries which expired earlier than the specified days are deleted. Only valid with `-n[ode] <node>`.
`-days`	Optional parameter. You can use it together with the `-delete` option - in this case, only entries which expired earlier than the specified days are deleted. Only positive number values and `0` are accepted. Value `0` means 'expired now or earlier'.
`-c[lean]`	Clean out all entries that have passed (for example, end-value is older than 'now'). It is recommended that you run this at regular intervals to prevent the list from becoming clogged with entries that have passed.
`-o[verride]`	Skip topology verification that the node is defined in ENM. This can be necessary when new entries are added before the nodes are defined in ENM. Only valid with the `-a[dd]` option.
`-m[ode]`	Set the way maintenance is handled by the FMX module. The available modes are as follows (from higher priority to the lower): — `-mode=MARK` - alarms from the node are marked as maintenance alarm, displayed in ENM and forwarded to NMS. — `-mode=NMS` - alarms are marked as maintenance alarm, displayed in ENM but not forwarded to NMS. — `-mode=NMSandOSS` - alarms are marked as maintenance alarm, are hidden in ENM, and not forwarded to NMS. The default option is `MARK` (used if no option or invalid option is given). If you define overlapping time periods with different modes, the highest priority mode is applied.

Options	Description
	Only valid with the `-a[dd]` option.
`-h[elp]`	Print the help page.

7.1.7 NSX Statistics

The Module Statistics enables you to understand the rules utilization and value. A counter is updated each time a trigger or exit has been passed.

You can access the Rule Statistics in different ways. You can view them as a table in the Module Management UI, within Module Statistics. You can export that table to a csv file. You can also fetch the statistics with the `fmxcli` command within the FMX nodes.

For more information about module statistics, refer to the *Automatic Alarm Handling (FMX)* online help.

7.1.7.1 Block Types Used by Statistics

The Block Type column lists what kind of block the counter is on. The block types are as follows:

Block Type	Description
END-RULE	The counter on the end block. The Block Name column in the line of the END-RULE describes what FMX has done with the event.
ENM-EVENT-TRIGGER	The counter for all new events that have arrived.
EVENT-CLEARING-TRIGGER	The counter for all clearings of events that have arrived.
MODULE-ACTIVATION-TRIGGER	The trigger issued when a module is activated or deactivated.
UNIX-TRIGGER	The counter for events that are internally triggered, usually by a script being executed.
UPDATE-TRIGGER	Some events can come to the ENM as an updated alarm.

7.1.7.2 Events Used by NSX End Rule Blocks in Statistics Output

The Block Name for END-RULEs are named according to the following naming convention:

`<FMX_action_on_the_event>.<Rule_Name>.<Extra_information>`

Example 4

`FILTERED.EXTERNAL_ALARM.short-lived`

The possible values of `<FMX_action_on_the_event>` are listed in the table below:

FMX_action_on_the_event	Description
CLEARED	Used in rules where a clearing is used as a trigger. Usually not relevant in statistics.

FMX_action_on_the_event	Description
FILTERED	The alarm is not shown in the ALV, because it is cleared during the short-filter time.
FREQUENT	Used in rules triggered by FMX-created FREQUENT alarms.
INFO	The alarm is shown in the ALV with additional information. The additional information can be, for example, the result of an MML command.
MULTIPLE	Used in rules triggered by FMX-created multiple alarms.
SOLVED	Used by the DIP_QUALITY_SUP rule when the alarm is reset by the `DTQSR` command, and in RADIO_TRANS_IP_GB_IF_FAULT rule when the clearing is handled.
UNIX-ERR	A Unix command has failed to execute. The name of the script is displayed in the `<Extra_information>` part of the END-RULE name. This can be a temporary or a persistent problem and it is recommended that you investigate its cause.

7.2 LTE Surveillance eXpert (LTX)

The purpose of the LTX Rule Modules Package is to assist operators with the management of alarms.

The LTX package focus is on filtering and correlation of most common LTE network alarms.

Requirements

- This Rule Module requires the following FMX Utility Modules to be loaded:
  - baseblocks-module
  - exclusive-region-module
  - enm-blocks-module
- All NSX Rule Modules are loaded.

7.2.1 LTX Module Parameters

You can control the LTX Rule Module behaviour on runtime with the Module Parameters.

The Rule Module uses its predefined default parameter values when it is first loaded in the system. You can changes the values anytime from Module Management UI by selecting the module and choosing Module Parameters. A pane opens in which you can change the values in the Current Value column. For more information, see the *Automatic Alarm Handling (FMX)* online help.

7.2.2 LTX Module Parameters

The following parameters are used in the LTX modules (other parameters are available, but they are no longer in use):

Module Parameter	Description
`Filter_Time`	Controls how many minutes to wait before checking whether an alarm is still active. The default value is 2. This results in a delay on alarm presentation, but it is necessary to recognize short-lived alarms while keeping the ALV clean. Decimal values are allowed. For example, use 2.5 for two and a half minutes.
`Frequent_Interval`	Controls how many minutes back in time the frequent check is made. The default value is 60. For example, assume that Frequent_Limit is set to 3 and Frequent_Interval is set to 30 (minutes), no FREQUENT alarm is active, but two alarms that the FMX is counting have occurred 10 and 20 minutes ago. If one more alarm arrives within 10 minutes, the threshold Frequent_Limit is reached and a FREQUENT alarm is issued. However, if it takes 15 minutes instead for the next alarm to arrive, the threshold is not reached. Then, the threshold counter remains at two, and there are five minutes left for the frequent limit to be reached. The Frequent_Interval parameter is also used for another purpose: The FMX automatically clears the FREQUENT alarm when no new original alarms occur during the Frequent_Interval.
`Frequent_Limit`	Controls how many instances of the alarm need to occur before a FREQUENT alarm is issued. For each situation, the counter is based on the Object of Reference (FDN) of the alarm. The default value is 5.
`Frequent_Poll_Time`	For Frequent alarms, FMX continuously updates the FMX_Additional attribute with the current number of alarms issued during the Frequent_Interval time period. The counter is adjusted on a polling basis. The time between each update is defined by the Frequent_Poll_Time. The recommended values of Frequent_Poll_Time are between a quarter of- to the whole of the Frequent_Interval. The default value is 5.
`Hide_During_Frequent`	TRUE of FALSE: defines whether the original triggering alarm(s) are hidden during a FREQUENT alarm situation or not. The default value is FALSE.

7.2.3 LTX Statistics

The Module Statistics enables you to understand the rules utilization and value. An counter is updated each time a trigger or exit has been passed.

You can access the Rule Statistics in different ways. You can view them as a table in the Module Management UI, within Module Statistics. You can export that table to a csv file. You can also fetch the statistics with the `fmxcli` command within the FMX nodes.

For more information about module statistics, refer to the *Automatic Alarm Handling (FMX)* online help.

Block Types Used by Statistics on page 116

7.2.3.1 Events Used by LTX End Rule Blocks in Statistics Output

The Block Name for END-RULEs are named according to the following naming convention:

`<FMX_action_on_the_event>.<Rule_Name>.<Extra_information>`

Example 5

```
FILTERED.EXTERNAL_ALARM.short-lived
```

The possible values of <FMX_action_on_the_event> are listed in the table below:

FMX_action_on_the_event	Description
CLEARED	Used in rules where a clearing is used as a trigger. Usually not relevant in statistics.
FILTERED	The alarm is not shown in the ALV, because it is cleared during the short-filter time.
FREQUENT	Used in rules triggered by FMX-created FREQUENT alarms.
INFO	The alarm is shown in the ALV with additional information. The additional information can be, for example, the result of an MML command.
MULTIPLE	Used in rules triggered by FMX-created multiple alarms.
SOLVED	Used by the DIP_QUALITY_SUP rule when the alarm is reset by the DTQSR command, and in RADIO_TRANS_IP_GB_IF_FAULT rule when the clearing is handled.
UNIX-ERR	A Unix command has failed to execute. The name of the script is displayed in the <Extra_information> part of the END-RULE name. This can be a temporary or a persistent problem and it is recommended that you investigate its cause.

7.3 WRAN Surveillance Expert (WRX)

The purpose of the WRAN Rule Modules Package is to assist operators with the management of alarms.

The WRAN package focus is on filtering and correlation of most common WCDMA network alarms.

Requirements

— This Rule Module requires the following FMX Utility Modules to be loaded:
  — baseblocks-module
  — exclusive-region-module
  — enm-blocks-module
— All NSX Rule Modules are loaded.

7.3.1 WRX Module Parameters

You can control the WRX Rule Module behaviour on runtime with the Module Parameters.

The Rule Module uses its predefined default parameter values when it is first loaded in the system. You can changes the values anytime from Module Management UI by selecting the module and choosing Module Parameters. A

pane opens in which you can change the values in the Current Value column. For more information, see the *Automatic Alarm Handling (FMX)* online help.

The following parameters are used in the WRX modules:

`<Alarm>.Filter_Time`	Controls how many minutes to wait before checking whether an alarm is still active. The default value is 2. This results in a delay on alarm presentation, but it is necessary to recognize short-lived alarms while keeping the ALV clean. Decimal values are allowed. For example, use 2.5 for two and a half minutes.
`<Alarm>.Frequent_Interval`	Controls how many minutes back in time the frequent check is made. The default value is 60. Example: Assume that Frequent_Limit is set to 3 and Frequent_Interval is set to 30 (minutes), no FREQUENT alarm is active, but two alarms that the FMX is counting have occurred 10 and 20 minutes ago. If one more alarm arrives within 10 minutes, the threshold Frequent_Limit is reached and a FREQUENT alarm is issued. However, if it takes 15 minutes instead for the next alarm to arrive, the threshold is not reached. Then, the threshold counter remains at two, and there are five minutes left for the frequent limit to be reached. The Frequent_Interval parameter is also used for another purpose: The FMX automatically clears the FREQUENT alarm when no new original alarms occur during the Frequent_Interval.
`<Alarm>.Frequent_Limit`	Controls how many instances of the alarm need to occur before a FREQUENT alarm is issued. For each situation, the counter is based on the Object of Reference (FDN) of the alarm. The default value is 5.
`<Alarm>.Frequent_Poll_Time`	For Frequent alarms, FMX continuously updates the FMX_Additional attribute with the current number of alarms issued during the Frequent_Interval time period. The counter is adjusted on a polling basis. The time between each update is defined by the Frequent_Poll_Time. The recommended values of Frequent_Poll_Time are between a quarter of- to the whole of the Frequent_Interval. The default value is 5.
`<Alarm>.Hide_During_Frequent`	TRUE of FALSE: defines whether the original triggering alarm(s) are hidden during a FREQUENT alarm situation or not. The default value is FALSE.
`Topology.valid_time`	How long relations between RBS's and UtranCell's and IubLinks are treated as not changed. Topology data is fetched when needed but cached for the time set. Time set in hours. The default value is 24.

7.3.2 Configure AlmDevice_ExternalAlarms presentation by WRX_Utran_External_Alarm module

The configuration of how the WRAN external alarms with SpecificProblem AlmDevice_ExternalAlarm are handled are defined in the file `/ericsson/tor/data/fmx/etc/modules/wrx/WranExternalAlarms.txt`. The file defines which severity different external alarms should have and for how long the time filter shall be.

Steps

Edit `WranExternalAlarms.txt` with any text editor.

The file is located in the `/ericsson/tor/data/fmx/etc/modules/wrx/` directory.

Note: Any configuration change is detected automatically and loaded into the rule.

7.3.2.1 WranExternalAlarms.txt

The file contains one row for each unique external alarm text. Each row consists of the following four parameters:

`<FilterTime>; <Severity>; <FMX Alarm Severity>; <External Alarm Text>`

`FilterTime`	The short live filter time in minutes, If set to X it means alarm is never presented - hard filtered. `0` means immidiately presented.
`Severity`	The specific problem of the selected error message.
`FMX Alarm Severity`	The severity that FREQUENT alarm should have.
`External Alarm Text`	The unique string in the Problem text of the alarm.

Example 6

```
2;Minor;Major;Door Open
4;;;High Temperature
4;;;Door Unlocked
1;;;Mains Alarm
1;Major;Major;Battery Low
4;;;Air Conditioner Failure
1;Major;Major;Batt. Voltage Alarm
0;;;Aircraft Light Faulty
```

7.4 ECM Rule Expert Package (ECX)

ECX is a collection of FMX rules grouped into ECX_VNF_Alarm_Correlation FMX Rule module and designed for correlation of alarms and error messages raised by a VNFM target against associated VNF NetworkElements. The purpose of the module is to raise on VNF the same alarm or error message raised on VNFM if the alarm or the error impacts the VNF (that means the vApp resource used by VNF) and to propagate clear and acknowledge from VNFM alarm or error message to the correlated VNF alarm or error message.

Rule Module	Action	Explanation
ECX_VNF_Alarm_Correlation	Correlation of the VNFM alarms and errors messages (type ECM) with associated VNF	The expected result is that raise, clear and acknowledge of the alarms or error messages on VNFM (type ECM) are propagated to interested VNF and vice-versa.

When a vApp alarm or error message is raised on VNFM and there is an associated VNF using the alarmed vApp resource, then the same alarm or error message is raised on VNF: additional information are added to both VNFM and VNF alarms or error messages.

Additional Information on VNFM alarm or error message are:

— affectedFDN containing VNF FDN

Additional Information on VNF alarm or error message are:

— originalFDN containing VNFM FDN

— originalAlarmId containing VNFM correlated generated alarm/error message identifier

— vAppId

— tenant name

When an alarm or error message is acknowledged on VNFM and a correlated VNF alarm or error message is present, then also correlated VNF alarm or error message is acknowledged.

When an alarm is cleared on VNFM and a correlated VNF alarm is present, then also correlated VNF alarm is cleared.

Requirements

— ECX requires the following FMX Utility Modules to be loaded:

  • baseblocks-module

  • enm-blocks-module

  • enmcli-blocks-module

— Rule module ECX_VNF_Alarm_Correlation must be loaded in FMX.

— ECX_VNF_Alarm_Correlation must be activated on the whole Network or on the VNFM (type ECM) and all the subtended VNFs

— VNFM (type ECM) configured on ENM.

— VNF configured on ENM and associated to VNFM.

7.4.1 ECX Module Parameters

You can control the ECX_VNF_Alarm_Correlation Rule Module behaviour on runtime with the Module Parameters.

The Rule Module uses its predefined default parameter values when it is first loaded in the system. You can changes the values anytime from Module Management UI by selecting the module and choosing Module Parameters. A pane opens in which you can change the values in the Current Value column. For more information, see the *Automatic Alarm Handling (FMX)* online help.

Module Parameter	Description
`suppress_original_alarm`	This parameter allows to enable or disable the suppression of original VNFM alarms/error messages correlated against a VNF: if the parameter is set to FALSE, the VNFM original alarms/error

Module Parameter	Description
	messages are shown on FM NBI, otherwise they are hidden. The default value is FALSE.

7.4.2 ECX Statistics

The Module Statistics enables you to understand the rules utilization and value. An counter is updated each time a trigger or exit is passed.

You can access the Rule Statistics in different ways. You can view them as a table in the Module Management UI, within Module Statistics. You can export that table to a csv file. You can also fetch the statistics with the `fmxcli` command within the FMX nodes.

For more information about module statistics, refer to the *Automatic Alarm Handling (FMX)* online help.

7.4.2.1 ECX Block Types Used by Statistics

The Block Type column lists what kind of block the counter is on. The block types are as follows:

Block Type	Description
END-RULE	The counter on the end block. The Block Name column in the line of the END-RULE describes what FMX has done with the event.
ENM-EVENT-TRIGGER	The counter for all new events that have arrived.
EVENT-CLEARING-TRIGGER	The counter for all clearings of events that have arrived.
ACKNOWLEDGE-TRIGGER	The counter for all acknowledgements of events that have arrived.

7.4.2.2 Events Used by ECX End Rule Blocks in Statistics Output

The Block Name for END-RULEs are named according to the following naming convention:

`<FMX_action_on_the_event>.<Rule_Name>.<Extra_information>`

```
For example:
Filter.AcknoledgedAlarmCorrelation
```

The possible values of `<FMX_action_on_the_event>` are listed in the table below:

FMX_action_on_the_event	Description
Filter	The alarm is not shown in the ALV, because it is cleared during the short-filter time.

7.5 GSM Surveillance eXpert (GRX)

The purpose of the GRX Rule Modules Package is to assist operators with the alarms management.

GRX package focus is on filtering and correlation of most common GSM Network alarms.

The GRX includes the following modules and functions:

— GRX_BTS_External rule module

— GRX_CELL_Logical rule module

— GRX_GB_Interface rule module

— GRX_MO_Fault rule module

These Rule Modules require that all NSX Rule Modules and the following FMX Utility Modules are loaded:

— `baseblocks-module`

— `exclusive-region-module`

— `enm-blocks-module`

— `enmmml-blocks-module`

— `semaphore`

— `fmx-utility package`

7.5.1 Configure GRX Module Parameters

You can control the GRX Rule Module behavior on runtime with configuration parameters.

The Rule Modules use their predefined default parameter values when firstly loaded in the system. Operator can change the values from Module Management User Interface by selecting the module and choosing Module Parameters. A pane opens and the valuescan be changed in the Current Value column. For more information, see the *Automatic Alarm Handling (FMX)* online help.

7.5.1.1 GRX Module Parameters

The following parameters are used in the GRX modules:

Table 7 GRX_GB_Interface

Module Parameter	Description	Default
Filter_Time	It controls how long (in minutes) to wait before checking if an alarm is still active. It is needed to recognize short-lived alarms and keeping the Alarm Monitor clean. Decimal values are allowed. For example, use 2.5 for two and a half minutes.	3
Frequent_Limit	It controls how many instances of the alarm need to occur before a FREQUENT alarm is issued.	3
Frequent_Interval	It controls for how many minutes back in time the frequent check is made. Example: Assume that the Frequent_Limit is set to 3 and Frequent_Interval is set to 30; no FREQUENT alarm is active, but two alarms, that FMX is counting, occurred 10 and 20 minutes ago. If one more alarm arrives within 10 minutes, the threshold is reached and a FREQUENT alarm is issued. However, if it takes 15 minutes for the last alarm to arrive, the threshold is not reached. Then, the threshold counter remains at two, and there are five minutes left for the frequent limit to be reached. The Frequent_Interval parameter is also used to clear the FREQUENT alarm when no further alarm notifications occur during the Frequent_Interval.	30
Frequent_Poll_Time	For Frequent alarms, the FMX continuously updates the FMX_Additional attribute with the current number of alarms issued during the Frequent_Interval. The counter is changed on polling basis. The time between each update is defined by the Frequent_Poll_Time. The recommended values of Frequent_Poll_Time are between a quarter of- to the whole of the Frequent_Interval.	1
Hide_During_FRQ	It defines if the original triggering alarms are hidden during a FREQUENT alarm situation. The possible values are TRUE and FALSE. The default value is FALSE.	TRUE

Table 8 GRX_BTS_External Module

Module Parameter	Description	Default
FREQUENT_SEVERITY	FREQUENT_SEVERITY=0,1-4, trigger	trigger
MAX_WAIT_TIME	Specifies maximum waiting time in seconds, that the clearing rule waits for a RADIO X-CEIVER ADMINISTRATION BTS EXTERNAL FAULT to get active again before it clears it..	20
WAIT_FM_CONFIRM	Specifies the time in seconds the rule waits for receipts of separated alarm creations or	20

Module Parameter	Description	Default
	clearings (from FM in OSS). This must normally not be modified.	

Table 9 GRX_Cell_Logical

Module Parameter	Description	Default
Frequent_Interval	The interval (in minutes) during which the frequent limit is checked.	60
Frequent_Limit	How many instances of the alarm that need to occur before a FREQUENT alarm is issued..	3
Frequent_Poll_Time	The time (in minutes) between frequent checks.	5
Hide_During_Frequent	Original alarm hidden when FRQ alarms shown.	FALSE
MML_timeout	Maximum time in seconds to wait for an MML session to finish.	180
TOPOLOGY_valid_time	How long in hours to keep relations between TG-RSITE and CELLS before fetching it again.	24
graceAfterBTSDOWN	Time in seconds to wait after BTS DOWN is created before continuing processing.	5
graceBeforeFRQ	Time in seconds to allow all simultaneous Cell alarms to arrive before counting them.	3
onlyBCCHforBTSDown	Only consider Cell alarms on BCCH when deciding on BTS DOWN alarms.	TRUE

Table 10 GRX_MO_FAULT

Module Parameter	Description	Default
Filter_Time	It controls how many minutes to wait before checking if an alarm is still active. It is needed in order to recognize short-lived alarms and keeping the Alarm Monitor clean. Decimal values are allowed. For example, use 2.5 for two and a half minutes.	3
Frequent_Enabled	Specifies if the alarms, of type RADIO X-CEIVER ADMINISTRATION MANAGED OBJECT FAULT, create alarms of the type FREQUENT RADIO X-CEIVER ADMINISTRATION MANAGED OBJECT FAULT if conditions for this are fulfilled.	TRUE
Frequent_Limit	It controls how many instances of the alarm need to occur before a FREQUENT alarm is issued.	5
Frequent_Interval	It controls for how many minutes back in time the frequent check is made. Example: Assume that the Frequent_Limit is set to 3 and Frequent_Interval is set to 30; no FREQUENT alarm is active, but two alarms, that FMX is counting, occurred 10 and 20 minutes ago. If one more alarm arrives within 10 minutes, the threshold is reached and a FREQUENT alarm is issued. However, if it takes 15 minutes for the last alarm to arrive, the threshold is not reached. Then, the threshold counter remains at two, and there are five minutes left for the frequent limit to be reached.	60

Module Parameter	Description	Default
	The Frequent_Interval parameter is also used to clear the FREQUENT alarm when no further alarm notifications occur during the Frequent_Interval.	
Frequent_Poll_Time	For Frequent alarms, the FMX continuously updates the FMX_Additional attribute with the current number of alarms issued during the Frequent_Interval. The counter is changed on polling basis. The time between each update is defined by the Frequent_Poll_Time. The recommended values of Frequent_Poll_Time are between a quarter of- to the whole of the Frequent_Interval.	120
Hide_During_FRQ	It defines whether the original triggering alarms are hidden during a FREQUENT alarm situation. The possible values are TRUE and FALSE.	TRUE
Frequent_Severity	Specifies how the perceived severity attribute of a FREQUENT RADIO X-CEIVER ADMINISTRATION MANAGED OBJECT FAULT must be set. **trigger** means that the value from the original alarm is used.	trigger
Max_TS_Block_Deblock	Specifies the maximum number of block-deblockMML commands to be sent within TS_Block_Deblock_Time. This is to set an upper limit for number of block-deblock commands sent.	3
Shortlived_Filter	Specifies if the alarms of type RADIO X-CEIVER ADMINISTRATION MANAGED OBJECT FAULT are processed by the short lived filter.	TRUE
Slogans_omit_from_shortlived	Specifies the Slogans to omit from shortlived.	MAINS FAILURE:PERMANENT FAULT
Slogans_send_RXMFP	Specifies the slogans for which RXMFP is sent.	BTS EXTERNAL:BTS INTERNAL:MAINS FAILURE:OPERATOR CONDITION:PERMANENT FAULT
TG-TRX-CELL_valid_time	The time in hours to remember TG-CELLS or TRX-CELL relations	24
TS_Block_Deblock	Specifies if the rule must use the block-deblock MML commands for alarms on time slots.	TRUE
TS_Block_Deblock_Time	Specifies the maximum time in minutes the rule allows Max_TS_Block_Deblock MML commands to be sent. This is to set an upper limit for number of block-deblock commands sent.	30
TS_MML_timeout	Specifies the maximum time in minutes that the rule waits for a MML answer.	5
TS_RXLTI_Enabled	Specifies if the RXLTI command is sent when a time-slot is blocked.	TRUE
WAIT_AFTER_BLOCK-DEBLOCK	Specifies the time in seconds that the rule waits after a block-deblock MML command have been sent before checking the state of the alarm.	20
WAIT_FM_CONFIRM	Specifies the time in seconds the rule waits for receipts of separated alarm creations/ clearings (from FM in OSS). This must normally not be modified.	20

7.6 Core Network Surveillance eXpert (CNX)

The purpose of the CNX Rule Modules Package is to assist operators with the alarms management.

CNX package focus is on filtering and correlation of most common Core Network alarms.

The CNX includes the following modules and functions:

— CN_External_Alarms rule module

— CN_Signalling_Disturbances rule module

— CN_Switching rule module

— CN_Transmission_disturbances rule module

This Rule Module requires that all NSX Rule Modules and the following FMX Utility Modules are loaded:

— `baseblocks-module`

— `exclusive-region-module`

— `enm-blocks-module`

— `enmmml-blocks-module`

7.6.1 Configure CNX Module Parameters

You can control the CNX Rule Module behavior on runtime with the Module Parameters.

The Rule Module uses its predefined default parameter values when first loaded in the system. You can change the values from Module Management User Interface by selecting the module and choosing Module Parameters. A pane opens and you can change the values in the Current Value column. For more information, see the *Automatic Alarm Handling (FMX)* online help.

7.6.1.1 CNX Module Parameters

The following parameters are used in the CNX modules:

Module Parameter	Description
`Filter_Time`	It controls how many minutes to wait before checking if an alarm is still active. This causes a delay on alarm presentation, but is needed in order to recognize short-lived alarms while keeping the Alarm Monitor clean. Decimal values are allowed. For example, use 2.5 for two and a half minutes.
`Frequent_Limit`	It controls how many instances of the alarm need to occur before a FREQUENT alarm is issued. For each situation, the counter is

Module Parameter	Description
	based on a combination of Managed Element and relevant object (for example, Cell-name).
Frequent_Interval	It controls how many minutes back in time the frequent check is made. Example: Assume that the Frequent_Limit is set to 3 and Frequent_Interval is set to 30; no FREQUENT alarm is active, but two alarms, that FMX is counting, occurred 10 and 20 minutes ago. If one more alarm arrives within 10 minutes, the threshold is reached and a FREQUENT alarm is issued. However, if it takes 15 minutes for the last alarm to arrive, the threshold is not reached. Then, the threshold counter remains at two, and there are five minutes left for the frequent limit to be reached. The Frequent_Interval parameter is also used for another purpose. The FMX automatically clears the FREQUENT alarm when no new original alarms occur during the Frequent_Interval.
Frequent_Poll_Time	For Frequent alarms, the FMX continuously updates the FMX_Additional attribute with the current number of alarms issued during the Frequent_Interval. The counter is adjusted on a polling basis. The time between each update is defined by the Frequent_Poll_Time. The recommended values of Frequent_Poll_Time are between a quarter of- to the whole of the Frequent_Interval.
Hide_During_FRQ	It defines whether the original triggering alarms are hidden during a FREQUENT alarm situation. The possible values are TRUE and FALSE, the default value is FALSE.
Wait_on_secondary	Controls how many minutes to wait in a correlation rule. When the main alarm (primary alarm) is ceased, the FMX waits for this amount of minutes, before checking if there are any secondary alarms still active. This grace period is used to make sure that the AXE has enough time to issue clearings for the secondary alarms.
Reset	It is a parameter used by the DIP_QUALITY_SUP rule, it can be TRUE or FALSE, and if set to true, the rule automatically resets the counter for the DIP quality by sending a DTQSR.
ST-DIP_valid_time	The parameter is used in the C7 SL FAILURE rule. The rule sends a few MML-commands to fetch the DIP related to the ST of the alarm. The state of the DIP is then printed by the DTSTP. The ST-DIP combination is considered static and is remembered internally for the number of hours set by the value of STDIP_valid_time. If the time since the ST-DIP combination was fetched exceeds the STDIP_valid_time, the sequence of MML-commands is sent yet again to get the new DIP value. The solution is used to limit the number of MML-sessions and thus increase performance.
run_STRDP	The parameter is used by the BLOCKING_SUPERVISION rule and determines if the STRDP command needs to be sent or not (the printout can be very big). The default value is FALSE.
STRDP_output	The parameter is used by the BLOCKING_SUPERVISION rule and determines if the STRDP output command is added to the problem text of the alarm or saved to file. If it is set to ALARM (default value), the STRDP result is added to the Problem text of the alarm, otherwise, it is saved in the /var/opt/ericsson/fmx/modules/CN_Transmission_Disturbances/STRDP...txt file.
DIP-R_valid_time	The parameter is used in the DIP_FAULT_SUP and DIP_UNAVAILABLE_STATE rules. The rule sends a few MML-commands to fetch the Route related to the DIP. This relation is considered static and remembered internally for the number of hours set by this parameter, similar to ST-DIP_valid_time.
Max_interval	It is used in the DIP_QUALITY_SUP rule, it sets the maximum length of a gliding time window, within which a maximum number of DTQSR commands can be issued for that DIP. For more information, refer to Max_reset_per_DIP on dl-MaxResetperDIP below. For a given combination of NE and DIP, the DTQSR command is sent Max_reset_per_DIP times during Max_interval at the most.
Max_reset_per_DIP	It is used in DIP_QUALITY_SUP rule, it sets the maximum number of reset attempts per DIP by the DTQSR command within the Max_interval parameter. When Max_reset_per_DIP is passed within Max_interval, a FREQUENT DIP QUALITY SUPERVISION alarm is issued.

Module Parameter	Description
Multiple_Limit	This parameter is used in the C7 SL FAILURE rule. It is the minimum number of CCITT7 SIGNALLING LINK FAILURE alarms on the same Link Set (LS), before the rule creates a MULTIPLE CCITT7 SIGNALLING LINK FAILURE for that LS.
Multiple_Poll_Time	This parameter is used in the Multiple C7.

7.6.2 CN_External_Alarms Rule Module

Alarms

Alarms handled by this module are:

— EXTERNAL ALARM (original) Normal processed

— FREQUENT EXTERNAL ALARM (FMX) Post-processed

Module Description

Important alarms for the operator are shown immediately, but less important ones pass a short-lived and frequent filter. This decision is based on the information in a text file, where the less important alarm slogans are listed.

The FREQUENT EXTERNAL ALARM is only issued if not already active for the same combination of NE, SLOGAN, and DETAILS, otherwise, the existing FREQUENT EXTERNAL ALARM is updated.

The FREQUENT EXTERNAL ALARM is eventually cleared when the situation is no longer frequent.

Short-lived Filter

This alarm is often short-lived and these instances are filtered out.

Frequent Filter

If the situation becomes frequent for the EXTERNAL ALARM, a FREQUENT EXTERNAL ALARM is created if not already active for the same object.

A FREQUENT EXTERNAL ALARM is updated in the FMX_Additional attribute with information on how frequent the alarm is.

It is possible to configure the rule to suppress a persistent EXTERNAL ALARM during a frequent situation by changing the value of the parameter.

Additional Attributes

Table 11

Attribute	Information
FMX_Status	`Rdy/FRQ/ERR/Off`
FMX_Additional	Frequent External Alarm: Number of alarms that last x minutes, [Alarm SLOGAN] External Alarm: [Alarm slogan]
FMX_Object	[Alarm details]

Configure CNX Parameters for External Alarms

The configuration for external alarms that are treated as short-lived is stored in the `/etc/opt/ericsson/cnx/external_alarms_short_filter.txt` file. To change the configuration, edit the file with any text editor.

7.6.3 CN_Signalling_Disturbances Rule Module

Alarms

Alarms handled by this module are:

— CCITT7 SIGNALLING LINK FAILURE (original) Normal processed

— SIGNALLING FAULT SUPERVISION (original) Normal processed

— FREQUENT CCITT7 SIGNALLING LINK FAILURE (FMX) Post-processed

— FREQUENT CCITT7 SIGNALLING LINK SUPERVISION (FMX) Post-processed

— MULTIPLE CCITT7 SIGNALLING LINK FAILURE (FMX) Post-processed

Module Description

The expected result is that short-lived instances are filtered out, while frequent situations are not missed by issuing a FREQUENT alarm..

CCITT7 SIGNALLING LINK FAILURE

By sending a series of MML-commands, the corresponding Route, DIP and its STATE are fetched. The FMX adds this information in an attribute displayed in the ALV.

If alarms appear from the same NE and LS combination, a new alarm - MULTIPLE CCITT7 SIGNALLING LINK FAILURE - is created. This new alarm contains a list of alarming STs as discovered from the included CCITT7 SIGNALLING LINK FAILURE instances. The concerned CCITT7 SIGNALLING LINK FAILURE instances are not displayed. Note that shortlived instances are not included in MULTIPLE alarms.

MML commands used:

```
EXSCP:DEV=<st> -> side1&side2

EXDEP:DEV=<side1/side2> -> SNT/DEVP & R

(for BSC, RADEP:DEV=<side1/side2> -> SNT/DEVP)

NTCOP:SNT=<snt/devp> -> DIP

DTSTP:DIP=<dip> -> STATE

C7LTP:LS=<ls>

C7TSP:ST=<st-list>
```

SIGNALLING FAULT SUPERVISION

If the situation becomes frequent, the alarm FREQUENT SIGNALLING FAULT SUPERVISION alarm is issued, if not already active for the same combination of NE and ACL.

The rule also adds the result of FAIAP:ACL=acl to the Problem Text for instances that are not short-lived.

MML commands used:

```
FAIAP:ACL=<acl>
```

Short-lived filter

Both CCITT7 SIGNALLING LINK FAILURE and SIGNALLING FAULT SUPERVISION alarms are often short-lived and these instances are filtered out.

Frequent Filter

If the situation becomes frequent, a FREQUENT alarm is created if not already active for the same object. The FREQUENT alarms are updated in the FMX_Additional attribute with information on how frequent the alarm is.

It is possible to configure the rule to suppress persistent CCITT7 SIGNALLING LINK FAILURE and SIGNALLING FAULT SUPERVISION alarms during a frequent situation by setting the parameter.

Note that if the parameter is set to TRUE in combination with MULTIPLE, handling can produce situations where the MULTIPLE alarm contains, for example, three STs while a fourth one with fluctuating behavior is only visible in the FREQUENT alarm. Thus, MULTIPLE and FREQUENT alarms for the same LS need to be viewed together.

Additional Attributes

Table 12

Attribute	Information
FMX_Status	`Inv/Rdy/FRQ/MLT/Rwe/ERR/Off`
FMX_Additional	CCITT7 SIGNALLING LINK FAILURE: (DIP, DIP-STATE, Route for non BSCs) FREQUENT CCITT7 SIGNALLING LINK FAILURE: (number of alarms last x minutes, DIP, Route for non BSCs) MULTIPLE CCITT7 SIGNALLING LINK FAILURE: (list of alarming STs) FREQUENT SIGNALLING FAULT SUPERVISION: (number of alarms last x min)
FMX_Object	CCITT7 SIGNALLING LINK FAILURE: (LS, SLC) FREQUENT CCITT7 SIGNALLING LINK FAILURE: (LS, SLC) MULTIPLE CCITT7 SIGNALLING LINK FAILURE: (LS)

Attribute	Information
Problem Text	CCITT7 SIGNALLING LINK FAILURE: (FCODE in text if applicable, DIP, DIP-STATE, Route for non-BSCs, C7TSP printout) FREQUENT CCITT7 SIGNALLING LINK FAILURE: (DIP, DIP-STATE, Route for non-BSCs, C7LTP printout) MULTIPLE CCITT7 SIGNALLING LINK FAILURE: (C7LTP and C7TSP printout) SIGNALLING FAULT SUPERVISION: (Result of the MML command FAIAP:ACL=)

7.6.4 CN_Switching Rule Module

Alarms

Alarms handled by this module are:

— SWITCHING NETWORK TERMINAL FAULT (original) Normal processed

— FREQUENT SWITCHING NETWORK TERMINAL FAULT (FMX) Post-processed

Module Description

The rule manages the SWITCHING NETWORK TERMINAL FAULT alarm. It handles short-lived, frequent and normal situations. Short-lived alarms are filtered out.

SWITCHING NETWORK TERMINAL FAULT

If the alarm becomes frequent, a FREQUENT SWITCHING NETWORK TERMINAL FAULT alarm is sent to the Alarm Monitor from the FMX, based on the SNT value. The STATE and FCODE values are also extracted and added to the alarm information.

It is possible to configure the rule to suppress persistent SWITCHING NETWORK TERMINAL FAULT alarms during a frequent situation by changing the parameter.

MML commands used:

```
NTCOP:SNT=<snt>
NTSTP:SNT=<snt>
```

Short-lived Filter

This alarm is often short-lived and these instances are filtered out.

Frequent Filter

If more than short-lived SWITCHING NETWORK TERMINAL FAULT alarms arrive in , a FREQUENT NETWORK TERMINAL FAULT alarm is created if not already active.

Additional Attributes

Table 13

Attribute	Information
FMX_Status	`Rdy/Rwe/FRQ/Off`
FMX_Additional	SWITCHING NETWORK TERMINAL FAULT: (STATE) FREQUENT SWITCHING NETWORK TERMINAL FAULT: (number of alarms that last x minutes, STATE) :
FMX_Object	SWITCHING NETWORK TERMINAL FAULT: (SNT) FREQUENT SWITCHING NETWORK TERMINAL FAULT: (SNT; STATE)
Problem Text	SWITCHING NETWORK TERMINAL FAULT: (FCODE in text, NTCOP printout, NTSTP printout) FREQUENT SWITCHING NETWORK TERMINALFAULT: (FCODE in text)

7.6.5 CN_Transmission_disturbances Rule Module

Alarms

Alarms handled by this module are:

- `BLOCKING SUPERVISION` (original) Normal processed
- `DIGITAL PATH FAULT SUPERVISION` (primary-1) Normal processed
- `DIGITAL PATH UNAVAILABLE STATE FAULT` (secondary-1) Normal processed
- `DIGITAL PATH QUALITY SUPERVISION` (original) Normal processed
- `SYNCHRONOUS DIGITAL PATH FAULT SUPERVISION` (primary-2) Normal processed
- `SYNCHRONOUS DIGITAL PATH UNAVAILABLE STATE FAULT` (secondary-2) Normal processed
- `FREQUENT DIGITAL PATH FAULT SUPERVISION` (FMX) Post-processed
- `FREQUENT DIGITAL PATH UNAVAILABLE STATE FAULT` (FMX) Post-processed
- `FREQUENT DIGITAL PATH QUALITY SUPERVISION` (FMX) Post-processed
- `FREQUENT SYNCHRONOUS DIGITAL PATH FAULT SUPERVISION` (FMX) Post-processed
- `FREQUENT SYNCHRONOUS DIGITAL PATH UNAVAILABLE STATE FAULT` (FMX) Post-processed

Module Description

The expected result is that short-lived instances are filtered out, while frequent situations are not missed by issuing a FREQUENT alarm.

BLOCKING SUPERVISION

The rule shows `BLOCKING SUPERVISION` alarms with additional information in the Problem Text. Printout from the MML command `STRDP:R=r,STATE=state` is the source for this additional information. The

command is sent once with `STATE = BLOC&SEAL&LIBL`.

The rule is also triggered by the repeated instances (with increasing or decreasing ACL) of this alarm coming from the same NE.

MML commands used:

```
STRDP:R=<r>,STATE=BLOC&SEAL&LIBL
```

DIGITAL PATH FAULT SUPERVISION

If the alarm is persistent, the rule issues some MML commands to find and present the route with the alarm (not for BSC Network Elements alarms).

Furthermore, `DIGITAL PATH UNAVAILABLE STATE FAULT` is regarded as a secondary alarm compared to `DIGITAL PATH FAULT SUPERVISION` for the same DIP. It is of no interest to show `DIGITAL PATH UNAVAILABLE STATE FAULT` for a DIP, if a `DIGITAL PATH FAULT SUPERVISION` alarm is already active for the same DIP. The rule correlates these two alarms.

When a `DIGITAL PATH FAULT SUPERVISION` alarm eventually ceases, the FMX checks if there are any `DIGITAL PATH UNAVAILABLE STATE FAULT` active alarms for the same DIP. FMX presents them to the operator. A grace period between the ceasing and the check is controlled through the parameter.

MML commands used:

```
DTDIP:DIP=<dip> -> SNT

NTCOP:SNT=<snt> -> DEV

EXDEP:DEV=<dev> -> R

STRSP:R=<r> -> Route_Quality

DTQDP:DIP=<dip> -> DIP_Quality
```

DIGITAL PATH UNAVAILABLE STATE FAULT

Short-lived occurrences of `DIGITAL PATH UNAVAILABLE STATE FAULT` alarms are filtered out, while keeping count on the frequency of them. If conditions are met, a `FREQUENT DIGITAL PATH UNAVAILABLE FAULT` alarm is issued, if not already present for the same DIP and NE combination.

If the alarm is persistent, the rule issues some MML commands to find and present the route with the alarm (not for BSC alarms).

Furthermore, `DIGITAL PATH UNAVAILABLE STATE FAULT` is regarded as a secondary alarm compared to `DIGITAL PATH FAULT SUPERVISION` for the same DIP. A check is made to see if any `DIGITAL PATH FAULT SUPERVISION` alarm is active for the same combination of DIP and NE. If so, the `DIGITAL PATH UNAVAILABLE STATE FAULT` alarm is not presented at this stage.

.

MML commands used:

```
DTDIP:DIP=<dip> -> SNT

NTCOP:SNT=<snt> -> DEV

EXDEP:DEV=<dev> -> R

STRSP:R=<r> -> Route_Quality

DTQDP:DIP=<dip> -> DIP_Quality
```

DIGITAL PATH QUALITY SUPERVISON

This alarm is not short-lived. To reduce the number of presented alarms, the alarm is automatically reset with the DTQSR command. It is possible to turn on or off the DTQSR part by configuring the parameters. The rule has a parameter that limits the number of resets to the same DIP in a defined time (). When this limit has been exceeded, a `FREQUENT DIGITAL PATH QUALITY SUPERVISION` alarm is issued. The alarm is cleared when the number of reset attempts for the time period fall below the threshold.

MML commands used:

```
DTQSR:DIP=<dip>,<DTQSRparam>
```

SYNCHRONOUS DIGITAL PATH FAULT SUPERVISION

Short-lived occurrences of `SYNCHRONOUS DIGITAL PATH FAULT SUPERVISION` alarms are filtered out, while keeping count on the frequency of them. If conditions are met, a `FREQUENT SYNCHRONOUS DIGITAL PATH FAULT SUPERVISION` alarm is issued if not already present for the same SDIP.

If the alarm is persistent, the rule issues MML commands and enriches the alarm with the corresponding DIP and STATE values.

.

MML commands used:

```
TPCOP:<sdip> ->DIP

DTSTP:DIP=<dip> -> STATE
```

Short-lived Filter

Short-lived alarms of the following types are filtered out and not shown unless the threshold is exceeded and the alarm is still active:

— DIGITAL PATH FAULT SUPERVISION

— DIGITAL PATH UNAVAILABLE STATE FAULT

— DIGITAL PATH QUALITY SUPERVISION

— SYNCHRONOUS DIGITAL PATH FAULT SUPERVISION

— SYNCHRONOUS DIGITAL PATH UNAVAILABLE STATE FAULT

Frequent Filter

For DIGITAL PATH FAULT SUPERVISION and DIGITAL PATH UNAVAILABLE STATE FAULT alarms, the FMX checks if the alarm is frequent for the combination of DIP and NE. If so, the FMX generates an alarm, FREQUENT DIGITAL PATH SUPERVISION FAULT or FREQUENT DIGITAL PATH UNAVAILABLE STATE FAULT, unless one for this combination of DIP and NE in the ALV exists. The expected result is that only alarms that are persistent are presented, while frequent situations are not missed.

For SYNCHRONOUS DIGITAL PATH FAULT SUPERVISION and SYNCHRONOUS DIGITAL PATH UNAVAILABLE STATE FAULT, the FMX checks if the alarm is frequent for the combination of SDIP and NE. If so, the FMX generates an alarm, FREQUENT SYNCHRONOUS DIGITAL PATH SUPERVISION FAULT or FREQUENT SYNCHRONOUS DIGITAL PATH UNAVAILABLE STATE FAULT, unless one combination of SDIP and NE in the Alarm Monitor exists.

For DIGITAL PATH QUALITY SUPERVISION, the FMX issues a FREQUENT DIGITAL PATH QUALITY SUPERVISION alarm for a given combination of DIP, NE and PARAM (BFF, SES, and so on) when a (configurable) number of reset attempts on the DIP have been issued.

The expected result is that only alarms that are persistent are presented, while frequent situations are not missed.

It is possible to configure the rule to suppress persistent DIGITAL PATH FAULT SUPERVISION, DIGITAL PATH UNAVAILABLE STATE FAULT, SYNCHRONOUS DIGITAL PATH FAULT SUPERVISION, and SYNCHRONOUS DIGITAL PATH UNAVAILABLE STATE FAULT alarms during a frequent situation by setting the parameters for the different rules.

Additional Attributes

Table 14

Attribute	Information
FMX_Status	Inv/Rdy/FRQ/Rwe/ ERR/Off
FMX_Additional	BLOCKING SUPERVISION: (Blocking-%, BLO, NDV) DIGITAL PATH FAULT SUPERVISION: (FAULT, Route for non BSC Network Elements) DIGITAL PATH UNAVAILABLE STATE FAULT: (Route for non BSC Network Elements, "(Prev.Sup.)" if brought back after correlation) DIGITAL PATH QUALITY SUPERVISION: (PARAM, "Not reset (limit reached)" if the maximum DTQSR reached). SYNCHRONOUS DIGITAL PATH FAULT SUPERVISION: (LAYER, FAULT) SYNCHRONOUS DIGITAL PATH UNAVAILABLE STATE FAULT: (LAYER, DIRECTION, "(Prev. Sup.)" if brought back after correlation) FREQUENT DIGITAL PATH FAULT SUPERVISION: (number of alarms that last x minutes, FAULT, Route for non BSC Network Elements) FREQUENT DIGITAL PATH UNAVAILABLE STATE FAULT: (number of alarms that last x minutes, Route for non BSC Network Elements) FREQUENT DIGITAL PATH QUALITY SUPERVISION: (PARAM)

Attribute	Information
	FREQUENT SYNCHRONOUS DIGITAL PATH FAULT SUPERVISION: (number of alarms that last x minutes, LAYER, FAULT) FREQUENT SYNCHRONOUS DIGITAL PATH UNAVAILABLE STATE FAULT: (number of alarms that last x minutes, LAYER, DIRECTION)
FMX_Object	BLOCKING SUPERVISION: (Route) DIGITAL PATH FAULT SUPERVISION: (DIP) DIGITAL PATH UNAVAILABLE STATE FAULT: (DIP) DIGITAL PATH QUALITY SUPERVISION: (DIP) SYNCHRONOUS DIGITAL PATH FAULT SUPERVISION: (SDIP) SYNCHRONOUS DIGITAL PATH UNAVAILABLE STATE FAULT: (SDIP) FREQUENT DIGITAL PATH FAULT SUPERVISION: (DIP) FREQUENT DIGITAL PATH UNAVAILABLE STATE FAULT: (DIP) FREQUENT DIGITAL PATH QUALITY SUPERVISION: (DIP) FREQUENT SYNCHRONOUS DIGITAL PATH FAULT SUPERVISION: (SDIP)

Attribute	Information
	FREQUENT SYNCHRONOUS DIGITAL PATH UNAVAILABLE STATE FAULT: (SDIP)
Problem Text	BLOCKING SUPERVISION: (STRDP printout, if configured and not redirected to file). DIGITAL PATH FAULT SUPERVISION: (DTQDP printout, STRSP printout for non BSC Network Elements) DIGITAL PATH UNAVAILABLE STATE FAULT: (DTQDP printout, STRSP printout for non BSC Network Elements) DIGITAL PATH QUALITY SUPERVISION: (Reason why DTQSR was not sent). SYNCHRONOUS DIGITAL PATH FAULT SUPERVISION: (DIP info) SYNCHRONOUS DIGITAL PATH UNAVAILABLE STATE FAULT: (DIP info) FREQUENT DIGITAL PATH FAULT SUPERVISION: (DTQDP printout, STRSP printout for non BSC Network Elements) FREQUENT DIGITAL PATH UNAVAILABLE STATE FAULT: (DTQDP printout, STRSP printout for non BSC Network Elements) FREQUENT DIGITAL PATH QUALITY SUPERVISION: (Frequency information) FREQUENT SYNCHRONOUS DIGITAL PATH FAULT SUPERVISION: (DIP info) FREQUENT SYNCHRONOUS DIGITAL PATH UNAVAILABLE STATE FAULT:

Attribute	Information
	(DIP info)

7.7 Radio Network eXpert (RNX)

RNX Rule Modules Package improves customer experience providing RAN features on ENM level using rules developed with Automatic Alarm Handling (FMX). FMX rules are packaged into rule modules which are handled in the FMX Module Management in ENM.

The rule modules are activated by default at ENM installation. If a rule module is deactivated, an ENM upgrade does not activate it again.

Prerequisites

- FMX is setup with users, user roles and ssh access.
- An `fmxenmcli` user with user roles Cmedit_Operator, Cmedit_Administrator, and FM_Operator roles must exist on the system. The user must have status "Enabled".

  Password Ageing must have Customize Password Ageing selected and the Enable Password Ageing tick box must NOT be ticked. See Create System User for Automatic Alarm Handling (FMX) Interaction with ENM CLI on page 71.
- To enable FMX rules to send AMOS commands, an fmxamos user must have been created and a configuration file edited. See Configure CPP Utility to Support AMOS Execution on page 74.
- Password-less login to the nodes must have been setup for the fmxamos user.
- Configuration is setup to enable FMX Rules to use SSH to the AMOS Server.

  This can be done in two ways as follows:

  - configuring ciphers for SSH. See Configure Ciphers for SSH on page 105.
  - changing in the configuration file `/ericsson/tor/data/fmx/etc/modules/ssh-block-module/ssh-block-module.properties` the `automaticallyUpdateThisFileOnLoad=true` to `false` and `useDefaultInsecureJschAlgorithms=false` to `true`.

7.7.1 Cell Manually Locked

> The feature aims to improve network availability.

Cell Manually Locked feature makes an operator aware that a cell is manually locked by issuing a “Cell Manually Locked” alarm in ENM if cells are locked for more than 30 minutes. For GRAN nodes, the alarm raised is “TRX Manually Locked”.

The 30-minute time between discovery of locked cell until alarm creation is a default value and configurable by parameter `AllowedLockTime`. The severity of the alarm is by default set to Major but can be changed by changing parameter `Severity`.

The Cell Manually Locked FMX rule suite consist of one rule module for each RAN technology, GRAN, WRAN and LTE, and one rule for common procedures.

Parameters are set per FMX rule module and can be accessed from Module Parameters in Automatic Alarm Handling (FMX) application in ENM.

Note: The rule cannot be activated for a limited set of nodes, it is either for the whole network or not at all.

FMX Rule Modules

— RNX_Manually_Locked_TRX-GRAN

— RNX_Manually_Locked_Cell-WRAN

— RNX_Manually_Locked_Cell-LTE

— RNX_Common_Procedures

Each of the Manually_Locked-modules has the following parameters:

Module Parameter	Description	Default
`AllowedLockedTime`	It specifies the allowed time in minutes between detection of a locked cell until a Cell Manually Locked Alarm is created.	30 min
`Severity`	It determines the severity (one of Critical, Major, Minor, Warning or Indeterminate) of the Cell Manually Locked alarm.	Major

FMX Utility Modules Required

— baseblocks-module

— exclusive-region-module

— enm-blocks-module

— enmcli-blocks-module

— RNX-utility-package

7.7.2 Health Check Rule

The purpose of the RNX_HealthCheck rule is to assure that accesses and other configurations required by the rules, included in the rule package, work. The rule raises internal ENM alarms to identify what is wrong.

FMX Rule Module

— RNX_HealthCheck

The FMX Rule module has the following parameter:

Module Parameter	Description	Default
`polltime`	It specifies how often the services are checked.	15 min

FMX Utility Modules Required

— baseblocks-module

— enm-blocks-module

— enmcli-blocks-module

— exclusive-region-module

— ssh-block-module

— cpp-block-module

7.7.3 DCGM Log Collection

The DCGM Log Collection feature reacts on certain alarms and sends the AMOS command `dcgm` to collect the logs automatically.

The FMX connection is configured with the following properties:

— `automaticallyUpdateThisFileOnLoad=false`

— `useDefaultInsecureJschAlgorithms=true`

The `automaticallyUpdateThisFileOnLoad` and `useDefaultInsecureJschAlgorithms` properties are located in the file:

`/ericsson/tor/data/fmx/etc/modules/ssh-block-module/ssh-block-module.properties`

The FMX rule module RNX_dcgcollection triggers on alarms specified in the file:

/var/opt/ericsson/fmx/etc/modules/RNX_dcgcollection/trigger_file

and it has the following configuration parameters:

Table 15 RNX_dcgcollection Configuration Parameters

Module Parameter	Description	Default
max_node_waiting_time_sec	It specifies the maximum waiting time for each dcgm log collection to complete. It is settable within a range of 25-120 minutes. If set outside of that range, the default value is used.	40 min
shortlived_time_minutes	It sets the time in minutes the alarm needs to be active for the log collection to start. If raised and cleared within shortlived_time_minutes, the alarm is filtered. If an alarm with identical SpecificProblem and ProblemText appears more than three times within one hour, then log collection is initiated regardless.	2 min

8 Node Log Management Administration Tasks

> This section contains the routine operation and maintenance tasks related to the administration of Node Log management.

8.1 Configure Node Log Automatic Housekeeping

This section describes how to configure automatic housekeeping parameters via PIB for Node Log service.

Automatic housekeeping allows to automatically remove from ENM shared file system the node log files uploaded through Node Log service when retention period has expired.

Two parameters can be used to configure the housekeeping:

`NodeLogHousekeepingRetentionPeriod`
: This is the retention period in days, the range is from 1 to 7 days (default is 7 days).

`NodeLogHousekeepingDayRecurrence`
: This is the daily frequency for housekeeping check of the ENM file system, the range is between 1 to 4 times (default is 4 times).

User has root access to the ENM server.

The housekeeping retention period parameter values are changed or read.

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

8.1.1 Node Log Housekeeping Parameters

Parameter Name	Default Value	Parameter Description	Value Range
`NodeLogHousekeepingRetentionPeriod`	7 days	Retention period duration in days to apply in the ENM Shared File System.	1..7 (days)
`NodeLogHousekeepingDayRecurrence`	4 times	Daily frequency for triggering the housekeeping check onto the ENM shared file system	1-4 (times)

9 Autonomic Incident Management Administration Tasks

> This section contains the procedures that can be used to initiate and maintain the Autonomic Incident Management application.

9.1 Setup and Initiate Autonomic Incident Management

> The setup and the initiation of the Autonomic Incident Management application must be done before operation.

The user must perform the following procedure to be able to configure Autonomic Incident Management (AIM) to identify incidents in the network.

Prerequisites

— ENM installed and deployed with Autonomic Incident Management expansion procedure complete.

— System administrator access.

Steps

1. Install the Autonomic Incident Management license.
    a. Copy the license file to the ENM CLI workspace.
    b. Run the install license command:

```
lcmadm install file:<file-name>
```

2. Verify the user role needed for Automatic Incident Management application, see ENM Security System Administrator Guide in CPI, Reference [4]. AIM application supports two predefined application-specific roles:
    — AIM_Operator
    — AIM_Administrator

Role	Description
AIM_Operator	Authorized for actions as an operator in Autonomic Incident Management (Read)

Role	Description
AIM_Administrator	Authorized for actions as an administrator in Autonomic Incident Management (Read, Update)

3. Create User.
    a. Go to *User Management* application.
    b. Click **Create User Profile**.
    c. Assign the username and set the password.
    d. Assign `AIM_Operator` role or `AIM_Administrator` role to the user.
    e. Ensure that `Status` switch is set to `Enabled`.
    f. Click **Save**.
4. Activate/Create KPIs in KPI Management application to be used in Autonomic Incident Management.

    For information on creating, activating, and setting up KPIs for effective detection of problems using Autonomic Incident Management, see Key Performance Indicators and Formula Definitions for Autonomic Incident Management in CPI, Reference [10].

Results

The defined and activated Monitoring KPIs and Service Impact KPIs can be used in Autonomic Incident Management Setup.

A maximum of 16 KPIs for each technology (eNodeB or RNC) can be activated in KPI Management and added to KPI selection panel.

9.2 Configure Incident Publishing to Northbound Interface

> An Autonomic Incident Management administrator can configure AIM to publish incidents to NBI.

By default, publishing of Incidents to NBI is disabled.

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

— A command window is open and you have super user privileges.

— You are connected to the MS.

Name	Parameter Description	Value Range	Default Value	Parameter Type
publishToNbi	Autonomic Incident Management administrator to configure AIM to publish incidents to NBI. This configuration is checked every 10 minutes and when publishToNbi attribute updated,it takes effect.	True, False	False	Boolean

The configuration is checked every 10 minutes and when this happens the publishToNbi attribute takes effect.

9.3 Configure to Raise a PM Event When No KPI Values Are Received from a Node

> An Autonomic Incident Management administrator can configure AIM to raise an event when no KPI values are received from a node. The event is raised stating `no performance data received` for the specified ROP.

The default behavior is not to raise an event. The feature must be turned on after the training period.

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

— A command window is open and you have super user privileges.

— You are connected to the required `MS`.

Name	Parameter Description	Value Range	Default Value	Parameter Type
raiseEventForNodeMissingAllKpis	Autonomic Incident Management administrator to configure AIM to raise an event when no KPI values are received from a node. The event is raised stating no performance data received for the specified ROP. This configuration is checked every 10 minutes and when attribute updated,it takes effect.	True, False	False	Boolean

The configuration is checked every 10 minutes and when this happens, the configuration change takes effect.

10 Analytic Session Record Administration Tasks

This section contains the routine operation and maintenance tasks related to the administration of the Analytic Session Record application.

10.1 Configure Long Running Session Timeout Period

An Analytic Session Record administrator can configure the duration of the long running session timeout period after which the incomplete long running session is output to the NBI.

The default time period is 60 minutes. Reducing this time period increases the frequency incomplete long running sessions are output and increasing the period reduces the frequency incomplete long running sessions are output.

Increasing the period also increases the possibility of data loss - if the timeout period is set to 10 hours there is the possibility of losing 10 hours of long running session data, if it is set to 1 hour, then there is the possibility of losing 1 hour of long running session data.

The configuration is checked every 5 minutes and when this happens the timeout period change takes effect.

Note: Changing this parameter may cause up to 2 minutes of data loss.

— A command window is open and you have super user privileges.

— You are connected to the required MS.

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

10.1.1 Long Running Session Timeout Parameters

Table 16 Long Running Session Timeout Parameters

Parameters Name	Default Value	Parameters Description	Value Range
asrLongRunningSessionTimeoutInMinutes	60	Controls the frequency of when long running sessions are output for ASR-L	15-2000 (minutes)
asrnLongRunningSessionTimeoutInMinutes	60	Controls the frequency of when long running sessions are output for ASR-N	15-2000 (minutes)

Steps

1. Update the asrLongRunningSessionTimeoutInMinutes configuration parameter using the following command.

```
python config.py update --app_server_address=<asrlservice_host_name>:8080 --
name=asrLongRunningSessionTimeoutInMinutes --value=<value>
```

   where `<asrlservice_host_name>` is the host name of the `asrl-service instance`.

   Example

```
python config.py update --app_server_address=asr-1-asrlservice:8080 --name=a
srLongRunningSessionTimeoutInMinutes --value=15
```

2. Read the asrLongRunningSessionTimeoutInMinutes configuration parameter using the following command.

```
python config.py read--app_server_address=<asrlservice_host_name>:8080 --nam
e=asrLongRunningSessionTimeoutInMinutes
```

   where `<asrlservice_host_name>` is the host name of the `asrl-service instance`.

   Example

```
python config.py read --app_server_address=asr-1-asrlservice:8080 --name=asr
LongRunningSessionTimeoutInMinutes
```

   Result:15

3. Update the asrnLongRunningSessionTimeoutInMinutes configuration parameter using the following command.

```
python config.py update --app_server_address=<asrlservice_host_name>:8080 --
name=asrnLongRunningSessionTimeoutInMinutes--value=<value>
```

   where `<asrlservice_host_name>` is the host name of the `asrl-service instance`.

   Example

```
python config.py update --app_server_address=asr-1-asrlservice:8080 --name=a
srnLongRunningSessionTimeoutInMinutes--value=15
```

4. Read the asrnLongRunningSessionTimeoutInMinutes configuration parameter using the following command.

```
python config.py read--app_server_address=<asrlservice_host_name>:8080 --nam
e=asrnLongRunningSessionTimeoutInMinutes
```

where <asrlservice_host_name> is the host name of the asrl-service instance.

Example

```
python config.py read --app_server_address=asr-1-asrlservice:8080 --name=asr →
nLongRunningSessionTimeoutInMinutes
```

Result:15

10.2 Configure Overload Protection Threshold Parameters

If the ASR application becomes overloaded, the processing times are unsustainable. Overload protection protects the ASR application from this scenario. When overload protection is enabled, the following periodic events are dropped to reduce the amount of processing:

— INTERNAL_PER_RADIO_UE_MEASUREMENT

— INTERNAL_PER_UE_TRAFFIC_REP

— INTERNAL_PER_UE_RB_TRAFFIC_REP

The indicator used to identify if ASR is overloaded is the "stage one time".

When the rolling average of the last six "stage one times" for a Kafka partition reaches the upper threshold asrMaxStageOneDurationInMilliseconds, the overload protection is enabled.

When the rolling average of the last six "stage one times" reaches the lower threshold asrAcceptableStageOneDurationInMilliseconds, the overload protection is disabled.

The status is automatically monitored every five minutes.

The following conditions must be respected:

— Fully deployed ENM system with a licensed ASR-L Value Pack installed

— Access to the Management Server

10.2.1 Validations for Overload Protection Threshold Parameters

For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

Table 17 Validation for Overload Protection Threshold Parameters

Parameters Name	Default Value	Parameters Description	Value Range	Other Validations
enableOverloadProtection	3500	When the rolling average of last six "stage one times" reaches the asrMaxStageOneDurationInMilliseconds, the overload protection is enabled.	1000–10000 (milliseconds)	Must be at least 1000ms more than asrAcceptableStageOneDurationInMilliseconds
disableOverloadProtection	2000	When the rolling average reduces to reach asrAcceptableStageOneDurationInMilliseconds, the overload protection is disabled.	1000–10000 (milliseconds)	Must be at least 1000ms less than asrMaxStageOneDurationInMilliseconds
action	N/A	This parameter states the action to perform on the PIB parameters.	[update|read]	N/A

10.2.2 Overload Protection Threshold Parameters

Log on to the Management Server (MS-1) as the litp-admin user, switch to the root user and go to the directory `/ericsson/tor/data/asrl/bin` .

```
[root@ms~]#cd /ericsson/tor/data/asrl/bin
[root@ms bin]#
```

The usage for this script is as follows:

```
This script requires two actions

1)Update with additional parameters such as:

1) ${ENABLE_OVERLOAD_PROTECTION_PARAMETER}

2) ${DISABLE_OVERLOAD_PROTECTION_PARAMETER}

2)Read with no additional parameters


Example of Update: ./asrPibParams.sh --action=update --enableOverloadProtection=
<asrMaxStageOneDurationInMilliseconds> and/or --disableOverloadProtection=<asrAc
ceptableStageOneDurationInMilliseconds>

./asrPibParams.sh --action=update --enableOverloadProtection=5000 --disableOverl
oadProtection=3000

./asrPibParams.sh --action=update --enableOverloadProtection=5000

./asrPibParams.sh --action=update --disableOverloadProtection=3000

Example of Read: ./asrPibParams.sh --action=read
```

Steps

1. Run the script to read the current values of `asrMaxStageOneDurationInMilliseconds` and `asrAcceptableStageOneDurationInMilliseconds` from PIB:

```
./asrPibParams.sh --action=read
```

Result:

```
The enable overload protection value is 3500
The disable overload protection value is 2000
```

2. To update the values that enable and disable overload protection, run the script with the `update` action:

```
./asrPibParams.sh --action=update --enableOverloadProtection=<asrMaxStageOne →
DurationInMilliseconds> --disableOverloadProtection=<asrAcceptableStageOneDu →
rationInMilliseconds>
```

Example

```
./asrPibParams.sh --action=update --enableOverloadProtection=5000 --disableO →
verloadProtection=2000
```

Result:

```
: INFORMATION ( ) : The enable overload protection has been updated to 5000
 : INFORMATION ( ) : The disable overload protection has been updated to 200 →
0
```

Note: It is possible to update just a single parameter via this script. See the script usage for further information.

10.3 Configure MDT Parameters on eNodeB Network Elements

The MDT fields in the ASR-L record are marked with Observability Level = MDT report, for more details see [8]. If it is observed that activating the MDT fields in the ASR-L record has a negative impact on ASR-L performance, then update the MDT parameters on eNodeB Network Elements as follows:.

Network Element Parameter	Value
`MdtConfiguration.reportTrigger`	PERIODICAL
`MdtConfiguration.reportIntervalMdt`	MS_10240

Network Element Parameter	Value
`MdtConfiguration.reportAmountMdt`	16

10.4 Configure Maximum Number of Bearers per ASR-L Session

An Analytic Session Record administrator can configure the Maximum number of Bearers supported per ASR-L session.

By default, ASR-L supports a maximum of 50 bearers per session.

ASR-L creates individual bearer records for each bearer established and released within a session. In rare circumstances, a bearer may be setup and released continually several hundreds of times within a session. This can cause the size of the ASR-L record to exceed the maximum size limit that can be represented in the ASR-L record header (for example 65 KB) with potential issues to the northbound applications.

An upper limit is defined for the maximum number of bearers per session which can be configured via PIB (Platform Integration Bridge) parameter.

Prerequisites

A command window is open and you have super user privileges.

Note: For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

Name	Parameter Description	Value Range	Default Value	Parameter Type
asrMaxBearerPerSession	Maximum number of Bearers supported per ASR-L session	1-75	50	Integer

10.5 Configure Maximum number of PDU Sessions and Data Radio Bearers (DRBs) per ASR-N Session

An Analytic Session Record administrator can configure the Maximum number of PDU Session and DRBs supported per ASR-N session.

By default, ASR-N supports a maximum of 25 PDU Sessions and DRBs each per session.

ASR-N creates individual PDU Session and DRB records for each bearer established and released within a session. In rare circumstances, a bearer may be setup and released continually several hundreds of times within a session. This causes the size of the ASR-N record to exceed the maximum size limit that can be

represented in the ASR-N record header, thereby causing potential issues to the northbound applications.

An upper limit is defined for the maximum number of bearers per session which can be configured via PIB (Platform Integration Bridge) parameter.

Prerequisites

— A command window is open and you have super user privileges.

— You are connected to the required MS.

Note: For details on how to view and modify PIB parameters, see Configuring PIB Parameters on page 8.

Name	Parameter Description	Value Range	Default Value	Parameter Type
asrnMaxPduSessionRecords	Maximum number of PDU Sessions supported per ASR-N session	1-50	25	Integer
asrnMaxDrbRecords	Maximum number of DRBs supported per ASR-N session	1-50	25	Integer

Reference List

[1] *ENM Basic Network Surveillance Interface Protocol Specification*1/155 19-cna 403 3034 Uen

[2] *ENM FM BNSI Northbound Interface Integration Programmers Guide*,4/198 17-aom 901 151 Uen

[3] *ENM Parameter List*, 1/190 59-aom 901 151 Uen

[4] *ENM Security System Administrator Guide*, 2/1543-aom 901 151 Uen

[5] *OSS Configuration for ENIQ Statistics*, 1/1546-aom 901 076 Uen

(Available from Ericsson Network IQ Statistics CPI Library)

[6] *ENM System Administrator Guide*, 1/1543-aom 901 151 Uen

[7] *ENM Configuration System Administrator Guide*, 1/1543-aom 901 151-1 Uen

[8] *Analytic Session Record (LTE) Specification*, 1551-cna 403 3416 Uen

(Available from node CPI Library)

[9] *ENM Identity and Access Management System Administrator Guide*, 2/1543-aom9010151-1 Uen

[10] *Key Performance Indicators and Formula Definitions for Autonomic Incident Management*, 10056-cna 403 3066 Uen