ERICSSON

User Guide, Operation and Maintenance

DESCRIPTION

9/1553-LZA 104 102/8 Uen B

Contents

1 Introduction

This chapter outlines the scope, structure and intended target groups of this document.

1.1 Scope

This document describes the Operation and Maintenance (O&M) functionality supported by the T12B software release for the SIU 02 (Site Integration Unit). For further information about this unit, refer to the Reference list.

The document covers fault, performance, configuration and software management. It also includes some general instructions, for example, how to read logs. The list of examples is not a complete list of actions possible, but is a selected set to give an idea and understanding of what can be done.

Note: In this document, where not explicitly mentioning IPv6, a description only applies for IPv4.

1.2 Target Group

The intended target groups for this document are:

- Radio Network Engineers.
- System Administrators.
- Field Technicians.
- Installation and Integration personnel.
- Site Engineers.

1.3 Main Changes

For information about the main changes in this Ericsson release, see Reference [1].

2 Capabilities

O&M of the SIU is controlled from the Operations Support System (OSS) or from a physical interface using the Command Line Interface (CLI). Example of a Graphical User Interface (GUI) for remote management is the Operations Support System Radio and Core (OSS-RC).

The following O&M functionality is available:

- Configuration Management (CM).
- Software Management (SWM).
- Fault and Alarm Management (FM).
- Performance Monitoring (PM).
- Dumps and Logs.

O&M via CLI of the CM, SM, and PM functionality supports IPv6 in addition to IPv4.

2.1 Users and Sessions

As a factory default, one user with no restrictions in access rights (a super user) and one regular user restricted with read-only permissions are defined. The super user (username *admin*) may login locally either via Telnet or Secure Shell (SSHv2) or remotely via SSH. The regular user (username *isp*) is restricted to SFTP sessions, see Section 2.1.3 on page 4.

2.1.1 Local Access

Local access is done via the local console (CONSOLE) port using a Local Maintenance Terminal (LMT) which can be a PC running a Telnet or SSH client.

The local console port is running a DHCP server which allows a LMT to detect IP settings automatically. The DHCP server is started automatically (together with the SSH and SFTP services) provided that the ports default configured network mask (255.255.255.0) has not been changed.

2.1.2 Remote Access

Remote access requires the SIU to be configured for remote operation and is done via an Ethernet port running SSH over IP.

2.1.3 SFTP File Transfer

The following Secure File Transfer Protocol (SFTP) functionality is available in the SIU:

- An SFTP client, used by O&M system functions to transfer files (for example downloading software archives and uploading PM data) from/to SFTP file servers at OSS.
- An SFTP server, used by the Automatic Data Collection (ADC) scripts to collect the In-Service Performance (ISP) log file from the node. The ADC scripts use the username *isp* to access the SFTP server.

For further information about management of users and sessions, see Reference [5].

3 Technical Description

This chapter provides a technical description of the O&M functionality.

3.1 Configuration Management

Configuration management is done using the Abis over IP Configuration Management (AIPCM) or OSS Common Explorer (CEX) applications in OSS together with the Network Element Support Server (NESS) and/or Network Element Distributed Support Server (NEDSS).

Configuration of the unit can also be done by downloading an XML file in a bulk CM session or executing CLI commands in a basic CM transaction. The Management Information Base (MIB) specifies the current instantiation of the Managed Information Model (MIM) in the unit. The MIM is a logical model of the SIU and describes Managed Object (MO) classes, their associations, attributes and operations.

The attribute *lastConfigChange* in MO **STN** can be used to display the SIU system time for the most recent configuration change. This value is updated in the MIB upon every successful **`commit`** or **`activate`** command.

The following MO classes are available:

- System created MOs;
  created with default values, some of which must be changed during installation.
- Manually created MOs;
  created, edited, or deleted using a bulk CM session or basic CM transaction.

For more information on MO classes and their attributes/values, see Reference [4].

3.1.1 Bulk CM Session

In a bulk CM session it is possible to:

- Backup and restore configuration.
- Upload or download configuration files.
- Activate new configurations.

A bulk CM session can be used in parallel with a software session, but not with a basic CM transaction. Only one bulk CM session can be active at a time. For commands available in a bulk CM session, see Reference [3].

3.1.2 Basic CM Transaction

In a basic CM transaction it is possible to:

- Retrieve information about transactions or MO hierarchies.
- Create and delete MOs.
- Set and retrieve MO attributes.
- Check validity and consistency of updated configurations (before they are committed).
- Commit (activate and persistently store) an updated configuration.

A basic CM transaction can be used in parallel with a software download session but not with a bulk CM session. Only one basic CM transaction can be active at a time. For commands available in a basic CM transaction, see Reference [3].

3.2 Software Management

The software management functionality includes the following:

- Software inventory; retrieval of software version information.
- Software upgrade; download of software packages from a Network Element Distributed Support Server (NEDSS).
- Software activation; installation and activation of downloaded software.

Software upgrade can be performed either using OSS and the Software Management Organizer (SMO) application or using CLI commands in a software session. For commands available in a software session, see Reference [3].

The figure below illustrates how SW archives and MIB files are used in different states:

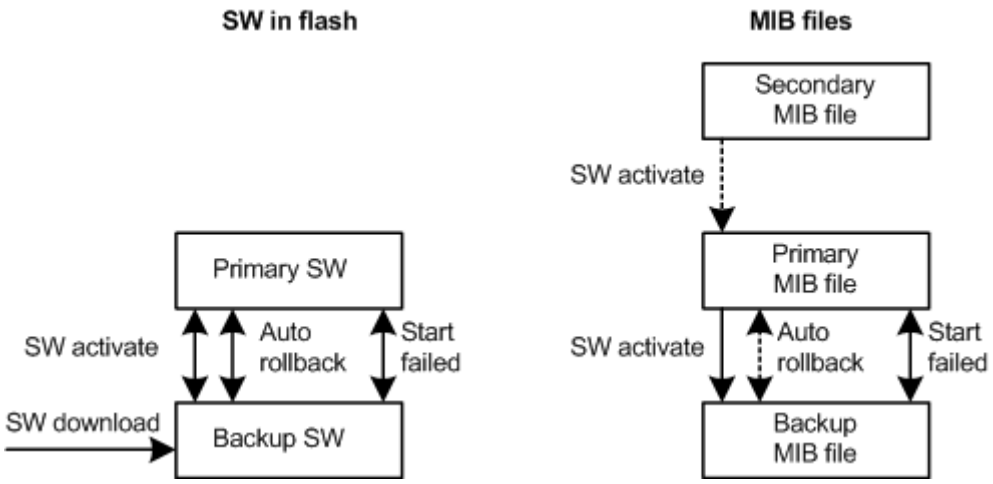

Figure 1 SW and MIB handling

The different states are described in the table below:

Table 1 Description of states for SW and MIB handling

State	Description
Normal start	At cold start of the SIU the Primary SW and the Primary MIB file are read into Random-Access Memory (RAM).
Start failed	If the SIU fails to start on the Primary SW, the SIU will copy the Backup SW and Backup MIB file to RAM and start using the backup versions instead. The alarm "Failed to Start Primary Software" will be raised. After a successful start on the Backup SW and MIB file, the Primary and Backup SW will be swapped as well as the Primary and Backup MIB file, that is SW stored as backup becomes primary.
SW download	At SW download the new SW overwrites the Backup SW.

State	Description
SW activate	At SW activation, the Primary MIB file overwrites the Backup MIB file and the Primary and Backup SW are swapped. This ensures that the previous operating SW and MIB are stored as backup if a start of the new SW fails. If there was a new configuration downloaded with the new SW, with delayed activation, the Secondary MIB file overwrites the Primary MIB file, ensuring that the new SW and the new configuration will be used at a restart of the unit.
Auto rollback	If no **`endSWSession`** command is received within the *autoRollbackTimer* minutes after SW activation, an auto rollback will be automatically initiated. At auto rollback, the Primary and Backup SW are swapped. If a new configuration was downloaded, also the Primary and Backup MIB files are swapped.

3.3 Alarm and Fault Management

Alarms are fault notifications issued by managed systems, which also issue alarm-cleared notifications after faults are resolved. Alarms are logged in the system log.

All alarms have their own Operating Instructions (OPI) document, see Reference [9] through Reference [32], describing the individual alarm and how to resolve what causes the alarm and get it ceased.

3.3.1 Alarm Subscriptions

Alarms are sent as SNMPv2c traps to the SNMP manager in the OSS. The SNMP manager is registered by creating an alarm subscription which must be defined before alarms can be sent. Ten concurrent subscriptions are possible, but each subscription requires a unique SNMP manager. Subscriptions are stored persistently and survive restarts. The list of current alarms is removed at a restart of the unit, regardless of the restart cause, and any persistent faults will be regarded as new alarms after the restart.

See document *SNMP Trap MIB Definition* for a description of the SNMP Trap MIB (Management Information Base).

3.3.2 SNMP Queries

The SIU only supports the "public" SNMP community string. The SIU does not support modifications of the configuration using SNMP.

3.3.3 Dumps

Dump files are created when a major software fault occurs that stops execution. The kernel will create a postmortem dump and store it in a designated dump area in memory. A *primary dump* is created the first time a serious error occurs. The subsequently following errors will be stored as a *secondary dump*. The dump files might be needed by Ericsson to investigate a problem and does not contain information of direct value for the operator.

Several dumps can be stored, where every dump has a unique number. The dump contains information about the process that was running when the serious fault was encountered. This is information like processor register, process stacks and information like global system load last statistics.

The dumps are stored in an area of RAM that is not cleared at cold restart. At power on the dumps are cleared.

3.3.4 Event Log

The SIU has a circular event log. The size of the circular log area is 5 MB which allows about 35000 log records to be written before wrap around occurs. Application software is allowed to write information about events detected in the system to the event log and each event is stored as a log record. Reading the log file is recommended to get clues when troubleshooting problems in the network. Log files might also be requested by Ericsson when investigating a reported problem.

The log is stored in an area of RAM that is not cleared at cold restart. At power on the log is cleared.

3.3.5 System Time

The system time in the SIU is started from the default value "1970-01-01 T00:00:00Z UTC". When a time server (NTP; Network Time Protocol or PTP; Precision Time Protocol) for system time has been defined and provides valid replies, the following happens:

- NTP: Set the system time based on the first time received from the time server.
- PTP over UDP: Sets the system time based on the first received Sync message negotiated with the time server.
- PTP over Ethernet: Sets the system time based on the first received Sync message with correct domain number (since no negotiation is required).
- Then, periodically adjusts the rate of system time so that the system time proceeds according to the time server.

The time server for system time could be the same, or different from, the time server defined for synchronization. When a time server for system time is not valid the system time is started from the default value and proceeds at the nominal rate based on the internal oscillator. When the time server for system time is defined with IP address "0.0.0.0" and the time server for synchronization has a valid address, the system time will be set/adjusted according to the synchronization time server (valid for NTP and PTP over UDP time servers).

The main purpose with time extracted by the SIU is to generate a system time, which is used by the alarm handling and as time stamps in log records and dumps.

The main purpose with frequency extracted by the SIU is to provide the radio parts of the Radio Base Stations (RBSs) with an accurate timing and frequency. See Reference [6].

3.4 Performance Management

The operator can subscribe to Performance Management (PM) data, that is all counters in all defined MO classes. The collected PM data is then compiled into files and pushed to a central storage, for example an NEDSS in OSS-RC.

For more information on MO classes and their counters, see Reference [4].

4 Engineering Guidelines

This chapter provides descriptions of how to perform some key O&M operations using CLI commands.

Examples of command lines are used in the procedures, some of which are longer than the page width of this document and require more than one line. Line break, enter, or carriage return characters must never be used when entering a command line.

4.1 Checking Software Versions

To check the current software versions, enter the command **rev**.

Example:

```
OSmon> rev
------------------ OSE modules -------------------------
oam.chk            Operation & Maintenance         R1M03
secmgrr.chk        Security Manager                R1F06
inetr.chk          MLPPP Daemon                    R1K01
ltp.chk            Local Terminal Port             R1K01
snc.chk            Synchronization                 R1J07
pd.chk             Packet Distributor              R1M01
cesopsn_pwr.chk    Circuit Emulation Service       R1A25
hdlc_pwr.chk       HDLC Pseudo-Wire Emulation      R1A12
p_relayr.chk       Packet Relay                    R1J07
profiler.chk       System Profiler                 R1F01
hwtest.chk         Hardware Tester                 R1K01
lcf_cp.chk         Local Connectivity Function     R1C01
bsp.drv            Board Support Package           R1M01
bootstrap.chk      Bootstrap                       R1M02
loader.drv         Software Loader                 P1M01
linuxload.drv      Linux Loader                    R1K01
------------------ Firmware modules --------------------
PBOOT              CXC 112 3777/1                  R1E01
------------------ Software archives -------------------
Primary:
OSE                CXP102138_1                     R1M03
Linux              -                               R1M03
Backup:
OSE                CXP102138_1                     R1M03
------------------ Active software ---------------------
OSE                                              Primary
Linux                                            PrimaryA
```

ny of the commands listed below can also be used to achieve the information:

Examples:

Check the product number and revision of the primary software.

```
OSmon> getMOAttribute STN=0 SW_PrimaryProductNumber
STN=0; SW_PrimaryProductNumber= CXP102138_1;
OperationSucceeded

OSmon> getMOAttribute STN=0 SW_PrimaryProductRevision
STN=0; SW_PrimaryProductRevision= R1M03;
OperationSucceeded
```

Check the product number and revision of the backup software.

```
OSmon> getMOAttribute STN=0 SW_BackupProductNumber
STN=0; SW_BackupProductNumber= CXP102138_1;
OperationSucceeded

OSmon> getMOAttribute STN=0 SW_BackupProductRevision
STN=0; SW_BackupProductRevision= R1M03;
OperationSucceeded
```

In case the system is executing on the wrong primary software, a system upgrade is recommended.

Note: When configuration and software has been verified, it is recommended to install the same software in both the primary and backup software archives. This is to avoid unwanted system behavior if an automatic rollback happens.

4.2 Upgrading Local System Software

The local system software is upgraded in a software session, which includes downloading the software to RAM and then activating it. A similar procedure (bulk CM session) can also be used to download and activate a bulk CM XML configuration file.

1. Connect to the CLI, either through a Local Maintenance Terminal (LMT) or remotely. For instructions, see "Accessing the SIU CLI" in Reference [8].

2. If connected through an LMT, enable SFTP on the local console port:

   Example:
   ```
   OSmon> uselocalsftp on
   ```

3. Check the product number of the unit.

   Example:
   ```
   OSmon> getMOAttribute STN=0,Equipment=0 productNumber
   STN=0,Equipment=0; productNumber= KDU 137 596/2;
   ```

   The output in this example; `KDU 137 596/2`, indicates it is a SIU 02.

4. Start a software session.

Example:
OSmon> ***startSWSession s1***

5. Download the new software by entering the SFTP URI, including authentication information, to the repository where it is stored. If an IPv6 address is used, it must be enclosed in square brackets.

   Note: The SIU 02 is the SFTP client and the SW to be downloaded should be fetched from an SFTP server. The SFTP server needs to be installed and run on the computer connected to the local console port.

   Note: When connecting through a LMT it is required to ensure that the SFTP traffic is not blocked by any firewalls resided between the SFTP client and server.

   Note: Ensure that a compatible software package is downloaded according to the product number achieved in step 3, that can for example be `SIU 02 T12A for KDU137596/2`.

   Examples:
   OSmon> ***downloadSW s1 sftp://jones:123abc@192.168.1.100/home/jones/SIUSW/siu_sw.tar***

   OSmon> ***downloadSW s1 sftp://jones:123abc@[2000:ffff:aaaa::abcd:1234]/home/jones/SIUSW/siu_sw.tar***

   Note: The file name of the software package must end with extension **.tar**.

   Note: The username "jones" and password "123abc" stated above are examples and should be replaced with your authentication information.

   Note: The IP addresses stated above are examples and should be replaced with the IP address to your SFTP server location.

   Note: Downloading the software takes approximately 1 minute and writing it to the flash takes about 5 minutes (depending on the CPU load and available bandwidth).

6. Check if the download is completed.

   Example:
   OSmon> ***getSWSessionStatus s1***

   Possible output from this command is: `Idle`, `DownLoadInProgress`, `DownloadFailed`, `DownloadCompleted`.

   If the message `DownloadFailed` is returned, make sure the following:

   - The SFTP server is running.
   - SFTP traffic to and from the SIU 02 is not blocked by any firewall.

- The SFTP URI is entered correct (username, password, path and file name).
- The IP address to the SFTP server is the one of your SFTP server.

If the message `DownLoadInProgress` is returned for more then 12 minutes, the download process should be terminated by issuing the command below.

Example:
`OSmon> endSWSession s1.`

Then, restart the session as described in step 4.

7. When the download is completed, activate the downloaded software.

   Example:
   `OSmon> activateSW s1`

   Note: This command closes any established traffic connections and restarts the unit on the new software. Activating the software takes about 1 minute and during this time the CLI is not available.

8. Log on again when the command prompt returns.

9. Check if the activation is completed.

   Example:
   `OSmon> getSWSessionStatus s1`

   Possible output from this command is: `ActivationInProgress`, `ActivationFailed`, `ActivationCompleted`.

10. When the activation is completed, terminate the software session.

    Example:
    `OSmon> endSWSession s1`

11. Check that the newly loaded SW is activated by entering the command ***rev***. The new SW should be stored in the primary software archive.

    Note: When configuration and software has been verified, it is recommended to install the same software in both the primary and backup software archives. This is to avoid unwanted system behavior if an automatic rollback happens.

12. If a bulk CM XML configuration file also should be downloaded, proceed to Section 4.3 on page 15. If not, terminate the Telnet session.

    Example:
    `OSmon> exit`

4.3 Downloading Configuration File

To download and activate a bulk CM XML configuration file, do the following:

1. Connect to the CLI, either through an LMT or remotely.
2. If connected through an LMT, enable SFTP on the console port:

   Example:
   OSmon> ***uselocalsftp on***
3. Start a bulk CM session:

   Example:
   OSmon> ***startSession bcm1***
4. Download the configuration file by entering the SFTP URI, including authentication information, to the repository where it is stored. If an IPv6 address is used, it must be enclosed in square brackets.

   Examples:
   OSmon> ***download bcm1 sftp://jones:123abc@192.168.1.100/home/jones/SIUCFG/siu82.xml***

   OSmon> ***download bcm1 sftp://jones:123abc@[2000:ffff:aaaa::abcd:1234]/home/jones/SIUCFG/siu82.xml***
5. Check if the download is completed.

   Example:
   OSmon> ***getSessionStatus bcm1***

   Possible output from this command is: `Idle`, `DownLoadInProgress`, `DownloadFailed`, `DownloadCompleted`.
6. When the download is completed, activate the downloaded configuration file.

   Example:
   OSmon> ***activate bcm1***

   Note: In case any changed attribute requires a restart, any established traffic connections are closed and the unit restarts. The restart also clears PM data and terminates O&M traffic in progress.
7. Log on again when the command prompt returns.
8. Check if the activation is completed.

   Example:
   OSmon> ***getSessionStatus bcm1***

Possible output from this command is: `ActivationInProgress`, `ActivationFailed`, `ActivationCompleted`.

9. When the activation is completed, terminate the session.

   Example:
   `OSmon>` ***`endSession bcm1`***

10. Terminate the Telnet session.

    Example:
    `OSmon>` **`exit`**

4.4 Resetting Configurations

It is not necessary to reset the configuration if it is already configured for the site. To read attribute values for a MO, use the following command: ***`getMOAttribute <MO-DN>`***.

For more information on definitions of MOs, attributes, and attribute values see Reference [4].

To reset the configuration to default factory settings, enter the command ***`resettofactorysetting`***.

A warning will appear asking for acknowledge if this command really should be executed.

If this command is executed it erases the MIB contents and clears persistent data including security and synchronization data. Software archives are not affected by this command.

Note: The factory default configuration for the SIU includes MO instances used by the auto integration process. When this command is executed these MO instances are created and the auto integration process is automatically started. To abort the auto integration process, all MO instances related to this function has to be manually removed. See Reference [8].

4.5 Retrieving Values from the MIB

See Reference [4] for a complete description of MO classes and their attributes/values.

1. Get the value of an attribute by using the ***`getMOAttribute`*** command and the `MO-DN` and the corresponding attribute to read.

   Example:
   `OSmon>` ***`getMOAttribute trans1 STN=0,IPInterface=0 primaryIP_Address`***
   `STN=0,IPInterface=0;primaryIP_Address=192.168.59.64`
   `OperationSucceeded`

When the command is executed within a basic CM transaction (between the commands **startTransaction** and **endTransaction**) the value returned is the value stored in RAM which might not be the same as the currently used value. If the value has been changed in the current transaction, the new value will be in use after execution of the **commit** command.

2. To read attribute values for a MO-DN currently in use (from the flash memory), use the command without transaction id.

   Example:

```
OSmon> getMOAttribute STN=0,EthernetInterface=0,LinkOAM=0
STN=0,EthernetInterface=0,LinkOAM=0; instanceId= 0;
STN=0,EthernetInterface=0,LinkOAM=0; activatedFeatures=
NONE;
STN=0,EthernetInterface=0,LinkOAM=0; discoveryState=
PASSIVE_WAIT;
STN=0,EthernetInterface=0,LinkOAM=0; dteMode= PASSIVE;
STN=0,EthernetInterface=0,LinkOAM=0; remoteDteInfo=
VSI:00 00 00 00 MAC:00:00:00:00:00:00;
STN=0,EthernetInterface=0,LinkOAM=0; maxPduRate= 10;
OperationSucceeded
```

3. Use the **-c** flag to also include counters for the specified MO-DN.

   Example:

```
OSmon> getMOAttribute STN=0,EthernetInterface=0,LinkOAM=0 -c
STN=0,EthernetInterface=0,LinkOAM=0; instanceId= 0;
STN=0,EthernetInterface=0,LinkOAM=0; activatedFeatures=
NONE;
STN=0,EthernetInterface=0,LinkOAM=0; discoveryState=
PASSIVE_WAIT;
STN=0,EthernetInterface=0,LinkOAM=0; dteMode= PASSIVE;
STN=0,EthernetInterface=0,LinkOAM=0; remoteDteInfo=
VSI:00 00 00 00 MAC:00:00:00:00:00:00;
STN=0,EthernetInterface=0,LinkOAM=0; maxPduRate= 10;
STN=0,EthernetInterface=0,LinkOAM=0; oamPduTransmitted=
0;
STN=0,EthernetInterface=0,LinkOAM=0; oamPduReceived= 0;
OperationSucceeded
```

4. To get the value of a specific attribute in all configured instances of a specific MO-DN, use **all** as *instanceId*.

   Example:

```
OSmon> getMOAttribute STN=0,IPInterface=all,MTU
STN=0,IPInterface=WAN; MTU= 1500;
STN=0,IPInterface=1,MTU= 1500;
OperationSucceeded
```

4.6 Changing Behavior through the MIB

See Reference [4] for a complete description of MO classes and their attributes/values.

1. Start a basic CM transaction.

   Example:
   OSmon> ***startTransaction trans1***

2. Set the attribute to change, by using the ***setMOAttribute*** command and the MO-DN and corresponding attribute to change.

   Example:
   OSmon> ***setMOAttribute trans1 STN=0 wakeUpRegistration 192.168.59.62***
   OperationSucceeded

3. Check the validity and consistency of the configuration.

   Example:
   OSmon> ***checkConsistency trans1***

4. The **commit** command is required to activate the configuration.

   Example:
   OSmon> ***commit trans1***

   Note: If attributes that require a restart are changed, the **commit** command is rejected and has to be reissued with the *forcedCommit* parameter.

   Example:
   OSmon> ***commit trans1 forcedCommit***

5. If the unit is restarted, this takes a few minutes and all traffic is closed during this time. In this case, reconnect and log on again.

6. End the transaction.

   Example:
   OSmon> ***endTransaction trans1***

4.7 Changing Behavior through Commands

See Reference [3] for available CLI commands.

1. Give a command.

   Example:
   OSmon> ***suspendPMMeasurements STN=0,MeasurementDefinition=0***
   OperationSucceeded

4.8 Changing Appearance of Default CLI Prompt

When using the CLI to administer multiple SIU nodes, it can be useful to include the node name to the default CLI prompt (OSmon>). The following attribute is used to change the appearance of the default CLI prompt:

Example:
```
OSmon> setMOAttribute STN=0 promptPrefix Node-x
OperationSucceeded
```

The prompt will then be "Node-x-OSmon>".

Note: During startup the prompt might be in its default form even if the above attribute has been set. It is likely there will be a short time before the whole MIB is processed and the prompt gets the new name.

4.9 Configuring System Clock Time Server

When a time server (NTP, PTP or PTP_ETH) is defined for system time, the SIU continuously keeps/tries to keep the system time in the same time as the time server.

When the time server for system time is defined with IP address "0.0.0.0" (default value) and a time server for synchronization has a valid address, the system time will be set/adjusted according to the synchronization time server (valid for NTP and PTP time servers).

Using PTP_ETH as a system clock time server requires that the SIU is configured to be synchronized using PTP_ETH. See Reference [6].

The following sub-sections describe how to explicitly configure a time server for synchronization of the SIU system clock.

4.9.1 NTP/PTP Time Server

Follow the steps below to configure an NTP or PTP time server to be used for synchronization of the SIU system clock:

1. Start a basic CM transaction.

   Example:
   ```
   OSmon> startTransaction trans1
   ```

2. Set the value to indicate if the system clock should be synchronized using a "NTP" or "PTP" time server. Default value is "NTP".

   Example:
   ```
   OSmon> setMOAttribute trans1 STN=0 systemClockTimeServerType
   PTP
   ```

3. If "NTP" was chosen as time server type, set the IP address to the time server and the local UDP port (default value: "123") to be used for synchronization of the system clock.

   Example:
   OSmon> ***setMOAttribute trans1 STN=0 systemClockTimeServer 192.168.62.92***
   OSmon> ***setMOAttribute trans1 STN=0 STN_systemClockUDP_Port 123***

4. If "PTP" was chosen as time server type, set the IP address to the time server and the number of the local and remote UDP port to be used for PTP general (default value: "320") and event (default value: "319") messages.

   Example:
   OSmon> ***setMOAttribute trans1 STN=0 systemClockTimeServer 192.168.62.92***
   OSmon> ***setMOAttribute trans1 STN=0 systemClockUDP_Port_General_PTP 320***
   OSmon> ***setMOAttribute trans1 STN=0 systemClockUDP_Port_Event_PTP 319***

5. Set the reference (dependency) to the IP interface to be used for synchronization of the system clock. The reference is either to an instance of MO **IPInterface** or **VirtualIPInterface** and is used for both system time communication and synchronization.

   Example:
   OSmon> ***setMOAttribute trans1 STN=0,Synchronization=0 depIP_Interface STN=0,IPInterface=0***

6. Set the DCSP value (default value: "0") for synchronization packets towards the time server. This attribute is used for both system time communication and synchronization.

   Example:
   OSmon> ***setMOAttribute trans1 STN=0,Synchronization=0 DSCP_synchronization 0***

   Note: When using a time server over satellite link this parameter shall be set to "46".

7. Check the validity and consistency of the configuration.

   Example:
   OSmon> ***checkConsistency trans1***

8. The **commit** command is required to activate the configuration.

   Example:
   OSmon> ***commit trans1***

Note: If attributes that require a restart are changed, the **commit** command is rejected and has to be reissued with the *forcedCommit* parameter.

Example:
OSmon> ***commit trans1 forcedCommit***

9. If the unit is restarted, this takes a few minutes and all traffic is closed during this time. In this case, reconnect and log on again.

10. End the transaction.

Example:
OSmon> ***endTransaction trans1***

4.9.2 PTP_ETH Time Server

Follow the steps below to explicitly configure a PTP_ETH time server to be used for synchronization of the SIU system clock:

1. Start a basic CM transaction.

Example:
OSmon> ***startTransaction trans1***

2. Ensure that "PTP_ETH" time server is configured as source for synchronization of the OCXO in the SIU. See Reference [6] for details.

3. Set the value to indicate that the system clock should be synchronized using a "PTP_ETH" time server. Default value is "NTP".

Example:
OSmon> ***setMOAttribute trans1 STN=0 systemClockTimeServerType PTP_ETH***

4. Set the PTP domain number for the system clock.

Example:
OSmon> ***setMOAttribute trans1 STN=0 systemClockPTPDomainNumber 0***

5. Check the validity and consistency of the configuration.

Example:
OSmon> ***checkConsistency trans1***

6. The **commit** command is required to activate the configuration.

Example:
OSmon> ***commit trans1***

Note: If attributes that require a restart are changed, the **commit** command is rejected and has to be reissued with the *forcedCommit* parameter.

Example:
OSmon> ***commit trans1 forcedCommit***

7. If the unit is restarted, this takes a few minutes and all traffic is closed during this time. In this case, reconnect and log on again.

8. End the transaction.

   Example:
   OSmon> ***endTransaction trans1***

4.10 Configuring Alarm Handling

A subscription must be defined before alarms and notifications can be sent. All alarms and notifications are sent to the SNMP manager in OSS defined by the IP address (*managerReference*) in the subscription.

Up to 10 active subscriptions is supported, each assigned a unique identity by the system, but there can only be one defined subscription for each IP address.

1. Check subscription IDs and status for active subscriptions:

   Examples:
   ```
   OSmon> getSubscriptionIds
   1
   OperationSucceeded

   OSmon> getSubscriptionStatus 1
   notSuspended 2 192.168.59.62
   OperationSucceeded
   ```

2. To define an additional subscription, do the following:

   Set the IP address (*managerReference* value) of the SNMP manager to which notifications are sent, and the time interval (*timeTick* value) between heartbeat notifications.

   Example:
   OSmon> ***subscribe 192.168.59.65 2***

   Note: To change the *timeTick* value for an existing subscription, an ***unsubscribe*** command needs to be performed followed by a subscription with the new value.

   Example:
   OSmon> ***unsubscribe 1***

3. Verify that subscriptions are sent:

Example:

```
OSmon> getMOAttribute STN=0 alarmSupervisionActive
STN=0; alarmsupervisionactive= true;
OperationSucceeded
```

If the value is "true", alarm supervision is active, that is, there is at least one active subscription.

4.11 Checking Alarms

To check the SNMP manager for active alarms do the following. Alternatively, check the log for alarms.

1. Enter the command:

   Example:

   ```
   OSmon> getalarmlist
   STN=0,TGTransport=p2;
   alarmId= 1;
   eventTime= 1970-01-01 T00:00:42;
   eventType= "communicationsAlarm(2)";
   perceivedSeverity= "critical(1)";
   ProbableCause= "m3100ConnectionEstablishmentError(22)";
   specificProblem= "STN-to-BSC Link Down";
   additionalText= "The Primary Control Connection to the BSC
                    is disconnected.";
   OperationSucceeded
   ```

Note: For a detailed description of what data is included in an alarm, see Section 4.12 on page 23. For more information on a specific alarm and the relevant actions to take, see Reference [9] through Reference [34].

4.12 Interpreting an Alarm

The following describes the data fields included in an alarm:

objectClass — The MO class that the alarm is mapped to.

objectInstance — Instance of the MO-DN issuing the alarm .

notificationId — Unique ID number for each notification in one notification category.

Notifications for `notifyNewAlarm`, `notifyClearedAlarm`, `heartbeatNotification` and `restartNotification` belong to the same category of notifications. For each new notification in the category, the selected ID number is increased by one.

eventTime — Date and Coordinated Universal Time (UTC) the alarm was triggered.

probableCause Probable cause of the alarm, as specified in ITU-T recommendations M3100, X.733 and X.736.

- x733Congestion(308).
- x733EquipmentMalfunction(315).
- x733LossOfFrame(328).
- x733LossOfSignal(329).
- x733OutOfMemory(332).
- x733PerformanceDegraded(334).
- x733PowerProblem(335).
- x733SoftwareError(346).
- x733VersionMismatch(357)
- x736OutOfService(347).
- x736AuthenticationFailure(600).
- m3100ConnectionEstablishmentError(22).
- m3100InvalidMessageReceived(23).
- m3100RemoteNodeTransmissionError(25).
- m3100HighTemperature(123).

perceivedSeverity

The severity of alarms are classified as follows:

- critical(1) – indicates that a service affecting condition has occurred and an immediate corrective action is required.
- major(2) – indicates that a service affecting condition has developed and urgent corrective action is required.
- minor(3) – indicates the existence of a non-service affecting fault condition and that corrective action should be taken in order to prevent a more serious (for example service affecting) fault later.
- warning(4) – indicates the detection of a potential or impending service affecting fault, before any significant effects have been felt.
- cleared(5) – indicates that the fault is no longer present.

alarmType Type of alarm according to ITU-T recommendations X.733 and X.736:

- communicationsAlarm(2).
- environmentalAlarm(3).
- equipmentAlarm(4).
- processingErrorAlarm(10).
- qualityOfServiceAlarm(11).
- securityServiceViolationAlarm(18).

specificProblem Short description of the specific problem. The specificProblem is also used to find the corresponding alarm OPI by title, see Reference [9] through Reference [34].

additionalText Used to supply extra information about the alarm. This alarm detail is not included in the `notifyClearedAlarm` notification.

alarmId Unique identifier for an individual alarm. It is the same for both the `notifyNewAlarm` and the `notifyClearedAlarm` notifications.

An example of the alarm details for the "E1/TI Loss of Signal" alarm is shown below:

```
objectClass           "E1/T1Interface"
objectInstance        "STN=0,E1T1Interface=1"
notificationId        5061
eventTime             833781600
probableCause         x733LossOfSignal(329)
perceivedSeverity     minor(3)
alarmType             communicationsAlarm(2)
specificProblem       "E1/T1 Loss of Signal"
additionalText        "LOS"
alarmId               22
```

The details indicate `STN=0,E1T1Interface=1` as the MO instance.

The *specificProblem* `"E1/T1 Loss of Signal"` is the slogan for the alarm and indicates that the "E1/T1 Loss of Signal" alarm OPI, Reference [10], must be used to resolve the alarm.

4.13 Configuring Performance Management

Follow the steps below to start the collection and transfer of PM data:

1. Stop any PM data reporting.

   Example:
   ```
   OSmon> suspendPMMeasurements STN=0,MeasurementDefinition=0
   OperationSucceeded
   ```

2. Start a basic CM transaction.

   Example:
   ```
   OSmon> startTransaction trans1
   ```

3. If needed, set the network element index to give the unit a unique identity in the PM reports.

   Example:
   ```
   OSmon> setMOAttribute trans1 STN=0,MeasurementDefinition=0
   neIndex 39
   ```

4. Check the validity and consistency of the configuration.

   Example:
   ```
   OSmon> checkConsistency trans1
   ```

5. The **`commit`** command is required to activate the configuration.

   Example:
   ```
   OSmon> commit trans1
   ```

Note: If attributes that require a restart are changed, the commit command is rejected and has to be reissued with the *forcedCommit* parameter.

Example:
OSmon> ***commit trans1 forcedCommit***

6. If the unit is restarted, this takes a few minutes and all traffic is closed during this time. In this case, reconnect and log on again.

7. End the transaction.

   Example:
   OSmon> ***endTransaction trans1***

8. The configuration for PM measurements are now done. The collection and transfer of PM data to OSS is started with the command **resumePMMeasurements**. Enter the SFTP URI, including authentication information, that defines where to upload the PM data. If an IPv6 address is used, it must be enclosed in square brackets.

   Examples:
   OSmon> ***resumePMMeasurements STN=0,MeasurementDefinition=0 useContainerFile sftp://jones:abc123@192.0.72.93/home/jones/ARCHIVE/SIU95***

   OSmon> ***resumePMMeasurements STN=0,MeasurementDefinition=0 useContainerFile sftp://jones:abc123@[2000:ffff:aaaa::abcd:1234]/home/jones/ARCHIVE/SIU95***

   Note: If the optional parameter *useContainerFile* is given, this indicates that the administrative file and all data files are combined into a container file before transfer. The file format for the container file is ".tar".

4.14 Managing the PM File Collection

In order to access PM data, the files need to be uploaded to a central storage via SFTP. To manage this in a controlled way, four types of PM files for each measurement report is created:

1. Administrative file - one for each Report Output Period.
2. Data file – one for each measured MO class for which there is data to report/Report Output Period.
3. Semaphore file - one for each file transfer session.
4. Container file - one for each file transfer session.

The uploaded PM files can include one or more gatherings of measurement data (one or more granularity periods). Data is reported for each created MO. The number of MOs reported for each granularity period can vary depending on

what MOs existed at the end of each granularity period. The number of data files uploaded each report period will vary due to varying number of MO classes.

The PM files are sent to the file server with a random offset on the time selected to avoid all units trying to upload files simultaneously. If the PM files cannot be uploaded, they are stored locally until the end of the next reporting period. The upload is then tried again and this is repeated until the files are uploaded or when the file is older than 24h. Locally stored files that have been uploaded or are older than 24h are deleted.

If the optional parameter *useContainertype* is given with command **`resumePMMeasurements`**, the administrative and data files are combined into a container file before transfer. The file format for the container file is ".tar". If a container file was never successfully transferred for a Report Output Period, the next Report Output Period files are appended before a new attempt is made, that is, a file transfer will always contain two files (container + semaphore).

4.14.1 File Storage Location

All files associated with a measurement report are pushed on the retrieving server into a structure in the directory given by the *fileStore* parameter set with command **`resumePMMeasurements`**. The administrative file and the data files (or the container .tar file) are stored in a subdirectory named "PM", and the semaphore file is stored directly at the directory given by the *fileStore* parameter. The directory structure with the subdirectory PM must be created manually on the `fileStore` before any files are uploaded.

4.14.2 Example of an Uploaded Measurement Report

This example illustrates how a measurement report looks like (shows only information about MO **STN** and **SuperChannel**):

```
MeasurementDefinition.filePrefix          STN
STN.STN_Name                              hostname
MeasurementDefinition.neIndex             1
Counter used for last NeKey               6
MeasurementDefinition.granularityPeriod   15 minutes
MeasurementDefinition.reportPeriod        60 minutes
First gathering starts                    Thu, 21 Apr 2005 06:45:00 U
InstanceId for MO STN                     0
    CPU Load for respective gathering     33%, 10% and 45% and 25%
    Failed PM uploads per gathering       3, 1, 4, and 2
InstanceId for MO TGTransport             TGSven
InstanceId for MO SuperChannel            3
    Downlink frames lost per gathering    7, 5, 8 and 6
    Downlink frames per gathering         300, 100, 400 and 250
    Uplink frames per gathering           99, 31, 50 and 29
```

Next NeKey = Unique number for each report period. (Calculated using the formula: 1 << 16 + 7 = 0x10007 = 65543).

Thu, 21 Apr 2005 06:45:00 UTC corresponds to 1114065900 seconds after 1 January 1970, which in turn corresponds to 18567765 minutes after 1 January 1970.

The above data creates the three files described in the following subsections.

4.14.3 Administrative File

The administrative file is named according to the following:
`STN_ADM_<StartTime>_<Granularity>.<Version>.bcp`

Identifiers placed inside <> are replaced according to the following:

StartTime — Minutes since 1970-01-01 00:00.00 GMT+0 as a decimal integer value, specifying the start time of the first measurement period in the report.

Granularity — The granularity period used for the gatherings of data, given by attribute *granularityPeriod* in MO **MeasurementDefinition**.

Version — 1.0.

Example:

File name: **STN_ADM_18567765_15.1.0.bcp**

```
65543,18567765,15,hostname,0
65544,18567770,15,hostname,0
65545,18567775,15,hostname,0
65546,18567810,15,hostname,0
```

The file contains one line for each gathering of measurement data:

```
<NeKey>, <StartTime>, <Granularity>, <Name>, <Status>
```

Identifiers placed inside <> are replaced according to the following:

NeKey — 32 bit unsigned decimal integer, of which the 16 Least Significant Bits (LSB) represent an internal counter, and the 16 Most Significant Bits (MSB) are the value of attribute *neIndex* in MO **MeasurementDefinition**:

Bit 0–15 is a unique identity generated by the SIU; and

Bit 16–31 is a unique identity of the SIU given by the attribute *neIndex* in MO **MeasurementDefinition**.

StartTime — Time in minutes since 1970-01-01 00:00.00 UTC, expressed as a decimal integer.

Granularity The granularity period given by attribute *granularityPeriod* in MO **MeasurementDefinition**.

Name The name of the network element as defined by attribute *STN_Name* in MO **STN**.

Status Will always be set to the integer value zero (0)

4.14.4 Data Files

The data files are named according to the following:
`STN_<MOClass>_<StartTime>_<Granularity>.<Version>.bcp`

Identifiers placed inside <> are replaced according to the following:

MOClass The name of the MO that the data file represents. For information about specific MO classes, see Reference [4].

StartTime Minutes since 1970-01-01 00:00.00 GMT+0 as a decimal integer value, specifying the start time of the first measurement period in the report.

Granularity The granularity period used for the gatherings of data, given by attribute *granularityPeriod* in MO **MeasurementDefinition**.

Version 1.0.

Example:

File name: **STN_STN_18567765_15.1.0.bcp**

```
65543,0,15,33,3
65544,0,15,10,1
65545,0,15,45,4
65546,0,15,25,2
```

File name: **STN_SuperChannel_18567765_15.1.0.bcp**

```
65543,TGSven;3,15,7,300,99
65544,TGSven;3,15,5,100,31
65545,TGSven;3,15,8,400,50
65546,TGSven;3,15,6,250,29
```

The data files contain one line for each gathering of measurement data for each MO instance of a specific MO class:

`<NeKey>, <InstanceId>, <Time>, <Counter1>, <Counter2>...`

Identifiers placed inside <> are replaced according to the following:

NeKey	Matches the NeKey used for the corresponding line in the administrative file.
InstanceId	A simplified MO-DN of the managed object (without class name) where the different levels are separated with ";" (excluding the top managed object **STN**). This gives the following syntax: <*instanceId* of the parent MO> ;<*instanceId* of the MO holding the counters>. And in the case of MO class **STN** or a child of MO class **STN**: <*instanceId* of the MO holding the counters>. MO class **STN** is identified with *InstanceId* = 0
Time	The actual measured time in minutes as an integer (Granularity Period)
Counter	A list of decimal integers representing the respective measured counters. The position of the counter in the list is according to the respective counter position given in Reference [4].

4.14.5 Container File

The container file is named according to the following:
`STN_ARC_<StartTime>_<Granularity>.<Version>.tar`

Identifiers placed inside <> are replaced according to the following:

StartTime	Minutes since 1970-01-01 00:00.00 GMT+0 as a decimal integer value, specifying the start time of the first measurement period in the report.
Granularity	The granularity period used for the gatherings of data, given by attribute *granularityPeriod* in MO **MeasurementDefinition**.
Version	1.0.

Example:

File name: **STN_ARC_18567765_15.1.0.tar**

4.14.6 Semaphore File

The semaphore file is empty and is always uploaded last, as a confirmation that all files have been successfully uploaded. Its presence indicates that the transfer of the other files are complete.

The semaphore file is named according to the following:
`sendfile_PM_<StartTime>`

Identifiers placed inside <> are replaced according to the following:

StartTime — Minutes since 1970-01-01 00:00.00 GMT+0 as a decimal integer value, specifying the start time of the first measurement period in the report.

Example:

File name: **sendfile_PM_18567765**

<empty>

4.15 Checking Counters

To check the counters for a specific MO-DN enter the following command:

Example:

```
OSmon> getcounters STN=0,EthernetInterface=3
============================================================
Counter                  | Value             | Change (2.6s)
============================================================
ifHCInOctets             | 785810115         | 1.17 Mbps
ifHCOutOctets            | 621870824         | 926.99 kbps
ifInErrors               | 0                 |
ifOutErrors              | 0                 |
ifHCInUcastPkts          | 654819            | +502
ifHCOutUcastPkts         | 517464            | +394
ifInDiscards             | 0                 |
ifOutDiscards            | 0                 |
ifHCInBroadcastPkts      | 13354             | +2
ifHCOutBroadcastPkts     | 13                |
ifHCInMulticastPkts      | 234               |
ifHCOutMulticastPkts     | 0                 |
ifInUnknownProtos        | 440               |
globalFrameDiscards      | 0                 |
============================================================
=====The output lists the counters with current value and a "Change" column
which indicates the delta for the counter since the previous execution of the
command. For counters that count octets, the "Change" column shows the
average value of the transmission speed during the period (2.6 seconds in
the example above).
```

4.16 Viewing Logs

Each log record stored in the event or system log contains the following information:

- Source file and line number.
- Process name.
- Time stamp.
- Type of event.
- Message string.

To read the log, use the following command:

`syslog <read|monitor> -s <system|event|messages>`

(for backwards compatibility reasons, the command **`log read`** and **`log monitor`** can be used as aliases).

Examples:

```
OSmon> syslog read -s event
```

This command prints log events to the terminal. The printing is aborted when the complete event log is written or with the key combination **Ctrl+c**.

```
OSmon> syslog monitor -s event
```

This command monitors ongoing events as they occur. The printing is aborted with the key combination **Ctrl+c**.

4.17 Dumps

Use the **`dump`** command with the **`-l`** flag to list all dump files saved in the node:

Example:

```
OSmon> dump -l
[PRIMARY DUMP ID 0x1]
user called : 1
error code  : 0x33844984
extra       : 0x00000000
restart reason :SW_UPGRADE

[PRIMARY DUMP ID 0x2]
user called : 1
error code  : 0x33844983
extra       : 0x00000000
restart reason :OAM Ordered RestartUse the dump command and add the dump file identity to print a specific dump:
```

Example:

```
OSmon> dump 0x1
[PRIMARY DUMP ID 0x1]

[ERROR HANDLER PARAMETERS]
user called : 1
error code  : 0x33844984
extra       : 0x00000000
...
...
...
[End Of Dump]
```

5 Concepts

Cold start

Start sequence where the entire RAM memory is cleared and all software is loaded from flash to RAM before started.

Warm start

Start sequence where the software is restarted without being reloaded from flash memory. The content of NVRAM sections and the event log is preserved.

Bulk CM file

A Bulk CM file specifies the configuration changes, delta information, that will be activated at command **`Activate`**. A Bulk CM file can cause MOs to be deleted and or created, and MO attributes to be changed.

Primary MIB file

At startup, the SIU reads configuration data from the Primary MIB. The Primary MIB file contains the configuration data that currently is, or will be, used by the resources.

Secondary MIB file

The Secondary MIB file is a second storage area in flash used when the Primary MIB file is being updated as part of a reconfiguration. The Secondary MIB file can either contain a new configuration or an old configuration depending on session and transaction state.

Backup MIB file

The Backup MIB file contains the MIB that shall be used together with the backup SW, in case reverting back to the backup SW is necessary.

All MIB files may also contain bulk CM download files as a result of function "Delayed activation".

Glossary

See Reference [2].

Reference List

General documents

[1] *Library Changes*

[2] *Glossary*

[3] *Command Descriptions*

[4] *Managed Object Model*

[5] *User Guide, Security*

[6] *User Guide, Synchronization*

[7] *SIU 02 Description*

[8] *SIU 02 Installation and Basic Configuration*

Alarm Operating Instructions (OPI) documents

[9] *Calibration Date Expired*

[10] *E1/T1 Loss of Signal*

[11] *E1/T1 Loss of Frame*

[12] *E1/T1 Alarm Indication Signal*

[13] *E1/T1 Remote Defect Indication*

[14] *Ethernet Interface Down*

[15] *Failed to Start Primary Software*

[16] *Hardware Fault*

[17] *Loss of Synchronization*

[18] *Multi Link PPP Bundle Down*

[19] *No Calibration from E1/T1*

[20] *No Calibration from Time Server*

[21] *PPP Link Down*

[22] *STN-to-BSC Link Down*

[23] *Temperature Outside Limits*

[24] *Transport Session Down*

[25] *Abis Local Connectivity License Mismatch*

[26] *Abis Local Connectivity License Blocked*

[27] *CES Loss of Connectivity*

[28] *Local Port Activation Attempt*

[29] *No Calibration from SynchE*

[30] *Service OAM*

[31] *E1/T1 Loopback Activation*

[32] *E1/T1 Unavailable Time*

[33] *Connectivity Fault*

[34] *OSPFv2 Authentication Failure*