ERICSSON

Command Descriptions

COMMAND DESCRIPTION

1/190 82-LZA 104 102/6 Uen E

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Contents

1	**Introduction**	**1**
1.1	Scope	1
1.2	Target Group	1
1.3	Main Changes	1
2	**General Format Definition**	**1**
2.1	Typographic Conventions	1
2.2	Command Line	2
2.3	Command Completion using <tab>	2
2.4	Invalid Keywords	3
2.5	Command Response	3
3	**Listing of Commands**	**3**
3.1	Generic Commands	3
3.2	Subscriptions	4
3.3	Basic CM	4
3.4	Bulk CM	5
3.5	Performance Management	5
3.6	SW Upgrade	6
4	**Command Descriptions**	**6**
4.1	AbortSessionOperation	6
4.2	AbortSWSessionOperation	7
4.3	Activate	8
4.4	ActivateSW	9
4.5	Backup	11
4.6	Caldate	12
4.7	Calexpdate	13
4.8	Calvalue	14
4.9	ChangePWD	14
4.10	CheckConsistency	15
4.11	ClearDynamicMACFwTable	16
4.12	Commit	17
4.13	CreateMO	18

4.14	DataEncryption	19
4.15	Debug	21
4.16	DeleteMO	21
4.17	Download	22
4.18	DownloadSW	24
4.19	Dump	25
4.20	EndSession	26
4.21	EndSWSession	27
4.22	EndTransaction	28
4.23	GenerateKey	29
4.24	GetAlarmList	29
4.25	GetContainment	30
4.26	GetCounters	32
4.27	GetIRPVersion	33
4.28	GetLinkOAM	34
4.29	GetMOAttribute	37
4.30	GetPendingConfiguration	40
4.31	GetSessionIds	41
4.32	GetSessionStatus	41
4.33	GetSubscriptionIds	42
4.34	GetSubscriptionStatus	43
4.35	GetSWSessionIds	44
4.36	GetSWSessionStatus	45
4.37	Gettime	46
4.38	GetTransactionId	47
4.39	GetTransactionStatus	47
4.40	Ping	48
4.41	Ping_Ethernet	49
4.42	ResetToFactorySetting	50
4.43	Restart	51
4.44	Restore	52
4.45	ResumePMMeasurements	54
4.46	ResumeSubscription	56
4.47	Rev	56
4.48	SetMOAttribute	57
4.49	Settime	59

4.50 ShowMACFwTable 59
4.51 SimplePing 60
4.52 StartSession 61
4.53 StartSWSession 62
4.54 StartTransaction 63
4.55 Subscribe 64
4.56 SuspendPMMeasurements 66
4.57 SuspendSubscription 66
4.58 Syslog 67
4.59 Timeservertest 68
4.60 TraceAllMOAttributeInLog 69
4.61 Traceroute 70
4.62 Traceroute_Ethernet 70
4.63 Unsubscribe 71
4.64 Upload 72
4.65 Uptime 74
4.66 UseLocalSFTP 74

5 Terminology 75

Glossary 77

Reference List 79

1 Introduction

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

1.1 Scope

This document describes the commands available in the Command Line Interface (CLI) of the SIU 02 (Site Integration Unit). For further information about this unit, refer to the Reference list.

Operation and Maintenance of the SIU is controlled from the Operations Support System (OSS), but defined CLI commands are also available via a Local Maintenance Terminal (LMT), see Reference [3].

Note: The generic command **`help`** lists commands of which some are not described in this document. These commands are not supported for customer use and must only be used by Ericsson personnel.

1.2 Target Group

The intended target groups for this document are:

- Radio Network Engineers.
- System Administrators.
- Network engineers.

1.3 Main Changes

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

2 General Format Definition

2.1 Typographic Conventions

The typographic conventions used in the CLI examples are as follows:

CommandToBeEnteredExaclyAsWritten ***InputParameterValuesSupplied ByTheUser***

```
Output parameter values displayed by the system
```

2.2 Command Line

Commands specified on the command line have the following syntax:

OSmon> **command** ***ParameterValue1 ParameterValue2 ...***

Entities on the command line are separated from each other by a space. All entities are case insensitive unless otherwise stated for the command. All entities are not used in all commands.

Parameter names are not used in the CLI. The values for parameters are given in a specific order. The order defines the parameter for which the value applies. The specific order of parameters for each command corresponds to the order of parameters in the tables of the command descriptions.

The maximum number of characters in the command line is 512.

2.3 Command Completion using <tab>

To display possible completions of a command, type the partial command followed by <tab>. If there are more than one potential match for the issued command, possible completions are displayed.

Example:
OSmon> **start**<tab>
startsession startswsession starttransaction

If the partially typed command uniquely identifies a command, the full command appears.

Tab completion can also be used on MO-DN.

Example:
OSmon> **getmoattribute** ***STN=0,E***<tab>
STN=0,E1T1Interface= STN=0,Equipment= STN=0,EthernetInterface=

The up and down arrow keys can be used to scroll through entered commands.

2.4 Invalid Keywords

The use of keywords is restricted in the parameters ***transactionId***, ***sessionId***, ***sessionIdSW***, ***instanceId*** and ***password***. These parameters cannot consist of the following restricted keywords:

- Managed Object (MO) names.
- Commands (including commands with the prefix test).
- Defined strings such as, forcedCommit, autoRollback, delayed, useNewConfiguration.
- Constant strings true, false, enable, disable (not case sensitive).

2.5 Command Response

When a command has been successfully received and syntax verified, a respond is given with the following status:

```
OperationSucceeded
```

Some commands result in a response with more parameters. The tables for each command show all output parameters.

3 Listing of Commands

In this section commands are grouped in different domains. Each domain has a subsection in which related commands are listed.

3.1 Generic Commands

This section lists generic commands related to the functionality and security.

- **caldate**
- **calexpdate**
- **calvalue**
- **clearMACFwTable**
- **changePWD**
- **dataEncryption**
- **debug**
- **dump**

- **generateKey**
- **getAlarmList**
- **getIRPVersion**
- **getLinkOAM**
- **gettime**
- **ping**
- **ping_ethernet**
- **resetToFactorySetting**
- **restart**
- **showMACFwTable**
- **simpleping**
- **rev**
- **settime**
- **syslog**
- **timeservertest**
- **traceallmoattributeinlog**
- **traceroute**
- **traceroute_ethernet**
- **uptime**
- **uselocalsftp**

3.2 Subscriptions

This section lists commands related to subscriptions to alarms and notifications. Subscribers are SNMP managers in network management systems such as the OSS.

- **subscribe**
- **unsubscribe**
- **getSubscriptionIds**
- **getSubscriptionStatus**
- **suspendSubscription**
- **resumeSubscription**

3.3 Basic CM

This section lists basic Configuration Management (CM) commands. Basic CM is used to change attribute values in the Managed Information Model (MIM) and to ensure that data changes are performed consistently.

Several attributes can be changed in a transaction by including a **setMOAttribute** command for each attribute to change. Each command performed in a transaction is validated immediately and executed when the **commit** command is received. The transaction is persistent at ordered commit. See Reference [4] for information about attribute values.

Basic CM commands:

- **checkConsistency**
- **getContainment**
- **getCounters**
- **startTransaction**
- **setMOAttribute**
- **getMOAttribute**
- **getTransactionId**
- **getTransactionStatus**
- **commit**
- **endTransaction**
- **createMO**
- **deleteMO**

3.4 Bulk CM

The commands listed in this section are related to the downloading and uploading of configuration files and the activation of new configurations. The configuration files hold configuration data for the Management Information Base (MIB). Files for download are generated by the OSS. Files are uploaded to be processed by the OSS.

- **startSession**
- **getSessionIds**
- **download**
- **upload**
- **backup**
- **restore**
- **getSessionStatus**
- **abortSessionOperation**
- **activate**
- **endSession**
- **getPendingConfiguration**

3.5 Performance Management

This section lists basic Performance Management (PM) commands used to collect performance data, for example, counters. The PM data is collected for each MO instance and periodically saved and uploaded to a file repository.

- **suspendPMMeasurements**
- **resumePMMeasurements**

3.6 SW Upgrade

The commands listed in this section are related to the downloading of software files and the activation of new software.

The downloading and the activation of software are performed in a software session.

- **startSWSession**
- **getSWSessionIds**
- **downloadSW**
- **getSWSessionStatus**
- **abortSWSessionOperation**
- **activateSW**
- **endSWSession**

4 Command Descriptions

This section describes the function of and parameters for each command on the CLI.

4.1 AbortSessionOperation

This command interrupts the uploading or downloading of bulk CM files. Any downloaded data is discarded. No uploaded data is deleted.

4.1.1 Input Parameters

Syntax: **abortsessionoperation** ***<sessionId>***

Table 1 Input Parameters

Parameter Name	Description
sessionId	Unique session identifier. Valid value: a string of up to 10 characters.

4.1.2 Output Parameters

Table 2 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.1.3 Example

```
OSmon> abortSessionOperation bcm1
OperationSucceeded
```

The *sessionId* is "bcm1".

4.2 AbortSWSessionOperation

This command interrupts software downloading. After the interruption the downloaded software is erased. The backup software stored is not affected by the download.

4.2.1 Input Parameters

Syntax: **abortSWSessionOperation** *<sessionIdSW>*

Table 3 Input Parameters

Parameter Name	Description
sessionIdSW	Unique session identifier. Valid value: a string of up to 10 characters.

4.2.2 Output Parameters

Table 4 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.2.3 Example

```
OSmon> abortSWSessionOperation s1
OperationSucceeded
```

The *sessionIdSW* is "s1".

4.3 Activate

This command activates requested changes of attribute values downloaded with the **download** command. In case any changed attributes require a restart, any established traffic connections are closed down and the unit restarts. The restart also clears PM data and terminates O&M traffic in progress.

A restart can take up to 1 minute and CLI sessions are terminated. A new login is required when the restart is completed.

4.3.1 Input Parameters

Syntax: **activate** ***<sessionId> [delayed] [autoRollback <autoRollbackTimer>]***

Table 5 Input Parameters

Parameter Name	Description
sessionId	Unique session identifier. Valid value: a string of up to 10 characters.
delayed (optional)	If set, the downloaded file is only stored persistently. The activation of the downloaded file will be performed on a system created MIB, when the SWSession is activated with the optional parameter *useNewConfiguration* set. If not set, the downloaded file will be activated immediately. The parameter is not possible to use in the same session as the parameter *autoRollback*. **Note:** System created MIB implies that the existing configuration is lost and that the configuration file must recreate all (but system created) information.
autoRollback <autoRollbackTimer> (optional)	If set, the unit will restart on the old configuration if not **endSession** or **endSWsession** have been received within the specified time (autoRollbackTimer) in minutes.

4.3.2 Output Parameters

Table 6 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.3.3 Example 1

```
OSmon> activate bcm1
OperationSucceeded
```

4.3.4 Example 2

```
OSmon> activate bcm1 delayed
OperationSucceeded
```

The *sessionId* is "bcm1"
The optional *delayed* parameter is included to ensure the activation of the downloaded file should be performed within a SWSession with the optional parameter *useNewConfiguration* set.

4.3.5 Example 3

```
OSmon> activate bcm1 autoRollback 10
OperationSucceeded
```

The *sessionId* is "bcm1"
The optional *autoRollback* parameter is included to restart the unit on the old configuration, if not **`endSession`** or **`endSWSession`** have been received before the *autoRollbackTimer*, with the value "10" (minutes), has expired.

4.4 ActivateSW

This command closes any established traffic connections and restarts the unit with the latest downloaded software. The restart also clears PM data and terminates O&M traffic and Telnet/SSH sessions in progress.

A restart can take up to 1 minute and CLI sessions are terminated. A new login is required when the restart is completed.

4.4.1 Input Parameters

Syntax: **activateSW** ***<sessionIdSW> [useNewConfiguration] [autoRollback <autoRollbackTimer>]***

Table 7 Input Parameters

Parameter Name	Description
sessionIdSW	Unique session identifier. Valid value: a string of up to 10 characters.
useNewConfiguration (optional)	If set, both the latest downloaded software and Bulk CM file will be activated. The requirement is that a Bulk CM session exists in ACTIVATION_IN_PROGRESS state and that it has been entered through the **activate** or **restore** command and with the parameter *delayed* set. If the requirement is not fulfilled, the **activateSW** command is rejected. If not set, only the latest downloaded software is activated after a restart.
autoRollback <autoRollbackTimer> (optional)	If set, the unit will restart on the old software if not **endSession** or **endSWSession** have been received within the specified time (*autoRollbackTimer*) in minutes.

4.4.2 Output Parameters

Table 8 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.4.3 Example 1

```
OSmon> activateSW s1
OperationSucceeded
```

The *sessionIdSW* is "s1".

4.4.4 Example 2

```
OSmon> activateSW s1 useNewConfiguration
OperationSucceeded
```

The *sessionIdSW* is "s1".
The optional parameter *useNewConfiguration* is included to ensure that both the latest software and Bulk CM File should be activated.

4.4.5 Example 3

```
OSmon> activateSW s1 autoRollback 10
OperationSucceeded
```

The *sessionIdSW* is "s1".
The optional parameter *autoRollback* is included to restart the unit on the old software, if not **endSession** or **endSWSession** have been received before the *autoRollbackTimer*, with the value "10" (minutes), has expired.

4.5 Backup

This command is used to backup a configuration file. During backup the configuration is encrypted with the provided password and uploaded to the specified SFTP path. Only MIM configuration is included into the backup (no PBOOT parameters or software). The configuration includes secure data (like passwords).

4.5.1 Input Parameters

Syntax: **backup** ***<sessionId> <FileRef> <passwd> [<progressInterv>]***

Table 9 Input Parameters

Parameter Name	Description
sessionId	Unique session identifier. Valid value: a string of up to 10 characters.
FileRef	Specifies a globally unique destination for the upload. The addressing of the file source must follow the syntax: "sftp://username:password@a.b.c.d/filepath" where "a.b.c.d" specifies the IP address of the destination host, and "filepath" specifies the directory structure and filename.

Parameter Name	Description
passwd	The password to be used for encryption of the configuration file. The same password is used when performing a **`restore`** of the configuration.
progressInterv (optional)	Defines a time in seconds after which a software download progress message should be sent to the SNMP manager. Valid values: 0 and 15–900. When set to "0" or not set, no notifications will be sent.

4.5.2 Output Parameters

Table 10 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.5.3 Example

```
OSmon> backup bcm1 sftp://smith:123abc@192.168.69.152/STNCFG/stn
82_CM.xml admin 30
OperationSucceeded
```

The *sessionId* is "bcm1".
The *FileRef* is "sftp://smith:123abc@192.168.69.152/STNCFG/stn82_CM.xml"
The *passwd* is "admin".
The optional parameter *progressInterv* is "30".

4.6 Caldate

This command displays or sets the calibration date.

4.6.1 Input Parameters

Syntax: **`caldate`** ***`[<(YYYYMMDD)>]`***

4.6.2 Output Parameters

Table 11 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.6.3 Example

```
OSmon> caldate
20100924
OperationSucceeded
```

4.7 Calexpdate

This command displays or sets the calibration expire date.

4.7.1 Input Parameters

Syntax: **`calexpdate`** ***`[<(YYYYMMDD)>]`***

4.7.2 Output Parameters

Table 12 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.7.3 Example

```
OSmon> calexpdate
20110924
OperationSucceeded
```

4.8 Calvalue

This command displays or sets the calibration value.

4.8.1 Input Parameters

Syntax: **calvalue** ***<(value)>***

4.8.2 Output Parameters

Table 13 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.8.3 Example

```
OSmon> calvalue
156
OperationSucceeded
```

4.9 ChangePWD

This command changes the password for a user. Only one default super user is supported with username **admin**.

The password is used when logging on to the CLI through a Telnet/SSH (local access) or SSH (remote access) connection.

The password is case sensitive and can consist of 1–100 characters. Allowed characters are A–Z, a–z, and 0–9.

4.9.1 Input Parameters

Syntax: **changePWD** ***<old_pass>*** ***<new_pass>*** ***<ver_pass>***

Table 14 Input Parameters

Parameter Name	Description
old_pass	Password currently valid for the "admin" user.

Parameter Name	Description
new_pass	New password for the "admin" user.
ver_pass	New password duplicated for verification.

4.9.2 Output Parameters

Table 15 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.9.3 Example

```
OSmon> changePWD hidden xyZ6789abc xyZ6789abc
OperationSucceeded
```

The *old_pass* is "hidden".
The *new_pass* and *ver_pass* is "xyZ6789abc".

4.10 CheckConsistency

This command checks the validity and consistency of an updated configuration. The check is useful before a **`commit`** command is performed to verify that the configuration contains a complete set of MOs and attributes.

4.10.1 Input Parameters

Syntax: **`checkConsistency`** ***`<transactionId>`***

Table 16 Input Parameters

Parameter Name	Description
transactionId	Unique transaction identifier. Valid value: a string of up to 10 characters.

4.10.2 Output Parameters

Table 17 Output Parameters

Parameter Name	Description
consistent	This parameter contains the result of the consistency check. The result is `consistent` or `not consistent`.
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.10.3 Example

```
OSmon> checkConsistency trans1
consistent
OperationSucceeded
```

The *transactionId* is "trans1".

4.11 ClearDynamicMACFwTable

This command removes entries in the MAC forwarding table.

4.11.1 Input Parameters

Syntax: **clearDynamicMACFwTable** ***all*** | ***<EthernetInterface> [<tagValue>]***

Table 18 Input Parameters

Parameter Name	Description
all	Specifies that all entries in the MAC forwarding table should be removed.
EthernetInterface (optional)	If this optional parameter is set, only entries in the MAC forwarding table of the specified MO **EthernetInterface** instance is removed.
tagValue (optional)	Specifies the tag value (VLAN Id) of table entries to remove. This parameter is optional and can only be used with the *EthernetInterface* option.

4.11.2 Output Parameters

None

4.11.3 Examples

OSmon> **clearDynamicMACFwTable** ***all***
This command removes all entries in the MAC forwarding table.

OSmon> **clearDynamicMACFwTable** ***STN=0,EthernetInterface=0***
This command removes all MAC forwarding table entries for the specified MO **EthernetInterface** instance.

OSmon> **clearDynamicMACFwTable** ***STN=0,EthernetInterface=0 200***
This command removes MAC forwarding table entries for the specified MO **EthernetInterface** instance with the specified *tagValue*.

4.12 Commit

This command commits all configuration requests in the transaction. The requests are performed with one or more **setMOAttribute**, **createMO**, and **deleteMO** commands issued before **commit**. Data regarding the transaction are stored persistently.

4.12.1 Input Parameters

Syntax: **commit** ***<transactionId> [forcedCommit] [autoRollback <autoRollbackTimer>]***

Table 19 Input Parameters

Parameter Name	Description
transactionId	Unique transaction identifier. Valid value: a string of up to 10 characters.
forcedCommit (optional)	If set , the unit restarts if attributes requiring a restart are changed. All traffic is closed down during the restart. If not set, and attributes requiring a restart are changed, the **commit** command is rejected.
autoRollback <autoRollbackTimer> (optional)	If set, the unit will restart on the old configuration if **endTransaction** has not been received within the specified time (autoRollbackTimer) in minutes.

4.12.2 Output Parameters

Table 20 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.12.3 Example 1

```
OSmon> commit trans1 forcedCommit
OperationSucceeded
```

The *transactionId* is "trans1".
The optional parameter *forcedCommit* is included if attributes requiring a restart are changed.

4.12.4 Example 2

```
OSmon> commit trans1 autoRollback 10
OperationSucceeded
```

The *transactionId* is "trans1".
The optional parameter *autoRollback* is included to restart the unit on the old configuration, if the **endTransaction** command has not been received before the *autoRollbackTimer*, with the value "10" (minutes), has expired.

4.13 CreateMO

This command creates a new MO instance. The command must be used within a transaction and does not support setting of attribute(s). The new MO instance will not take affect until a **commit** command is executed.

For further information about MO classes, see Reference [4].

4.13.1 Input Parameters

Syntax: **createMO** *<transactionId> <MO-DN>*

Table 21 Input Parameters

Parameter Name	Description
transactionId	Unique transaction identifier. Valid value: a string of up to 10 characters.
MO-DN	Base object in a containment hierarchy for the new MO instance.

4.13.2 Output Parameters

Table 22 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.13.3 Example

```
OSmon> createMO trans1 STN=0,E1T1Interface=0
OperationSucceeded
```

The *transactionId* is "trans1".
The *MO-DN* is "STN=0,E1T1Interface=0".

4.14 DataEncryption

This command activates encryption of sensitive configuration data such as passwords and encryption keys when stored on the node in configuration files. If encryption has been enabled it can only be disabled by having the configuration reset to factory settings. See Section 4.42 on page 50

4.14.1 Input Parameters

Syntax: **dataencryption** *<on/print>*

Table 23 Input Parameters

Parameter Name	Description
on	This option will activate encryption of sensitive data if not already activated.
print	This option will return actual state - `activated` or `notActivated` - of encryption of sensitive data.

4.14.2 Output Parameters

Table 24 Output Parameters

Parameter Name	Description
encryptionstatus	This parameter is only relevant for the print option. It indicates either of the following: (a) Encryption of sensitive data activated. (b) Encryption of sensitive data not activated.
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.14.3 Example 1

```
OSmon> dataEncryption on
```

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

```
Warning:  All configuration files will be encrypted,
backup and secondary configuration to be removed, continue
(ok/cancel)?
OperationSucceeded
```

4.14.4 Example 2

```
OSmon> dataEncryption print
notActivated
OperationSucceeded
```

4.15 Debug

This command is used to enable the debug mode of the CLI.

4.15.1 Input Parameters

Syntax: **debug** ***[on|off]***

Table 25 Input Parameters

Parameter Name	Description
on	Enables debug mode of the CLI.
off	Disables debug mode of the CLI.

4.15.2 Output Parameters

None.

4.15.3 Example

`OSmon>` **debug** ***on***
Once executed with "on" option, the command prompt changes to:
`DBG:OSmon>`

debug ***off***
Once executed with "off" option, the command prompt changes back to normal mode:
`OSmon>`

4.16 DeleteMO

This command deletes a MO instance. It is only possible to delete manually created MO instances. Child MOs are automatically deleted. The deletion of the MO instances will not take affect until a **commit** command is executed.

For further information about MO classes, see Reference [4].

4.16.1 Input Parameters

Syntax: **deleteMO** ***<transactionId>*** ***<MO-DN>***

Table 26 Input Parameters

Parameter Name	Description
transactionId	Unique transaction identifier. Valid value: a string of up to 10 characters.
MO-DN	Base object in a containment hierarchy from where the deletion will be performed.

4.16.2 Output Parameters

Table 27 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.16.3 Example

```
OSmon> deleteMO trans1 STN=0,E1T1Interface=0
OperationSucceeded
```

The *transactionId* is "trans1"
The *MO-DN* is "STN=0,E1T1Interface=0".

4.17 Download

This command orders the downloading of a bulk CM XML configuration file. When the complete file has been downloaded, the requested attribute values are stored in RAM.

All new attribute values that have not been stored in flash with the command **`activate`** or **`endSession`** are lost at power off, spontaneous restart, push button reset, or any command causing a reset.

4.17.1 Input Parameters

Syntax: **`download`** ***`<sessionId> <FileRef> [<progressInterv>]`***

Table 28 Input Parameters

Parameter Name	Description
sessionId	Unique session identifier. Valid value: a string of up to 10 characters.
FileRef	Specifies a globally unique file reference from which the data is retrieved and downloaded. The addressing of the file source must follow the syntax: "sftp://username:password@a.b.c.d/filepath" where "a.b.c.d" specifies the IP address of the file source host and "filepath" specifies the directory structure and filename.
progressInterv (optional)	Defines a time (in seconds) after which a software download progress message should be sent to the SNMP manager. Valid values: 0 and 15–900. When set to "0" or not set, no notifications will be sent.

4.17.2 Output Parameters

Table 29 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.17.3 Example

```
OSmon> download bcm1 sftp://smith:123abc@192.168.69.152/STNCFG/stn82_synch.xml 15
OperationSucceeded
```

The *sessionId* is "bcm1".
The *FileRef* is "sftp://smith:123abc@192.168.69.152/STNCFG/stn82_synch.xml"
The optional parameter *progressInterv* is "15" (seconds).

4.18 DownloadSW

This command orders the downloading of new software. The downloaded software is stored in the RAM until it is activated with the **activateSW** command or until the session is ended with the **endSWSession** command. Any restart of the unit or abortion with the **abortSWSessionOperation** command during software downloading, erases the downloaded software.

4.18.1 Input Parameters

Syntax: **downloadSW** ***<sessionIdSW> <FileRef> [<progressInterv>]***

Table 30 Input Parameters

Parameter Name	Description
sessionIdSW	Unique session identifier. Valid value: a string of up to 10 characters.
FileRef	Specifies a globally unique file reference from which the data is retrieved and downloaded. The addressing of the file source must follow the syntax: sftp://username:password@a.b.c.d/filepath where "a.b.c.d" specifies the IP address of the file source host and "filepath" specifies the directory structure and filename.
progressInterv (optional)	Defines a time (in seconds) after which a software download progress message should be sent to the SNMP manager. Valid values: 0 and 15–900 When set to "0" or not set, no notifications will be sent.

4.18.2 Output Parameters

Table 31 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.18.3 Example

OSmon> **downloadSW** ***s1 sftp://smith:123abc@192.168.69.152/STNSW/STN_sw.tar.gz 30***
OperationSucceeded

The *sessionIdSW* is "s1".
The *FileRef* is "sftp://smith:123abc@192.168.69.152/STNSW/stn_sw.tar.gz"
The optional parameter *progressInterv* is "30" (seconds).

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).

4.19 Dump

This command reads the most recent restart dump. Restart dumps are created whenever a board restart is made as a result of a suspected fault. A primary dump is created whenever the system restarts during stable operation. Secondary dumps are created if more restarts are encountered before the system has again achieved a state of stable operation.

4.19.1 Input Parameters

Syntax: **dump** ***[-l] [<dump-number>]***

Table 32 Input Parameters

Parameter Name	Description
-l	The -l flag causes a list of available dumps to be produced so that a specific dump-number can be displayed in a subsequent command.
dump-number	The number as presented by **dump -l** to be displayed.

4.19.2 Output Parameters

None.

4.19.3 Example 1

OSmon> **dump**
Shows the most recent primary dump and the last subsequent secondary dump (if any).

4.19.4 Example 2

```
OSmon> dump -1
[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 Restart
```

Shows a list on all primary and secondary dump files saved in the node.

4.19.5 Example 3

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

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

Shows the requested dump.

4.20 EndSession

This command ends a bulk CM session. Operations in progress related to the bulk CM session are terminated.

4.20.1 Input Parameters

Syntax: **endsession** ***<sessionId>***

Table 33 Input Parameters

Parameter Name	Description
sessionId	Unique session identifier. Valid value: a string of up to 10 characters.

4.20.2 Output Parameters

Table 34 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.20.3 Example

```
OSmon> endSession bcm1
OperationSucceeded
```

The *sessionId* is "bcm1".

4.21 EndSWSession

This command ends a software upgrading or downgrading session. Operations in progress related to the software session are terminated. Downloaded software remains in flash regardless of whether it has been activated or not.

4.21.1 Input Parameters

Syntax: **endSWSession** *<sessionIdSW>*

Table 35 Input Parameters

Parameter Name	Description
sessionIdSW	Unique session identifier. Valid value: a string of up to 10 characters.

4.21.2 Output Parameters

Table 36 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.21.3 Example

```
OSmon> endSWSession s1
OperationSucceeded
```

The *sessionIdSW* is "s1".

4.22 EndTransaction

This command ends a basic CM transaction.

Note: If the command is performed before a **commit** command, all changes in the transaction are lost.

4.22.1 Input Parameters

Syntax: **endTransaction** *<transactionId>*

Table 37 Input Parameters

Parameter Name	Description
transactionId	Unique transaction identifier. Valid value: a string of up to 10 characters.

4.22.2 Output Parameters

Table 38 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.22.3 Example

```
OSmon> endTransaction trans1
OperationSucceeded
```

The *transactionId* is "trans1".

4.23 GenerateKey

This command generates a new host key, which is used for security and encrypts the communication with clients. Clients using SSH need the new host key to connect to the SIU and at the point in time when the host key is changed in the SIU, the clients need to be configured to change host key on request. To maintain IP security this command must be executed at least once a year.

4.23.1 Input Parameters

None.

4.23.2 Output Parameters

Table 39 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.23.3 Example

```
OSmon> generateKey
OperationSucceeded
```

4.24 GetAlarmList

This command returns a list of all active alarms.

4.24.1 Input Parameters

None.

4.24.2 Output Parameters

Table 40 Output Parameters

Parameter Name	Description
alarmInformationList	This parameter contains a list of all active alarms.
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.24.3 Example

```
OSmon> getalarmlist
STN=0,EthernetInterface=0;
alarmId= 1;
eventTime= 1970-01-01 T00:00:42;
eventType= "communicationsAlarm(2)";
perceivedSeverity= "major(2)";
probableCause= "x733LossOfSignal(329)";
specificProblem= "Ethernet Interface Down";
additionalText= "Ethernet Interface Link Down - physical
port 0";
OperationSucceeded
```

4.25 GetContainment

This command retrieves all MO instances in a specific MO containment hierarchy under (and including) the specified *MO-DN*.

4.25.1 Input Parameters

Syntax: **getContainment** ***[<transactionId>] <MO-DN>***

Table 41 Input Parameters

Parameter Name	Description
transactionId (optional)	Unique transaction identifier. Valid value: a string of up to 10 characters. Optional parameter when the command is executed in a transaction (between the commands **`startTransaction`** and **`endTransaction`**. The value returned when used in a transaction is the value stored in the RAM which might not be the same as the currently used value. Changing MO attributes with the **`setMOAttribute`** command creates a temporary copy of the MO model in the RAM. The values in the RAM are used after the next **`commit`** command. This parameter is not used when the command is executed outside a transaction. The returned value will in this case be the currently used value stored in the flash memory.
MO-DN	MO instance in a containment hierarchy.

4.25.2 Output Parameters

Table 42 Output Parameters

Parameter Name	Description
containment	All MO instances included in the specified containment hierarchy including the MO defined with the *MO-DN* parameter.
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.25.3 Example

```
OSmon> getContainment trans1 STN=0

STN=0;
STN=0,Equipment=0;
STN=0,EthernetInterface=0;
STN=0,IPInterface=0;
```

```
STN=0,Synchronization=0;
STN=0,MeasurementDefinition=0;
STN=0,E1T1Interface=0;

OperationSucceeded
```

The optional parameter *transactionId* is "trans1".
The *MO-DN* is "STN=0".

4.26 GetCounters

This command displays counters for the specified MO.

4.26.1 Input Parameters

Syntax: **getcounters** ***<MO-DN>***

Table 43 Input Parameters

Parameter Name	Description
MO-DN	The MO instance that is read.

4.26.2 Output Parameters

Table 44 Output Parameters

Parameter Name	Description
Counter	The counter name.
Value	The value of the counter.
Change	The Change column 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 below).

4.26.3 Example

OSmon> **getcounters** ***STN=0,EthernetInterface=5***

```
============================================================
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            |
=========================================================
```

4.27 GetIRPVersion

This command displays the Integration Reference Point (IRP) version used. The IRP version specifies supported versions of the OSS - SIU O&M interface.

4.27.1 Input Parameters

None.

4.27.2 Output Parameters

Table 45 Output Parameters

Parameter Name	Description
currentVersionNumber	Shows the IRP version currently used.
versionNumberSet	List of IRP versions supported by the SIU. Ericsson's *versionNumberSet* attribute behavior differs from that specified in the 3GPP specification.
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.27.3 Example

```
OSmon> getIRPVersion
5.5 5.5,5.2,5.1,5.0,3.6,3.5,3.4,3.2,3.1,3.0
OperationSucceeded
```

The *currentVersionNumber* is "5.5" (leftmost).

The *versionNumberSet* is "5.5,5.2,5.1,5.0,3.6,3.5,3.4,3.3,3.2,3.1,3.0".

The IRP number is included in the bulk CM configuration file that is uploaded/downloaded from/to the SIU to/from OSS. This means that the SIU accepts bulk CM downloads with configurations having any of the versions specified in the *versionNumberSet*. All bulk CM configurations exported by the SIU as a result of the **upload** command are using the *currentVersionNumber* (the highest supported IRP version in that software version).

4.28 GetLinkOAM

This command displays LinkOAM discovery state for local and remote DTE as well as LinkOAM interface status and statistics.

4.28.1 Input Parameters

Syntax: **getlinkoam** ***discoverystate*** | ***interfacestatus*** | ***statistics*** ***<MO-DN>***

Table 46 Input Parameters

Parameter Name	Description
discoverystate	Displays LinkOAM discovery state for local and remote DTE for the specified MO instance.
interfacestatus	Displays LinkOAM interface status for the specified MO instance.
statistics	Displays LinkOAM statistics for the specified MO instance.
MO-DN	The MO instance that is read.

4.28.2 Output Parameters

Table 47 Output Parameters

Parameter Name	Description
discoverystate	Discovery mode (ACTIVE / PASSIVE). Discovery state (FAULT_STATE / ACTIVE_SEND_LOCAL / PASSIVE_WAIT / SEND_LOCAL_REMOTE / SEND_LOCAL_REMOTE_OK / SEND_ANY). Unidirection mode (SUPPORTED / NOT SUPPORTED). Remote loopback (SUPPORTED / NOT SUPPORTED). Remote DTE MAC address. Remote DTE vendor oui.
interfacestatus	See example.
statistics	Counters; the counter name and value. Local event logs; value and event name. Remote event logs; value and event name.

4.28.3 Examples

```
OSmon> getlinkoam discoverystate STN=0,EthernetInterface=0,LinkOAM=0

============================================================
Discovery state for local and remote DTE
============================================================
Local:
 Mode:                              ACTIVE
 State:                             SEND_ANY
 Unidirection:                      NOT SUPPORTED
 Remote loopback:                   NOT SUPPORTED
============================================================
Remote:
  MAC address:                      00:00:00:05:01:21
  Vendor(oui):                      79 64 0
  Mode:                             ACTIVE
============================================================

OSmon> getlinkoam interfacestatus STN=0,EthernetInterface=0,LinkOAM=0
```

```
============================================================
Ethernet LinkOAM Interface Status
============================================================
General:
 Mode:                      ACTIVE
 PDU Max Rate:              10 packets per second
 PDU Min Rate:              1 packet per second
 Link timeout:              5 seconds
Link Monitoring:
 Status:                    NOT SUPPORTED
Frame Error:                NOT SUPPORTED
 Window:
 Low threshold:
 High threshold:
Frame Period Error:         NOT SUPPORTED
 Window:
 Low threshold:
 High threshold:
Frame Seconds Error:        NOT SUPPORTED
 Window:
 Low threshold:
 High threshold:
============================================================
```

OSmon> **getlinkoam** ***statistics STN=0,EthernetInterface=0,LinkOAM=0***

```
============================================================
Ethernet oam statistics
============================================================
Counters:
  Total OAMPDU Tx:                                  0
  Total OAMPDU Rx:                                  0
  Information OAMPDU Tx:                            0
  Information OAMPDU Rx:                            0
  Loopback control OAMPDU Tx:                       0
  Loopback control OAMPDU Rx:                       0
  Unsupported OAMPDU Rx:                            0
============================================================
Local event logs:
  0          Errored symbol period records
  0          Errored frame records
  0          Errored frame period records
  0          Errored frame seconds records
============================================================
Remote event logs:
  0          Errored symbol period records
  0          Errored frame records
  0          Errored frame period records
  0          Errored frame seconds records
============================================================
```

4.29 GetMOAttribute

This command displays MO attribute values.

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

4.29.1 Input Parameters

Syntax: **getMOAttribute** ***[<transactionId>*** | ***<MO-DN>*** | ***<flag>] [<MO-DN>*** | ***<flag>*** | ***<attributeId>] [<attributeId>]***

Table 48 Input Parameters

Parameter Name	Description
transactionId (optional)	Unique transaction identifier. Valid value: a string of up to 10 characters. Optional parameter used when the command is executed in a transaction (between the commands **startTransaction** and **endTransaction**). The value returned when used in a transaction is the value stored in the RAM which might not be the same as the currently used value. Changing MO attributes with the **setMOAttribute** command creates a temporary copy of the MO model in the RAM. The values in the RAM are used after the next **commit** command. This parameter is not used when the command is executed outside a transaction. The returned value will in this case be the currently used value stored in the flash memory.
MO-DN (optional)	The MO instance that is read.
flag (optional)	The only current alternative is **-c**. When this flag is used, the command will return values from normal attributes and counters.
attributeId (optional)	Identifies the attribute to be returned.

4.29.2 Output Parameters

Table 49 Output Parameters

Parameter Name	Description
attributeIdOut	Values for the optional MO instance *attributeId* specified as the input parameter.
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.29.3 Example 1

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

The optional parameter *transactionId* is "trans1".
The *MO-DN* is "STN=0,IPInterface=0".
The optional parameter *attributeId* is "primaryIP_Address".

The output parameter *attributeIdOut* (the primary IP address of the unit) is "192.168.59.64". If the value has been changed in the current transaction, the new value from the not commited MO model is returned. A changed value will be in use after execution of the **`commit`** command.

4.29.4 Example 2

In this example the optional *flag* **-c** is used to get normal attributes and counters. If only the **`getmattribute`** command (without flag) is used, the result would be attributes without counters.

```
OSmon> getMOAttribute -c
STN=0; instanceId= 0;
STN=0; operationalState= enabled;
STN=0; depIP_Interface= ;
STN=0; depLocalRoutingPolicy= ;
STN=0; wakeUpRegistration= 0.0.0.0;
STN=0; wakeUpEventInterval= 1;
STN=0; SW_PrimaryProductNumber= CXP102128_1;
STN=0; SW_PrimaryProductRevision= R2B04;
STN=0; SW_BackupProductNumber= CXP102128_1;
STN=0; SW_BackupProductRevision= R2B04;
STN=0; alarmSupervisionActive= false;
STN=0; DSCP_OperationAndMaintenance= 0;
```

```
STN=0; DSCP_OM_FileTransfer= 0;
STN=0; STN_PGW_KeepalivePeriod= 1;
STN=0; STN_PGW_L2TP_MaxTransmissions= 5;
STN=0; STN_PGW_L2TP_RetransmissionCap= 4;
STN=0; STN_PGW_L2TP_InitialRetransmissionPeriod= 1;
STN=0; STN_Name= ;
STN=0; restartReason= orderedReset;
STN=0; systemClockTimeServer= 0.0.0.0;
STN=0; systemClockTimeServerType= NTP;
STN=0; STN_systemClockUDP_port= 123;
STN=0; systemClockUDP_Port_General_PTP= 320;
STN=0; systemClockUDP_Port_Event_PTP= 319;
STN=0; contactWithFileServer= true;
STN=0; failedPMUploads= 0;
STN=0;Equipment=0; instanceId= 0;
STN=0;Equipment=0; serialNumber= CB4E547179;
STN=0;Equipment=0; productRevision= R1E;
STN=0;Equipment=0; productNumber= KDU 137 596/2;
STN=0;Equipment=0; manufacturingDate= 20100926;
STN=0;Equipment=0; productName= SIU02;
STN=0;RoutingTable=0; instanceId= 0;
STN=0;Synchronization=0; instanceId= 0;
STN=0;Synchronization=0; TS_STN_SynchStatus= synchronized;
STN=0;Synchronization=0; TS_UsedTimeServer= ;
STN=0;Synchronization=0; TS_NoTS_Reselections= 0;
STN=0;Synchronization=0; calibrationExpireDate= 20111014;
STN=0;Synchronization=0; synchType= notDefined;
STN=0;Synchronization=0; DSCP_Synchronization= 0;
STN=0;Synchronization=0; depIP_Interface= ;
STN=0;Synchronization=0; usedSynchSource= ;
STN=0;MeasurementDefinition=0; instanceId= 0;
STN=0;MeasurementDefinition=0; neIndex= 0;
STN=0;MeasurementDefinition=0; filePrefix= STN;
STN=0;MeasurementDefinition=0; granularityPeriod= 15;
STN=0;MeasurementDefinition=0; reportPeriod= 15;
STN=0;MeasurementDefinition=0; measurementActive= false;
OperationSucceeded
```

4.29.5 Example 3

The two examples below show the use with an individual MO-DN. First the normal attributes are displayed by excluding the **-c** flag. Then the **-c** flag is used and the result is both normal attributes and counters of MO **LinkOAM**.

```
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
```

```
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
```

4.30 GetPendingConfiguration

This command is used to determine if there is a bulk CM download file ready to be activated with a **activateSW** command together with the SW. If a file exists, a filename and MIM version is returned.

4.30.1 Input Parameters

None.

4.30.2 Output Parameters

Table 50 Output Parameters

Parameter Name	Description
filename	Specifies the filename of the bulk CM file excluding directory information. No existing bulk CM download file is indicated with `--`.
mimVersion	Specifies the MIM version from the bulk CM file header.
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.30.3 Example

```
OSmon> getPendingConfiguration
stn82_CM.xml 3.0
OperationSucceeded
```

The output parameter *filename* is "stn82_CM.xml".
The output parameter *mimVersion* is "3.0".

4.31 GetSessionIds

This command returns the identity of existing bulk CM sessions.

4.31.1 Input Parameters

None

4.31.2 Output Parameters

Table 51 Output Parameters

Parameter Name	Description
sessionIdList	This parameter shows a list of sessionId(s) for existing bulk CM session(s). No existing bulk CM session is indicated with `--`.
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.31.3 Example

```
OSmon> getSessionIds
bcm1
OperationSucceeded
```

The output parameter *sessionIdList* is "bcm1".

4.32 GetSessionStatus

This command returns the status of a specific bulk CM session.

4.32.1 Input Parameters

Syntax: **getsessionstatus** ***<sessionId>***

Table 52 Input Parameters

Parameter Name	Description
sessionId	Unique session identifier. Valid value: a string of up to 10 characters.

4.32.2 Output Parameters

Table 53 Output Parameters

Parameter Name	Description
sessionState	Indicates the state of the specified bulk CM session. Possible values: "Idle", "UploadInProgress", "UploadFailed", "UploadCompleted", "DownLoadInProgress", "DownloadFailed", "DownloadCompleted", "ActivationInProgress", "ActivationFailed", "ActivationCompleted".
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.32.3 Example

```
OSmon> getSessionStatus bcm1
DownLoadInProgress
OperationSucceeded
```

The *sessionId* is "bcm1".
The output parameter *sessionState* is "DownLoadInProgress".

4.33 GetSubscriptionIds

This command returns all existing subscriptions.

4.33.1 Input Parameters

None.

4.33.2 Output Parameters

Table 54 Output Parameters

Parameter Name	Description
subscriptionIdSet	This parameter shows a list of existing subscriptionId(s). No existing subscriptionId(s) is indicated with `--`.
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.33.3 Example

```
OSmon> getSubscriptionIds
1 2 3
OperationSucceeded
```

The output parameter *subscriptionIdSet* is "1", "2" and "3".

4.34 GetSubscriptionStatus

This command returns the characteristics of subscriptions.

4.34.1 Input Parameters

Syntax: **`getSubscriptionStatus`** ***`<subscriptionId>`***

Table 55 Input Parameters

Parameter Name	Description
subscriptionId	The subscriptionId for which characteristics will be returned. Valid values: 1–10

4.34.2 Output Parameters

Table 56 Output Parameters

Parameter Name	Description
subscriptionState	Specifies the state of the specified subscription. Possible values: `notSuspended` (active) or `suspended`.
timeTick	Specifies the time interval (in minutes) between heartbeat notifications. If the value is "0", heartbeat is turned off.
managerReference	Specifies the IP address of the SNMP manager to which notifications are sent (the subscriber) when this subscription is active.
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.34.3 Example

```
OSmon> getSubscriptionStatus 1
notSuspended 2 192.168.59.62
OperationSucceeded
```

The *subscriptionId* is "1".
The output parameter *subscriptionState* is "notSuspended".
The output parameter *timeTick* is "2".
The output parameter *managerReference* is "192.168.59.62".

4.35 GetSWSessionIds

This command returns the identity of existing software sessions.

4.35.1 Input Parameters

None

4.35.2 Output Parameters

Table 57 Output Parameters

Parameter Name	Description
sessionIdList	This parameter shows a list of existing sessionId(s). No existing sessionId is indicated with `--`.
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.35.3 Example

```
OSmon> getSWSessionIds
s1
OperationSucceeded
```

The output parameter *sessionIdList* is "s1".

4.36 GetSWSessionStatus

This command returns the status of a specific software upgrading or downgrading session.

4.36.1 Input Parameters

Syntax: **`getSWSessionStatus`** ***`<sessionIdSW>`***

Table 58 Input Parameters

Parameter Name	Description
sessionIdSW	Unique session identifier. Valid value: a string of up to 10 characters.

4.36.2 Output Parameters

Table 59 Output Parameters

Parameter Name	Description
sessionState	Indicates the state of the specified software upgrading/downgrading session. Possible values: "Idle", "DownLoadInProgress", "DownloadFailed", "DownloadCompleted", "ActivationInProgress", "ActivationFailed", "ActivationCompleted".
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.36.3 Example

```
OSmon> getSWSessionStatus s1
DownLoadInProgress
OperationSucceeded
```

The *sessionIdSW* is "s1".
The output parameter *sessionState* is "DownLoadInProgress".

4.37 Gettime

This command displays the current system clock setting.

4.37.1 Input Parameters

None.

4.37.2 Output Parameters

Table 60 Output Parameters

Parameter Name	Description
Current Time	Specifies the current time.

4.37.3 Example

```
OSmon> gettime
Current Time [1970-01-02 06:26:17.266]
```

4.38 GetTransactionId

This command returns the identity of the transaction (transactionId).

4.38.1 Input Parameters

None.

4.38.2 Output Parameters

Table 61 Output Parameters

Parameter Name	Description
transactionId	Specifies the unique transaction identifier. No existing transaction is indicated with `--`.
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.38.3 Example

```
OSmon> getTransactionId
 OSmon session ID          transaction ID
         241                   trans1
OperationSucceeded
```

The output parameter *transactionId* is "trans1".

4.39 GetTransactionStatus

This command returns the status of the transaction, and is only possible to perform within a transaction. If the command is executed outside a transaction, then it returns `OperationFailed`.

4.39.1 Input Parameters

Syntax: **getTransactionStatus** ***<transactionId>***

Table 62 Input Parameters

Parameter Name	Description
transactionId	Unique transaction identifier. Valid value: a string of up to 10 characters.

4.39.2 Output Parameters

Table 63 Output Parameters

Parameter Name	Description
transactionState	The transaction state is "Idle" when **startTransaction** is performed but **commit** is not yet performed (for example while several **setMOAttribute** commands are performed). Possible values: "Idle", "CommitFailed", "CommitCompleted".
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.39.3 Example

```
OSmon> getTransactionStatus trans1
Idle
OperationSucceeded
```

The *transactionId* is "trans1".
The output parameter *transactionState* is "Idle".

4.40 Ping

This command is a SIU adapted tool to execute ping test. Use the **help** argument to list all options available for this command.

4.40.1 Input Parameters

Syntax: **ping** ***destination***

Table 64 Input Parameters

Parameter Name	Description
destination	Specifies the IP address of the host to which echo requests are sent to test the connectivity.

4.40.2 Output Parameters

Table 65 Output Parameters

Parameter Name	Description
response	This parameter is a list of the responses.

4.40.3 Example

OSmon>**ping** ***<destination IP address***

4.41 Ping_Ethernet

This command is used to send out LoopBack Messages (LBM) from the SIU. LBMs are used for fault verification, typically performed after fault detection. They can also confirm successful initiation or restoration of connectivity.

4.41.1 Input Parameters

Syntax: **ping_ethernet** ***<MAC | mepid MEPID> <domain [TYPE <no-name|mac>] DOMAIN_NAME> source-mepid SOURCE_MEPID [vlan VLANID] [tlv_data_bytes DATA <1-1452>|tlv_pattern PATTERN <1-4>] [pcp PCP_PRI <0-7>] [loopback-reply-timeout TIMEOUT <1-10>] [loopback-tx-interval INTERVAL <1-10>] [loopback-count COUNT <1-10>]***

Table 66 Input Parameters

Parameter Name	Description
MAC	The destination MAC address of LBMs. Its format follows hh:hh:hh:hh:hh:hh.
mepid MEPID	Specifies the value of remote MEP ID which is going to receive LBMs. The *MAC* and *mepid* are exclusive.
domain [TYPE <no-name|mac>] DOMAIN_NAME	Specifies the related domain name in which the LBMs should be sent. The maintenance domain type needs to be specified if it is *no-name* or *mac*.

Parameter Name	Description
source-mepid SOURCE_MEPID	Specifies the value of the MEP ID of the LBM sender.
vlan VLANID	Specifies the value of VLAN ID. It should be included if attachment point is VLAN tagged.
tlv_data_bytes DATA <1-1452>	Specifies the number of bytes in TLV data.
tlv_pattern PATTERN <1-4>	Specifies the pattern of Test TLV. Patterns 1, 2, 3, and 4 have "abc", "1234", "a1b2c", and "1a2b3c" respectively. The *tlv_data_bytes* and *tlv_pattern* should be exclusive.
pcp PCP_PRI <0-7>	Specifies PCP VLAN priority (the p-bit). The range is from 0 to 7 with 7 as default value.
loopback-reply-timeout TIMEOUT <1-10>	Specifies the time (in seconds) to wait for a LoopBack Reply (LBR). The range is between 1 and 10 seconds with 1 second as default.
loopback-tx-interval INTERVAL <1-10>	Specifies the interval between each LBM. The range is between 1 to 10 seconds with 1 second as default.
loopback-count COUNT <1-10>	Specifies the number of LBMs to send. Its range is from 1 to 10 with 3 as default.

4.41.2 Output Parameters

Table 67 Output Parameters

Parameter Name	Description
success rate	The ratio of number of LBMs sent and replies received.

4.41.3 Example

```
OSmon>ping_ethernet mepid 8191 domain ericsson0.com source-mepid
1 vlan 340 tlv_pattern 4 loopback-count 1

success rate is 100% (1/1)
```

4.42 ResetToFactorySetting

This command resets the configuration of the unit to default factory settings. MIB content is erased and persistent data including security and synchronization data is cleared. Software archives are not affected by this command.

4.42.1 Input Parameters

None.

4.42.2 Output Parameters

Table 68 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.42.3 Example

```
OSmon> resetToFactorySetting
```

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

```
Warning:  All user settings will be removed, continue
(ok/cancel)?

OperationSucceeded
```

4.43 Restart

This command initializes a restart of the unit. All connections to the unit are reset as part of the operation. The restart also clears PM data. If a subscription exists, sending heartbeat notifications is resumed when the restart is completed.

A restart can take up to 1 minute and CLI sessions are terminated. A new login is required when the restart is completed.

4.43.1 Input Parameters

None.

4.43.2 Output Parameters

Table 69 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.43.3 Example

```
OSmon> restart
OperationSucceeded
```

4.44 Restore

This command is used to restore a bulk CM XML configuration file.

Note: A restore is manually executed in two steps; using the commands **`download`** and **`restore`**.

During restore, the backed up configuration is first downloaded from the provided SFTP path (using the **`download`** command), then decrypted (using the **`restore`** command) with the same password (as entered when the file was backed up using the **`backup`** command) and finally an attempt to activate it is done.

If the restored configuration IRP version is supported by current software, the configuration can be activated immediately with warm restart.

If the restored configuration IRP version is not supported by current software, another software version needs to be installed (**`downloadSW`** and **`activateSW`** commands).

4.44.1 Input Parameters

Syntax: **`restore`** ***`<sessionId> <passwd> [delayed] [checkOnly]`***

Table 70 Input Parameters

Parameter Name	Description
sessionId	Unique session identifier. Valid value: a string of up to 10 characters.

Parameter Name	Description
passwd	The password to be used when the downloaded file is being decrypted. This password is the same as the one used when the file was backed up.
delayed (optional)	If set, the restored file is only stored persistently. The activation of the restored file will be performed on a system created MIB, when the **`SWSession`** is activated with the optional parameter *useNewConfiguration* set. If not set, and the configuration version is supported, the configuration will be activated immediately with node warm restart. If set, and the configuration version is not supported, an error message is printed that the configuration requires another SW version.
checkOnly (optional)	If set, only a message containing configuration version and ability to restore on current SW will be printed.

4.44.2 Output Parameters

Table 71 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.44.3 Example 1

```
OSmon> restore bcm1 admin
OperationSucceeded
```

The *sessionId* is "bcm1".
The *passwd* is "admin".

4.44.4 Example 2

```
OSmon> restore bcm1 admin delayed
OperationSucceeded
```

The sessionId is "bcm1".
The passwd is "admin".
The optional parameter *delayed* is included if the activation of the restored file should be performed within a **SWSession** with the optional parameter *useNewConfiguration* set.

4.44.5 Example 3

```
OSmon> restore bcm1 admin checkOnly
OperationSucceeded
```

The *sessionId* is "bcm1".
The *passwd* is "admin".
The optional parameter *checkOnly* is included to only get a message containing configuration version and the ability to restore on current configuration.

4.45 ResumePMMeasurements

This command initiates/enables PM data uploading and specifies details of the file store destination.

4.45.1 Input Parameters

Syntax: **resumePmMeasurements** *<PMMeasurementInstance> [useContainerFile] <fileStore>*

Table 72 Input Parameters

Parameter Name	Description
PMMeasurementInstance	Instance of PM measurement MO. The value is always "STN=0,MeasurementDefinition=0".

Parameter Name	Description
useContainerFile (optional)	If the optional parameter *useContainerFile* is given, this indicates that the administrative file and all data files will be combined into a container file before transfer. The file format for the container file is ".tar".
fileStore	An SFTP URI, including authentication information, that defines where to upload the PM reports. The addressing of the file destination must follow the syntax: sftp://username:password@a.b.c.d/filepath where "a.b.c.d" specifies the IP address of the file destination host and "filepath" is the directory structure specified from the root. **Note:** In addition to the "filepath", a subdirectory named "PM" must exist at the file store destination.

4.45.2 Output Parameters

Table 73 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.45.3 Example

```
OSmon> resumePMMeasurements STN=0,MeasurementDefinition=0
useContainerFile sftp://jones:abc123@192.0.72.93/home/jones/ARC
HIVE/STN95
OperationSucceeded
```

The *PMMeasurementInstance* is "STN=0,MeasurementDefinition=0".
The *useContainerFile* specifies that the administrative file and all data files should be combined into a container (.tar) file before transfer.
The *fileStore* is "sftp://jones:abc123@192.0.72.93/home/jones/ARCHIVE/STN95".

4.46 ResumeSubscription

This command activates a defined, but suspended, subscription. The result is alarms and notifications being sent.

4.46.1 Input Parameters

Syntax: **`resumeSubscription`** ***`<subscriptionId>`***

Table 74 Input Parameters

Parameter Name	Description
subscriptionId	The identity for the subscription to resume. Valid values: 1–10.

4.46.2 Output Parameters

Table 75 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.46.3 Example

```
OSmon> resumeSubscription 2
OperationSucceeded
```

The *subscriptionId* is "2".

4.47 Rev

This command returns revisions of everything.

4.47.1 Input Parameters

None.

4.47.2 Output Parameters

See example.

4.47.3 Example

```
OSmon> rev
----------------- OSE modules ------------------------
oam.chk           Operation & Maintenance        R2G02
secmgrr.chk       Security Manager               R2G02
inetr.chk         MLPPP Daemon                   R2D03
ltp.chk           Local Terminal Port            R2D03
snc.chk           Synchronization                R2G01
pd.chk            Packet Distributor             R1L_1
cesopsn_pwr.chk   Circuit Emulation Service      R1F03
hdlc_pwr.chk      HDLC Pseudo-Wire Emulation     R1H07
p_relayr.chk      Packet Relay                   R1F05
profiler.chk      System Profiler                R1L_1
lcf_cp.chk        Local Connectivity Function    R1C01
bsp.drv           Board Support Package          R2G01
bootstrap.chk     Bootstrap                      R2G02
loader.drv        Software Loader                R1L_1
linuxload.drv     Linux Loader                   R2G01

----------------- Firmware modules -------------------
PBOOT             CXC 112 3777/1                 R1E01

----------------- Software archives ------------------
Primary:
OSE               CXP102128_2                    R2G02
Linux             -                              R2G02

Backup:
OSE               CXP102128_2                    R2G02

----------------- Active software --------------------
OSE                                            Primary
Linux                                          Primary
```

4.48 SetMOAttribute

This command sets MO attribute values. One attribute value for one MO instance can be defined in each command. The command must be used within a transaction.

Changing one or more MO attributes with the **`setMOAttribute`** command creates a temporary copy of the MO model that includes all the changes. A **`commit`** command is required to activate this temporary version of the MO model and make it the currently used and persistent MO model.

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

4.48.1 Input Parameters

Syntax: **setMOAttribute** ***<transactionId> <MO-DN> <attributeId> <attributeValue>***

Table 76 Input Parameters

Parameter Name	Description
transactionId	Unique transaction identifier. Valid value: a string of up to 10 characters.
MO-DN	The MO instance that is modified.
attributeId	This parameter identifies the attribute whose value is to be modified.
attributeValue	The value to be used in the modification of the attribute.

4.48.2 Output Parameters

Table 77 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.48.3 Example 1

```
OSmon> setMOAttribute trans1 STN=0 wakeUpRegistration
192.168.59.62
OperationSucceeded
```

This command line defines the address to which wake-up event should be sent.
The *transactionId* is "trans1".
The *MO-DN* is "STN=0".
The *attributeId* is "wakeUpRegistration".
The *attributeValue* is "192.168.59.62".

4.48.4 Example 2

```
OSmon> setMOAttribute trans1 STN=0,EthernetInterface=0 mode auto
OperationSucceeded
```

This command line defines that the "auto" mode shall be used on the specified Ethernet interface.
The *transactionId* is "trans1".
The *MO-DN* is "STN=0, EthernetInterface=0".
The *attributeId* is "mode".
The *attributeValue* is "auto".

4.49 Settime

This command sets the value of the system clock.

4.49.1 Input Parameters

Syntax: **settime** *Year(1970->) Mon(1-12) Day(1-31) Hour(0-23) Min(0-59) Sec(0-59) Msec(0-999)*

4.49.2 Output Parameters

None.

4.49.3 Example

```
OSmon> settime 2010 11 18 16 25 4 6
```

4.50 ShowMACFwTable

This command shows all entries (dynamically learned and statically configured) in the MAC forwarding table.

4.50.1 Input Parameters

Syntax: **showMACFwTable** *all* | *<EthernetInterface> [<tagValue>]*

Table 78 Input Parameters

Parameter Name	Description
all	Specifies that all entries in the MAC forwarding table should be shown.

Parameter Name	Description
EthernetInterface (optional)	If this optional parameter is set , only entries in the MAC forwarding table of the specified MO **EthernetInterface** instance is shown.
tagValue (optional)	Specifies the tag value (VLAN Id) of table entries to show. This parameter is optional and can only be used with the *EthernetInterface* option.

4.50.2 Output Parameters

None

4.50.3 Examples

Each entry in the MAC forwarding table is displayed on a separate line. For each entry, the following information is displayed: MAC address, tag value (VLAN Id), EthernetInterface and timeout (aging time). The first row displays the legend (MACAddress, tagValue, EthernetInterface and Timeout) and the columns are separated by white spaces and/or tabs.

OSmon> **showMACFwTable** ***all***
This command displays all entries in the MAC forwarding table.

OSmon> **clearMACFwTable** ***STN=0,EthernetInterface=0***
This command displays MAC forwarding table entries for the specified MO **EthernetInterface**.

OSmon> **clearMACFwTable** ***STN=0,EthernetInterface=0 200***
This command displays MAC forwarding table entries for the specified MO **EthernetInterface** with specified *tagValue*.

4.51 SimplePing

This command tests whether the unit could reach a defined destination IP address by sending three echo requests from a source IP address through the IP network. The test checks the IP network connectivity and can be useful at installation prior to taking the unit into service.

4.51.1 Input Parameters

Syntax: **simpleping** ***<source address> <destination address>***

Table 79 Input Parameters

Parameter Name	Description
source address	Specifies the IP address from which the echo requests are sent to a defined destination IP address. For the operation to succeed, the source IP address must be the primary IP address or one of the secondary IP addresses defined by the attribute *depIP_interface* of MO **STN**.
destination address	Specifies the IP address of the host to which the three echo requests are sent to test the connectivity. The time between send requests is 107 ms and the packet size is 56 data bytes.

4.51.2 Output Parameters

Table 80 Output Parameters

Parameter Name	Description
response	This parameter is a list of the responses.
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.51.3 Example

```
OSmon> simpleping 192.168.59.62 192.168.69.152
3/3 received OK
OperationSucceeded
```

The *source address* is "192.168.59.62".
The *destination address* is "192.168.69.152".

4.52 StartSession

This command starts a bulk CM session. A bulk CM session can be used in parallel with a software upgrading session but cannot be used in parallel with a basic CM transaction. Only one bulk CM session can be active at the time.

4.52.1 Input Parameters

Syntax: **startsession** ***<sessionId> [<ossIpAddress>]***

Table 81 Input Parameters

Parameter Name	Description
sessionId	Unique session identifier. Valid value: a string of up to 10 characters.
ossIpAddress (optional)	The IP address of the SNMP Manager to which notifications are sent. If used, this IP address is only used as a trap receiver while the session exists. When the command **endSession** is given, traps will no longer be sent to this address. If omitted, no notification is sent.

4.52.2 Output Parameters

Table 82 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.52.3 Example

```
OSmon> startSession bcm1 192.168.59.62
OperationSucceeded
```

The *sessionId* is "bcm1".
The optional parameter *ossIpAddress* is "192.168.59.62".

4.53 StartSWSession

This command starts a software upgrading or downgrading session. One basic CM transaction or one bulk CM session can be used in parallel with a software session. Only one software session can be active at the time.

4.53.1 Input Parameters

Syntax: **startSWSession** ***<sessionIdSW> [<ossIpAddress>]***

Table 83 Input Parameters

Parameter Name	Description
sessionIdSW	Unique session identifier. Valid value: a string of up to 10 characters.
ossIpAddress (optional)	The IP address of the SNMP Manager to which notifications are sent. If used, this IP address is only used as a trap receiver while the SW session exists. When the command **endSWSession** is given, traps will no longer be sent to this address. If omitted, no notification is sent.

4.53.2 Output Parameters

Table 84 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.53.3 Example

```
OSmon> startSWSession s1 192.168.59.62
OperationSucceeded
```

The *sessionIdSW* is "s1".
The optional parameter *ossIpAddress* is "192.168.59.62".

4.54 StartTransaction

This command starts a basic CM transaction. Basic CM transactions can be used in parallel with software upgrading sessions but cannot be used in parallel with bulk CM sessions. Only one basic CM transaction can be active at the time.

Basic CM commands can be executed within a basic CM transaction. Each command performed in a transaction is validated immediately and executed

when the **commit** command is received. The transaction is persistent at ordered commit.

4.54.1 Input Parameters

Syntax: **startTransaction** *<transactionId> [<ossIpAddress>]*

Table 85 Input Parameters

Parameter Name	Description
transactionId	Unique transaction identifier. Valid value: a string of up to 10 characters.
ossIpAddress (optional)	The IP address of the SNMP Manager to which notifications are sent. If used, this IP address is only used as a trap receiver while the transaction exists. When the command **endTransaction** is given, traps will no longer be sent to this address. If omitted, no notification is sent.

4.54.2 Output Parameters

Table 86 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.54.3 Example

```
OSmon> startTransaction trans1 192.168.59.62
OperationSucceeded
```

The *transactionId* is "trans1".
The optional parameter *ossIpAddress* is "192.168.59.62".

4.55 Subscribe

This command defines the IP address of an SNMP manager (which is an OSS application) and the time interval between heartbeat notifications. When the command is issued, alarms and notifications are sent to the SNMP manager.

There can be up to 10 active subscriptions but each subscription requires a unique SNMP manager.

4.55.1 Input Parameters

Syntax: **subscribe** ***<managerReference> <timeTick>***

Table 87 Input Parameters

Parameter Name	Description
managerReference	The IP address of the SNMP manager to which notifications are sent.
timeTick	The time interval (in minutes) between heartbeat notifications. If the value is set to "0", heartbeat is turned off. Valid values: 0–10.

4.55.2 Output Parameters

Table 88 Output Parameters

Parameter Name	Description
subscriptionId	The identity of the subscription assigned by the system. Possible values: 1–10.
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.55.3 Example

```
OSmon> subscribe 192.168.59.62 2
1
OperationSucceeded
```

The *managerReference* is "192.168.59.62".
The *timeTick* is "2".
The output parameter *subscriptionId* is "1".

4.56 SuspendPMMeasurements

This command stops/disables PM data collection and PM data upload.

4.56.1 Input Parameters

Syntax: **suspendPmMeasurements** ***<PMMeasurementInstance>***

Table 89 Input Parameters

Parameter Name	Description
PMMeasurementInstance	The instance of the PM measurement MO. The value is always "STN=0,MeasurementDefinitio n=0".

4.56.2 Output Parameters

Table 90 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.56.3 Example

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

The *PMMeasurementInstance* is "STN=0,MeasurementDefinition=0".

4.57 SuspendSubscription

This command disables/stops the sending of alarms and notifications for the specified subscription.

4.57.1 Input Parameters

Syntax: **suspendSubscription** ***<subscriptionId>***

Table 91 Input Parameters

Parameter Name	Description
subscriptionId	The identity for the subscription to suspend. Valid values: 1–10.

4.57.2 Output Parameters

Table 92 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.57.3 Example

```
OSmon> suspendSubscription 2
OperationSucceeded
```

The *subscriptionId* is "2".

4.58 Syslog

This command reads or monitors the system, event, or messages log. Each log record stored in the log contains the following information: source file and line number, process name, time stamp, type of event, and a message string.

Note: For backwards compatibility reasons, "log read" and "log monitor" can be used as aliases.

4.58.1 Input Parameters

Syntax: **`syslog`** ***`<read|monitor> -s <system|event|messages> [-f filterspec] [-n numlines]`***

Table 93 Input Parameters

Parameter Name	Description
-s <system|event|messages>	Specifies whether it is the system, event, or messages log that are the source. Default is "system".

Parameter Name	Description
[-f filterspec]	Allows definition of textual search string for which a match is done on the fill event entry (time stamps, process name, and so on).
[-n numlines]	Specifying this option will print the latest numlines entries. If combined with a -f specification it will print the latest numlines matches.

4.58.2 Output Parameters

The command shows lines of formatted events. Each event line includes: Time stamp, process name, file name, event type, and information string.

4.58.3 Examples

OSmon> **syslog** ***read -s event***
This command prints log events to the terminal. The printing is aborted when the whole 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**.

4.59 Timeservertest

This command is used to check the connectivity to and quality on the network towards a time server. If the attribute *synchType* (in MO **Synchronization**) is set to value "timeServer" and calibration is ongoing, the synchronization algorithm will freeze during the test. Once a test is started, 10 packets per second will be sent towards the time server and the test will be ready within 0,5 minutes. If the network has long delay (Satellite link), this will be detected by the test and the packet rate will be reduced to 1 packet per second and the test will be ready within 5 minutes. A time server test does not survive either a warm start or a cold start.
The result of a test shows the network quality and packet loss and is cleared after the result is reported.

4.59.1 Input Parameters

Syntax: **timeservertest** ***<(start [instanceId]/result)>***

Table 94 Input Parameters

Parameter Name	Description
start	Starts the test with current used time server (*synchType* must be set to "timeServer").
instanceId (optional)	Used to activate the test with an optional time server. The value is the *instanceId* of an configured time server (*synchType* can be different from the value "timeServer").
result	Used to retrieve the result of a test.

4.59.2 Output Parameters

None.

4.59.3 Example

```
OSmon> timeservertest start
OSmon> timeservertest start 2
OSmon> timeservertest result
```

4.60 TraceAllMOAttributeInLog

This command prints values of all MIB configuration attributes to the log. This can be used to get the complete configuration.

Note: Counter values are returned as raw data. Accurate values of counters should be read using the **`resumePmMeasurements`** command.

4.60.1 Input Parameters

None.

4.60.2 Output Parameters

Table 95 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.60.3 Example

```
traceallmoattributeinlog
OperationSucceeded
```

Use the command **syslog** ***read -s event*** to print the log.

4.61 Traceroute

This command is a SIU adapted tool to execute traceroute test. Use the **help** argument to list all options available for this command.

4.61.1 Input Parameters

Syntax: **traceroute** ***host [packetlen]***

Table 96 Input Parameters

Parameter Name	Description
host	The destination host to traceroute to.
packetlen (optional)	Specifies an alternate probe packet length (default is 40).

4.61.2 Output Parameters

None.

4.61.3 Example

OSmon> **traceroute** ***<host IP address>***

4.62 Traceroute_Ethernet

This command is used to send out LinkTrace Messages (LTM) from the SIU. Multicast LTMs are used to perform path discovery and fault isolation.

4.62.1 Input Parameters

Syntax: **traceroute_ethernet** ***MAC <domain [type <no-name>|mac>] DOMAIN_NAME | level LEVEL> [vlan VLANID]***

Table 97 Input Parameters

Parameter Name	Description
MAC	The destination MAC address of LTMs. Its format follows hh:hh:hh:hh:hh:hh.
domain [type <no-name>|mac>] DOMAIN_NAME	Specifies the related maintenance domain name in which the LTMs should be sent. The maintenance domain type needs to be specified if it is *no-name* or *mac*.
level LEVEL	Specifies the level of the maintenance domain in which the LTMs should be sent. The parameters *level* and *domain* are exclusive.
vlan VLANID	Specifies the value of VLAN ID. It should be included if attachment point is VLAN tagged.

4.62.2 Output Parameters

Table 98 Output Parameters

Parameter Name	Description
MP Mac	Destination MAC address of the LTM.
Hops	Hop count when the LTM passes the device.
Relay-action	Indicates whether the destination MAC address if found by forwarding device. "RlyHit" indicates that the destination MAC address as been successfully found.

4.62.3 Example

```
OSmon> traceroute_ethernet 00:04:96:34:63:C4 level 0 vlan 250

MP Mac                      Hops            Relay-action
00:04:96:34:63:c4             1               RlyFDB
00:04:96:34:63:c4             2               RlyHit
```

4.63 Unsubscribe

This command disables/stops the sending of alarms and notifications and removes the specified subscription. When **unsubscribe** is performed for the last defined subscription, and attribute *wakeUpRegistration* in MO **STN** is defined, sending of wakeUpMessages is done instead of alarms and notifications.

4.63.1 Input Parameters

Syntax: **unsubscribe** *<subscriptionId>*

Table 99 Input Parameters

Parameter Name	Description
subscriptionId	The identity for the subscription to remove. Valid values: 1–10.

4.63.2 Output Parameters

Table 100 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.63.3 Example

```
OSmon> unsubscribe 1
OperationSucceeded
```

The *subscriptionId* is "1".

4.64 Upload

This command orders the uploading of a bulk CM XML configuration file to the destination file specified. If the specified file already exists, it will be replaced by the new file.

4.64.1 Input Parameters

Syntax: **upload** *<sessionId> <FileRef> <fileContent> [<progressInterv>]*

Table 101 Input Parameters

Parameter Name	Description
sessionId	Unique session identifier. Valid value: a string of up to 10 characters.

Parameter Name	Description
FileRef	Specifies a globally unique destination for the upload. The addressing of the file source must follow the syntax: sftp://username:password@a.b.c.c/filepath where "a.b.c.d" specifies the IP address of the destination host, and "filepath" specifies the directory structure and filename.
fileContent	Defines which data from the MIB that should be uploaded. Valid value:"CM" The value "CM" results in uploading of configuration data from all MOs except MO **MeasuredMOClass**.
progressInterv (optional)	Defines a time (in seconds) after which a software download progress message should be sent to the SNMP manager. Valid values: 0 and 15–900 If set to "0" or not set, no notifications will be sent.

4.64.2 Output Parameters

Table 102 Output Parameters

Parameter Name	Description
status	Indicates either of the following: (a) `OperationSucceeded`. (b) `OperationFailed` - followed by the specified or unspecified reason.

4.64.3 Example

```
OSmon> upload bcm1 sftp://smith:123abc@192.168.69.152/STNCFG/
stn82_CM.xml CM 30
OperationSucceeded
```

The *sessionId* is "bcm1".
The *uploadDataFileReference* is "sftp://smith:123abc@192.168.69.152/STN CFG/stn82_CM.xml".
The *fileContent* is "CM".

The optional parameter *progressInterv* is "30".

4.65 Uptime

This command displays for how long the system has been running.

4.65.1 Input Parameters

None.

4.65.2 Output Parameters

Table 103 Output Parameters

Parameter Name	Description
Uptime	Specifies the uptime for the system.

4.65.3 Example

```
OSmon> uptime
Uptime:  0 days, 23 hours, 23 minutes, 2 seconds
```

4.66 UseLocalSFTP

When set to on, this command causes the SIU to use the local Console port (instead of the WAN interface) when transferring files via SFTP. At power on, or when the unit is restarted, this command is set to the default value (off).

4.66.1 Input Parameters

Syntax: **uselocalsftp** *<on|off>*

Table 104 Input Parameters

Parameter Name	Description
on	The local port will be used for SFTP file transfers.
off (default)	The WAN interface will be used for SFTP file transfers.

4.66.2 Output Parameters

None.

4.66.3 Example

```
OSmon> uselocalsftp on
```

5 Terminology

autoRollback — The autoRollback functionality makes the unit automatically revert back to the previous configuration (MIB) and/or the previous SW if OSS or the user fails to contact the unit and execute the **endSession** / **endSWSession** / **endTransaction** command within a specified time.

autoRollback is a parameter to the commands **activate**, **activateSW** and **commit**.

Reset — An event triggering a restart of the SIU.

Restart — A reset followed by a cold start or warm start.

Cold start — A start sequence where the entire RAM memory is cleared and all software is loaded from flash to RAM before started.

Warm start — A 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**. The Bulk CM file can cause MOs to be deleted and or created, and MO attributes to be changed.

Managed Object — A Managed Object (MO) is a software object that encapsulates the manageable characteristics and behavior of a particular resource. The MO is an instance of a MO class defined in a MIM. An MO class has attributes that provide information used to characterize the objects that belong to the class. Furthermore,

an MO class can have operations that represent the behavior relevant for that class. An MO class may support notifications that provide information about an event occurrence within a resource. For further information about MO classes, see Reference [4].

MO-DN Each Managed Object (MO) is identified with a Distinguished Name (DN) that expresses its containment hierarchy.

MIM Managed Information Model. A logical model which describes MO classes, their associations, attributes and operations; and describes all possible MIBs.

MIB Management Information Base. The MIB specifies the current instantiation of the MIM; contains MOs, their associations and attributes.

The MIB is configured either by the OSS or from a local terminal. The **`setMOAttribute`** and **`getMOAttribute`** commands will set and get values on specific MO attributes. The MIB must be configured to ensure proper function and for OSS to be able to connect to the unit.

transaction Basic CM transactions ensure that data changes in the MIM are performed consistently. Several attributes can be changed in a transaction by including a **`setMOAttribute`** command for each attribute to change. Each command performed in a transaction is validated immediately and executed when the commit command is received.

session Bulk CM session commands are related to the downloading and uploading of configuration files and activation of new configurations.

SWsession The downloading and activation of software are performed in a software session.

Glossary

See Reference [2].

Reference List

[1] *Library Changes*

[2] *Glossary*

[3] *User Guide, Operation and Maintenance*

[4] *Managed Object Model*

[5] *SIU 02 Description*