

Command Descriptions

COMMAND DESCRIPTION

Copyright

© Ericsson AB 2012. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

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



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	2
2.2	Command Line	2
2.3	Command Completion using <tab>	2
2.4	Invalid Keywords	3
2.5	Command Response	3
3	Generic Commands	3
3.1	Gettime	3
3.2	Settime	4
3.3	Caldate	4
3.4	Calexpdate	5
3.5	Calvalue	6
3.6	ChangePWD	6
3.7	DataEncryption	7
3.8	GenerateKey	9
3.9	Rev	9
3.10	GetIRPVersion	10
3.11	GetOSmonSessions	12
3.12	ShowMACFwTable	12
3.13	ClearDynamicMACFwTable	13
3.14	GetLinkOAM	14
3.15	Ping_Ethernet	17
3.16	Traceroute_Ethernet	19
3.17	Debug	20
3.18	Syslog	20
3.19	Dump	21
3.20	CollectDCGInformation	23
3.21	Ping	24



3.22	Ping6	25
3.23	SimplePing	25
3.24	SimplePing6	26
3.25	Traceroute	28
3.26	Traceroute6	28
3.27	Tracepath	29
3.28	Tracepath6	29
3.29	getOSPF	30
3.30	TraceAllMOAttributeInLog	35
3.31	Timeservertest	36
3.32	Uptime	36
3.33	GetTemperature	37
3.34	GetAlarmList	37
3.35	ResetToFactorySetting	38
3.36	Restart	39
3.37	UseLocalSFTP	40
4	Basic CM Transaction	41
4.1	StartTransaction	41
4.2	CreateMO	42
4.3	DeleteMO	43
4.4	GetMOAttribute	44
4.5	SetMOAttribute	48
4.6	CheckConsistency	49
4.7	Commit	50
4.8	EndTransaction	52
4.9	GetContainment	52
4.10	GetCounters	54
4.11	GetTransactionId	55
4.12	GetTransactionStatus	56
5	Bulk CM Session	57
5.1	StartSession	57
5.2	Download	58
5.3	Upload	60
5.4	Backup	62
5.5	Restore	64



5.6	Activate	66
5.7	EndSession	68
5.8	GetPendingConfiguration	69
5.9	GetSessionIds	70
5.10	GetSessionStatus	71
5.11	AbortSessionOperation	72
6	Software Session	72
6.1	StartSWSession	73
6.2	DownloadSW	74
6.3	ActivateSW	75
6.4	EndSWSession	77
6.5	GetSWSessionIds	78
6.6	GetSWSessionStatus	78
6.7	AbortSWSessionOperation	79
7	Alarm Subscriptions	80
7.1	Subscribe	80
7.2	Unsubscribe	82
7.3	GetSubscriptionIds	82
7.4	GetSubscriptionStatus	83
7.5	SuspendSubscription	84
7.6	ResumeSubscription	85
8	Performance Management	86
8.1	SuspendPMMeasurements	86
8.2	ResumePMMeasurements	87
9	Terminology	88
	Glossary	91
	Reference List	93





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

This chapter describes typographic conventions used in this document as well as general format definitions for the command line.



2.1 Typographic Conventions

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

CommandToBeEnteredExaclyAsWritten

InputParameterValuesSuppliedByTheUser

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, EthernetInte  
rface=
```

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 such as, 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.

If a command fails, the below respond is given followed by the specified or unspecified reason:

```
OperationFailed
```

3 Generic Commands

This chapter lists commands related to the generic functionality, security and troubleshooting of the SIU.

3.1 Gettime

This command displays the current system clock setting.

3.1.1 Input Parameters

None.



3.1.2 Output Parameters

Table 1 Output Parameters

Parameter Name	Description
Current Time	Specifies the current time.

3.1.3 Example

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

3.2 Settime

This command sets the value of the system clock.

3.2.1 Input Parameters

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

3.2.2 Output Parameters

None.

3.2.3 Example

```
OSmon> settime 2011 10 18 16 25 4 6
```

3.3 Caldate

This command displays or sets the current calibration date which shows when the calibration of the OCXO in the SIU has been latest performed.

3.3.1 Input Parameters

Syntax: **caldate** [*<YYYYMMDD>*]



3.3.2 Output Parameters

Table 2 Output Parameters

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

3.3.3 Example

```
OSmon> caldate
20100924
OperationSucceeded
```

3.4 Calexpdate

This command displays or sets the current calibration expire date which shows when calibration of the OCXO in the SIU will expire.

3.4.1 Input Parameters

Syntax: `calexpdate [<YYYYMMDD>]`

3.4.2 Output Parameters

Table 3 Output Parameters

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

3.4.3 Example

```
OSmon> calexpdate
20110924
OperationSucceeded
```



3.5 Calvalue

This command displays or sets the current calibration value which is used for frequency generation. A valid value is -4000 to +4000.

If the value is manually changed, the frequency of the synchronization signals distributed from the SIU may change rapidly and may also result into a temporary "Loss of Synchronization" alarm.

3.5.1 Input Parameters

Syntax: `calvalue [<value>]`

3.5.2 Output Parameters

Table 4 Output Parameters

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

3.5.3 Example

```
OSmon> calvalue
156
OperationSucceeded
```

3.6 ChangePWD

This command changes the password for the super user (username **admin** with default password "hidden") or for the regular user (username **isp** with default password "ispevents").

The password is used to authenticate the user when logging on to the unit through a Telnet, SSH or SFTP connection.

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

3.6.1 Input Parameters

Syntax: `changePWD {user <userId>} | <old_pass> <new_pass> <ver_pass>`



Table 5 Input Parameters

Parameter Name	Description
user <i>userId</i> (optional)	If this option is used, it specifies the <i>userId</i> of the regular user for whom the password must be changed. isp is the only value allowed.
old_pass (optional)	If the <i>user</i> option is not used, this parameter must be present and set to the currently valid password for the admin user.
new_pass	New password for the user.
ver_pass	New password duplicated for verification.

3.6.2 Output Parameters

Table 6 Output Parameters

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

3.6.3 Example 1

```
OSmon> changePWD hidden xyz6789abc xyz6789abc
OperationSucceeded
```

This changes the password for the super user. The *old_pass* is "hidden". The *new_pass* and *ver_pass* are "xyz6789abc".

3.6.4 Example 2

```
OSmon> changePWD user isp xyz6789abc xyz6789abc
OperationSucceeded
```

This changes the password for the regular user. The *userId* is "isp". The *new_pass* and *ver_pass* are "xyz6789abc".

3.7 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 default factory settings using the `resettofactorysetting` command.

3.7.1 Input Parameters

Syntax: `dataencryption <on/print>`

Table 7 Input Parameters

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

3.7.2 Output Parameters

Table 8 Output Parameters

Parameter Name	Description
encryptionstatus	This parameter is only relevant for the <code>print</code> option. It indicates the actual state (activated or notActivated) of encryption of sensitive data.
status	Indicates either of the following: OperationSucceeded. OperationFailed - followed by the specified or unspecified reason.

3.7.3 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)? ok
OperationSucceeded
```

```
(ok/cancel)? cancel
OperationFailed - Command interrupted by user
```



3.7.4 Example 2

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

3.8 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.

3.8.1 Input Parameters

None.

3.8.2 Output Parameters

Table 9 Output Parameters

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

3.8.3 Example

```
OSmon> generateKey
OperationSucceeded
```

3.9 Rev

This command displays revisions of current installed software archives and which one (primary or backup) that is the active software. Revisions for OSE and firmware modules as also displayed.

3.9.1 Input Parameters

None.



3.9.2 Output Parameters

See example.

3.9.3 Example

```

OSmon> rev
----- OSE modules -----
oam.chk          Operation & Maintenance      R2C17
secmgrr.chk      Security Manager            R2C02
inetr.chk        MLPPP Daemon                R2C12
ltp.chk          Local Terminal Port         R2C02
snc.chk          Synchronization             R2C16
pd.chk           Packet Distributor          R2C12
cesopsn_pwr.chk Circuit Emulation Service    R2C12
hdlc_pwr.chk     HDLC Pseudo-Wire Emulation R2C12
p_relayr.chk     Packet Relay                 R2C12
profiler.chk     System Profiler              P1F01
hwtest.chk       Hardware Tester              R2C12
lcf_cp.chk       Local Connectivity Function  R1C01
bsp.drv          Board Support Package        R2C16
bootstrap.chk   Bootstrap                    R2C06
loader.drv       Software Loader              R2C02
linuxload.drv   Linux Loader                  R2C16

----- Firmware modules -----
PBOOT           CXC 112 3777/1              R1E01
FPGA            -                            0x2206

----- Software archives -----
Primary:
OSE             CXP102138_1                 R2C17
Linux           -                            R2C17

Backup:
OSE             CXP102138_1                 R2A20

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

```

3.10 GetIRPVersion

This command displays the Integration Reference Point (IRP) version currently used and a list of supported IRP versions.

The IRP number is included in the bulk CM XML 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 XML configuration files, exported by the SIU as a result of the `upload` command, are using the *currentVersionNumber* (the highest supported IRP version in that software version).

The table below shows the mapping between software release and supported IRP version:

Table 10 Software release and supported IRP version

Software release	IRP version
T10A	5.0
T10B	5.1
T11A	5.2
T11B	5.5
T12A	5.8
T12B	5.11

3.10.1 Input Parameters

None.

3.10.2 Output Parameters

Table 11 Output Parameters

Parameter Name	Description
<code>currentVersionNumber</code>	Shows the IRP version currently used.
<code>versionNumberSet</code>	List of IRP versions supported by the SIU. Ericsson's <i>versionNumberSet</i> attribute behavior differs from that specified in the 3GPP specification.
<code>status</code>	Indicates either of the following: <code>OperationSucceeded</code> . <code>OperationFailed</code> - followed by the specified or unspecified reason.

3.10.3 Example

```
OSmon> getIRPVersion
5.11 5.11,5.8,5.5,5.2,5.1,5.0,3.6,3.5,3.4,3.2,3.1,3.0
OperationSucceeded
```



The *currentVersionNumber* is "5.11" (leftmost).
 The *versionNumberSet* is "5.11,5.8,5.5,5.2,5.1,5.0,3.6,3.5,3.4,3.3,3.2,3.1,3.0".

3.11 GetOSmonSessions

This command shows the user(s) currently logged in to the SIU.

3.11.1 Input Parameters

None.

3.11.2 Output Parameters

See example.

3.11.3 Example

```
OSmon> getosmonsessions
      ID      type      IP      login time      idle time      last command
  13382      LCT
      22:03:17      00:00:00      getosmonsessi
```

3.12 ShowMACFwTable

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

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.

3.12.1 Input Parameters

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

Table 12 Input Parameters

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



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

3.12.2 Output Parameters

None

3.12.3 Example 1

```
OSmon> showMACFWTable all
```

This command displays all entries in the MAC forwarding table.

3.12.4 Example 2

```
OSmon> showMACFWTable STN=0,EthernetInterface=0
```

This command displays MAC forwarding table entries for the specified MO **EthernetInterface**.

3.12.5 Example 3

```
OSmon> showMACFWTable STN=0,EthernetInterface=0 200
```

This command displays MAC forwarding table entries for the specified MO **EthernetInterface** with specified *tagValue*.

3.13 ClearDynamicMACFWTable

This command removes specified entries in the MAC forwarding table.

3.13.1 Input Parameters

Syntax: `clearDynamicMACFWTable all | [<EthernetInterface>] [<tagValue>]`

Table 13 Input Parameters

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



Parameter Name	Description
EthernetInterface (optional)	If set, only entries in the MAC forwarding table of the specified MO EthernetInterface instance is removed.
tagValue (optional)	The tag value (VLAN Id) of table entries to remove. This parameter is optional and can only be used with the <i>EthernetInterface</i> option.

3.13.2 Output Parameters

None

3.13.3 Example 1

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

3.13.4 Example 2

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

3.13.5 Example 3

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

3.14 GetLinkOAM

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

3.14.1 Input Parameters

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



Table 14 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 in a containment hierarchy that is read.

3.14.2 Output Parameters

Table 15 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). Link Events (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.

3.14.3 Example 1

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



```
=====
Discovery state for local and remote DTE
=====
Local:
  Mode:                PASSIVE
  State:               PASSIVE_WAIT
  Unidirection:       NOT SUPPORTED
  Remote loopback:    NOT SUPPORTED
  Link Events:        NOT SUPPORTED
=====
Remote:
  MAC address:        00:00:00:05:00:00
  Vendor(oui):        00 00 00
  Mode:               PASSIVE
=====
```

3.14.4 Example 2

```
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:
=====
```

3.14.5 Example 3

```
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
=====

```

3.15 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.

3.15.1 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 <1-2042>] [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 16 Input Parameters

Parameter Name	Description
MAC	The destination MAC address of LBMs. Its format follows hh:hh:hh:hh:hh:hh.
mepid MEPID	The value of remote MEP ID which is going to receive LBMs. The <i>MAC</i> and <i>mepid</i> are exclusive.



Parameter Name	Description
domain [TYPE <no-namemac>] DOMAIN_NAME	The related domain name in which the LBMs should be sent. The maintenance domain type needs to be specified if it is <i>no-name</i> or <i>mac</i> .
source-mepid SOURCE_MEPID	The value of the MEP ID of the LBM sender.
vlan VLANID	The value of VLAN ID. It should be included if attachment point is VLAN tagged.
tlv_data_bytes DATA <1-2042>	The number of bytes in TLV data.
tlv_pattern PATTERN <1-4>	The pattern of Test TLV. Patterns 1, 2, 3, and 4 have "abc", "1234", "a1b2c", and "1a2b3c" respectively. The <i>tlv_data_bytes</i> and <i>tlv_pattern</i> should be exclusive.
pcp PCP_PRI <0-7>	The PCP VLAN priority (the p-bit). The range is from 0 to 7 with 7 as default value.
loopback-reply-timeout TIMEOUT <1-10>	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>	The interval between each LBM. The range is between 1 to 10 seconds with 1 second as default.
loopback-count COUNT <1-10>	the number of LBMs to send. Its range is from 1 to 10 with 3 as default.

3.15.2 Output Parameters

Table 17 Output Parameters

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

3.15.3 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)
```



3.16 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.

3.16.1 Input Parameters

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

Table 18 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 <i>no-name</i> or <i>mac</i> .
level LEVEL	Specifies the level of the maintenance domain in which the LTMs should be sent. The parameters <i>level</i> and <i>domain</i> are exclusive.
vlan VLANID	Specifies the value of VLAN ID. It should be included if attachment point is VLAN tagged.

3.16.2 Output Parameters

Table 19 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.

3.16.3 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
```



3.17 Debug

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

3.17.1 Input Parameters

Syntax: `debug on/off`

Table 20 Input Parameters

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

3.17.2 Output Parameters

None.

3.17.3 Example 1

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

3.17.4 Example 2

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

3.18 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, the command "log read" and "log monitor" can be used as aliases.



3.18.1 Input Parameters

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

Table 21 Input Parameters

Parameter Name	Description
<code>-s <system event messages></code>	Specifies whether it is the system, event, or messages log that are the source. Default is "system".
<code>[-f filterspec]</code>	Allows definition of textual search string for which a match is done on the fill event entry (time stamps, process name, and so on).
<code>[-n numlines]</code>	Specifying this option will print the latest numlines entries. If combined with a <code>-f</code> specification it will print the latest numlines matches.

3.18.2 Output Parameters

None.

3.18.3 Example 1

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**.

3.18.4 Example 2

OSmon> `syslog monitor -s event`

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

3.19 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.



3.19.1 Input Parameters

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

Table 22 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 of the dump to be displayed as presented by <code>dump -l</code> .

3.19.2 Output Parameters

None.

3.19.3 Example 1

```
OSmon> dump
```

This command shows the most recent primary dump and the last subsequent secondary dump (if any).

3.19.4 Example 2

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

```
[PRIMARY DUMP ID 0x2]
user called : 1
error code  : 0x33844983
extra       : 0x00000000
restart reason :OAM Ordered Restart
```

This command shows a list on all primary and secondary dump files saved in the node.

3.19.5 Example 3

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

```
[ERROR HANDLER PARAMETERS]
user called : 1
```



```
error code : 0x33844984
extra      : 0x00000000
...
...
...
[End Of Dump]
```

This command shows the requested dump.

3.20 CollectDCGInformation

This command gathers relevant debug/troubleshooting information and print it on screen. The content can also be uploaded to an SFTP server as a tar.gz file.

3.20.1 Input Parameters

Syntax: `collectDCGInformation [<FileRef>]`

Table 23 Input Parameters

Parameter Name	Description
FileRef (optional)	<p>An SFTP URI, including authentication information, that defines where to upload the tar.gz file. The addressing of the file source must follow the syntax:</p> <pre>"sftp://username:password@<remotehost>/<filepath>"</pre> <p>where <remotehost> specifies either the IPv4 address ("a.b.c.d") or the IPv6 address in square brackets ("[a:b::c]") of the file destination host, and <filepath> specifies the directory structure and filename.</p> <p>If omitted, the content is printed on screen.</p> <p>Note: The password must not contain a space or any of the following characters: @, %, ; or ".</p>

3.20.2 Output Parameters

Table 24 Output Parameters

Parameter Name	Description
status	<p>Indicates either of the following:</p> <p>OperationSucceeded.</p> <p>OperationFailed - followed by the specified or unspecified reason.</p>



3.20.3 Example 1

The example below prints the debug/troubleshooting information on screen.

```
OSmon> collectDCGInformation
...
...
OperationSucceeded
```

3.20.4 Example 2

The examples below gathers the debug/troubleshooting information and uploads it to the specified SFTP URI.

```
OSmon> collectDCGInformation sftp://jones:123abc@192.168.69.15
2/home/jones/SIUDCG/siu_dcg.tar.gz
OperationSucceeded
```

```
OSmon> collectDCGInformation sftp://jones:123abc@[2000:ffff:aa
aa::abcd:1234]/SIUDCG/siu_dcg.tar.gz
OperationSucceeded
```

The optional parameter *FileRef* is "sftp://jones:123abc@192.168.69.152/home/jones/SIUDCG/siu_dcg.tar.gz" or "sftp://jones:123abc@[2000:ffff:aaaa::abcd:1234]/home/jones/SIUDCG/siu_dcg.tar.gz".

3.21 Ping

This command can be used to send ICMPv4 echo requests to a remote IPv4 host to test connectivity. Omit the "destination" parameter to list all options available for this command.

3.21.1 Input Parameters

Syntax: `ping <destination>`

Table 25 Input Parameters

Parameter Name	Description
destination	The IPv4 address of the destination host to which echo requests are sent to test the connectivity.



3.21.2 Output Parameters

Table 26 Output Parameters

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

3.21.3 Example

```
OSmon> ping 192.168.1.1
```

3.22 Ping6

This command can be used to send ICMPv6 echo requests to a remote IPv6 host to test connectivity. Omit the "destination" parameter to list all options available for this command.

3.22.1 Input Parameters

Syntax: `ping6 <destination>`

Table 27 Input Parameters

Parameter Name	Description
destination	The IPv6 address of the destination host to which echo requests are sent to test the connectivity.

3.22.2 Output Parameters

Table 28 Output Parameters

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

3.22.3 Example

```
OSmon> ping6 fe80::200:ff:fe00:8
```

3.23 SimplePing

This command is a simplified version of `ping`, used to test if the SIU can reach a particular destination through the IP network from a particular source address.



The SIU sends three echo requests to the defined destination address and waits 1 second for an echo response.

3.23.1 Input Parameters

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

Table 29 Input Parameters

Parameter Name	Description
source address	The IP address from which the echo requests are sent. For the operation to succeed, this address must be one of the IP addresses defined in the SIU.
destination address	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.

3.23.2 Output Parameters

Table 30 Output Parameters

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

3.23.3 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".

3.24 SimplePing6

This command is a simplified version of `ping6`, used to test if the SIU can reach a particular destination through the IPv6 network from a particular source address.



The SIU sends three echo requests to the defined destination address and waits 1 second for an echo response. The time between send requests is 107 ms and the packet size is 56 data bytes.

3.24.1 Input Parameters

Syntax: `simpleping6 <source MO-DN> <destination address>`

Table 31 Input Parameters

Parameter Name	Description
source MO-DN	The MO instance in a containment hierarchy from which the echo requests are sent. For the operation to succeed, the source MO-DN must be one of the following: IPv6Interface, IPv6Address, or IPv6VirtualInterface.
destination address	The IPv6 address of the host to which the three echo requests are sent to test the connectivity.

3.24.2 Output Parameters

Table 32 Output Parameters

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

3.24.3 Example

```
OSmon> simpleping6 STN=0,IPv6Interface=0,IPv6Address=0
2001:db8::1
3/3 received OK
OperationSucceeded
```

The *source MO-DN* is "STN=0,IPv6Interface=0,IPv6Address=0".
The *destination address* is "2001:db8::1".



3.25 Traceroute

This command allows to trace the route to a remote IPv4 host. Omit the "host" parameter to list all options available for this command.

3.25.1 Input Parameters

Syntax: `traceroute <host>`

Table 33 Input Parameters

Parameter Name	Description
host	The IPv4 address of the destination host to trace the route to.

3.25.2 Output Parameters

None.

3.25.3 Example

```
OSmon> traceroute 192.168.1.1
```

3.26 Traceroute6

This command allows to trace the route to a remote IPv6 host. Omit the "host" parameter to list all options available for this command.

3.26.1 Input Parameters

Syntax: `traceroute6 <host>`

Table 34 Input Parameters

Parameter Name	Description
host	The IPv6 address of the destination host to trace the route to.

3.26.2 Output Parameters

None.



3.26.3 Example

```
OSmon> traceroute6 2001:abcd::12
```

3.27 Tracepath

This command can be used to trace the path to a remote IPv4 host discovering MTU along this path. Omit the "destination" parameter to list all options available for this command.

3.27.1 Input Parameters

Syntax: `tracepath <destination>`

Table 35 Input Parameters

Parameter Name	Description
destination	The IPv4 address of the destination host to trace the path to.

3.27.2 Output Parameters

None.

3.27.3 Example

```
OSmon> tracepath 192.168.1.1
```

3.28 Tracepath6

This command can be used to trace the path to a remote IPv6 host discovering MTU along this path. Omit the "destination" parameter to list all options available for this command.

3.28.1 Input Parameters

Syntax: `tracepath6 <destination>`

Table 36 Input Parameters

Parameter Name	Description
destination	The IPv6 address of the destination host to trace the path to.



3.28.2 Output Parameters

None.

3.28.3 Example

```
OSmon> tracepath6 2001:abcd::12
```

3.29 getOSPF

This command is used to print the OSPFv2 database, interfaces, border routers, neighbors, and the OSPFv2 routing table.

3.29.1 Input Parameters

Syntax: `getOSPF <OSPFv2Proc MO-DN> [border-routers | database [asbr-summary | external | max-age | network | nssa-external | router | summary] | interface | neighbor | route]`

Table 37 Input Parameters

Parameter Name	Description
OSPFv2Proc MO-DN	The instance of MO OSPFv2Proc for which information will be displayed. If this parameter is included, only the information for the specified MO-DN will be displayed.
border-routers	Displays the Area Border Routers (ABRs) and Autonomous System Boundary Routers (ASBRs).
interface	Displays interface information for OSPF.
neighbor	Displays information on OSPF neighbors.
route	Displays the OSPF routing table.
database	Displays the OSPF database general information.
database asbr-summary	Displays the ASBR summary link states.
database external	Displays external link states.
database max-age	Displays Link State Advertisements (LSAs) in MaxAge list. It maintains the list of all the LSAs in the database which have reached the max-age which is 3600 seconds.
database network	Displays network link states.



Parameter Name	Description
database nssa-external	Displays Not So Stubby Area (NSSA) external link state.
database router	Displays router link states.
database summary	Displays network summary link states.

3.29.2 Output Parameters

See examples.

3.29.3 Example 1

```
OSmon> getOSPF STN=0,OSPFv2Proc=0 border-routers
OSPF process STN=0,OSPFv2Proc=0 internal Routing Table
```

Codes: i - Intra-area route, I - Inter-area route

```
i 10.11.14.34 [11] via 10.11.14.193,
STN=0,OSPFv2Proc=1,OSPFv2Area=1,OSPFv2Network=1, ABR, ASBR
```

3.29.4 Example 2

```
OSmon> getOSPF STN=0,OSPFv2Proc=1 interface
STN=0,OSPFv2Proc=1,OSPFv2Area=1,OSPFv2Network=1 is up, line protocol
Internet Address 10.1.14.34/27, Area STN=0,OSPFv2Proc=1,OSPFv2Area=1
Process ID 1, Router ID 10.11.14.194, Network Type BROADCAST, Cost 1
Transmit Delay is 1 sec, State DROther, Priority 0
No designated router on this network
No backup designated router on this network
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmission
  No Hellos (Passive interface)
Neighbor Count is 0, Adjacent neighbor count is 0
Crypt Sequence Number is 49
Hello received 0 sent 1, DD received 0 sent 0
LS-Req received 0 sent 0, LS-Upd received 0 sent 0
LS-Ack received 0 sent 0, Discarded 0
```

3.29.5 Example 3

```
OSmon> getOSPF STN=0,OSPFv2Proc=1 neighbor
OSPF process STN=0,OSPFv2Proc=1:
```

Neighbor ID	Pri	State	Dead Time	Address	Interface	Inst
10.11.14.1	1	Full/DR	00:00:39	10.11.14.161	STN=0,OSPFv2Pr	
10.11.14.2	1	Full/DR	00:00:33	10.11.14.193	STN=0,OSPFv2Pr	



3.29.6 Example 4

```
OSmon> getOSPF STN=0,OSPFv2Proc=1 route
OSPF process STN=0,OSPFv2Proc=1:
Codes: C - connected, D - Discard, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2

N2 0.0.0.0/0 [11/1] via 10.11.14.193,
   STN=0,OSPFv2Proc=1,OSPFv2Area=1,OSPFv2Network=1

C 10.1.14.32/27 [10] is directly connected,
  STN=0,OSPFv2Proc=1,OSPFv2Area=1,OSPFv2Network=2

O 10.11.14.0/27 [11] via 10.11.14.161,
  STN=0,OSPFv2Proc=1,OSPFv2Area=1,OSPFv2Network=1
                                     via 10.11.14.193,
  STN=0,OSPFv2Proc=1,OSPFv2Area=1,OSPFv2Network=2
```

3.29.7 Example 5

```
OSmon> getOSPF STN=0,OSPFv2Proc=1 database
OSPF Router with ID (10.11.14.194) (Process ID STN=0,OSPFv2Proc=1)

          Router Link States (Area STN=0,OSPFv2Proc=1,OSPFv2Area=1)
[NSSA])
Link ID   ADV Router   Age   Seq#           CkSum  Link count
10.11.14.1 10.11.14.1   1729 0x80000017    0xcd04 3
10.11.14.2 10.11.14.2   1734 0x80000017    0xc846 3

          Net Link States (Area STN=0,OSPFv2Proc=1,OSPFv2Area=1)
[NSSA])
Link ID   ADV Router   Age   Seq#           CkSum
10.11.14.2 10.11.14.2   948 0x80000007    0x9029
10.11.14.34 10.11.14.34 1129 0x80000007    0xdd7a

          Summary Link States (Area STN=0,OSPFv2Proc=1,OSPFv2Area=1)
[NSSA])
Link ID   ADV Router   Age   Seq#           CkSum  Route
10.11.14.128 10.11.14.34 1050 0x80000007    0xe97c 10.11.14.128/27

          NSSA-external Link States (Area STN=0,OSPFv2Proc=1,OSPFv2Area=1)
[NSSA])
Link ID   ADV Router   Age   Seq#           CkSum  Route           Ta
0.0.0.0   10.11.14.34 945 0x80000007    0x1f47 E2 0.0.0.0/0     0
192.168.2.0 10.11.14.194 1746 0x80000007    0xede5 E1 192.168.2.0/24 0
```



3.29.8 Example 6

```
OSmon> getOSPF STN=0,OSPFv2Proc=0 database external
  OSPF Router with ID (10.11.14.194) (Process ID STN=0,OSPFv2Proc=0)

AS External Link States

LS age: 298
Options: 0x2 (*|---|E|)
LS Type: AS-external-LSA
Link State ID: 10.10.100.0 (External Network Number)
Advertising Router: 10.10.11.50
LS Seq Number: 80000001
Checksum: 0x7033
Length: 36
Network Mask: /24
  Metric Type: 2 (Larger than any link state path)
  TOS: 0
  Metric: 20
  Forward Address: 10.10.11.50
  External Route Tag: 0
```

3.29.9 Example 7

```
OSmon> getOSPF STN=0,OSPFv2Proc=0 database network
  OSPF Router with ID (10.11.14.194) (Process ID STN=0,OSPFv2Proc=0)

Net Link States (Area STN=0,OSPFv2Proc=1,OSPFv2Area=1)

LS age: 1835
Options: 0x28 (-|DC|N/P|)
LS Type: network-LSA
Link State ID: 10.11.14.2 (address of Designated Router)
Advertising Router: 10.11.14.2
LS Seq Number: 80000008
Checksum: 0x8e2a
Length: 32
Network Mask: /27
  Attached Router: 10.11.14.2
  Attached Router: 10.11.14.1
```

3.29.10 Example 8

```
OSmon> getOSPF STN=0,OSPFv2Proc=0 database nssa-external
  OSPF Router with ID (10.11.14.194) (Process ID STN=0,OSPFv2Proc=0)

NSSA-external Link States (Area STN=0,OSPFv2Proc=1,OSPFv2Area=1 [NS
LS age: 700
```



```
Options: 0x20 (-|-|DC|-|-|-|-|-)
LS Type: AS-NSSA-LSA
Link State ID: 0.0.0.0 (External Network Number For NSSA)
Advertising Router: 10.11.14.34
LS Seq Number: 80000009
Checksum: 0x1b49
Length: 36
Network Mask: /0
    Metric Type: 2 (Larger than any link state path)
    TOS: 0
    Metric: 1
    NSSA: Forward Address: 0.0.0.0
    External Route Tag: 0
```

3.29.11

Example 9

```
OSmon> getOSPF STN=0,OSPFv2Proc=0 database router
  OSPF Router with ID (10.11.14.194) (Process ID STN=0,OSPFv2Proc=0)

Router Link States (Area STN=0,OSPFv2Proc=1,OSPFv2Area=1 [NSSA])
LS age: 311
Options: 0x28 (-|-|DC|-|N/P|-|-|-)
Flags: 0x0
LS Type: router-LSA
Link State ID: 10.11.14.1
Advertising Router: 10.11.14.1
LS Seq Number: 8000001a
Checksum: 0xc707
Length: 60
  Number of Links: 3

  Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.11.14.2
    (Link Data) Router Interface address: 10.11.14.1
    Number of TOS metrics: 0
    TOS 0 Metric: 1

  Link connected to: a Transit Network
    (Link ID) Designated Router address: 10.11.14.98
    (Link Data) Router Interface address: 10.11.14.97
    Number of TOS metrics: 0
    TOS 0 Metric: 1
```

3.29.12

Example 10

```
OSmon> getOSPF STN=0,OSPFv2Proc=0 database summary
  OSPF Router with ID (10.11.14.194) (Process ID STN=0,OSPFv2Proc=0)
```



```

Summary Link States (Area STN=0,OSPFv2Proc=1,OSPFv2Area=1 [NSSA])
LS age: 1688
Options: 0x28 (-|-|DC|-|N/P|-|-|-)
LS Type: summary-LSA
Link State ID: 10.11.14.128 (summary Network Number)
Advertising Router: 10.11.14.34
LS Seq Number: 80000009
Checksum: 0xe57e
Length: 28
Network Mask: /27
          TOS: 0 Metric: 1

```

3.30 TraceAllMOAttributeInLog

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

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

3.30.1 Input Parameters

None.

3.30.2 Output Parameters

Table 38 Output Parameters

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

3.30.3 Example

```

OSmon> traceallmoattributeinlog
OperationSucceeded

```

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



3.31 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.

The result of a test shows the network quality and packet loss and is cleared after the result is reported.

3.31.1 Input Parameters

Syntax: `timeservertest start [<instanceId>] | result`

Table 39 Input Parameters

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

3.31.2 Output Parameters

None.

3.31.3 Examples

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

3.32 Uptime

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

3.32.1 Input Parameters

None.



3.32.2 Output Parameters

Table 40 Output Parameters

Parameter Name	Description
Uptime	Specifies the uptime for the system.

3.32.3 Example

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

3.33 GetTemperature

This command displays the temperature from different sources in the unit.

3.33.1 Input Parameters

None.

3.33.2 Output Parameters

Table 41 Output Parameters

Parameter Name	Description
Temp sensor <x>	Specifies the temperature for the sensor.

3.33.3 Example

```
OSmon> getTemperature
Network processor: 78 C, 173 F
Temp sensor 1 (near FPGA): 46 C, 114 F
Temp sensor 2 (near OCXO): 46 C, 114 F
OperationSucceeded
```

3.34 GetAlarmList

This command returns a list of all active alarms.

3.34.1 Input Parameters

None.



3.34.2 Output Parameters

Table 42 Output Parameters

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

3.34.3 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
```

3.35 ResetToFactorySetting

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

Note: The factory default configuration for the SIU includes MO instances used by the auto integration process. When this command is executed these MO instances are created and the auto integration process is automatically started. To abort the auto integration process, all MO instances related to this function have to be manually removed. See Reference [6].

3.35.1 Input Parameters

None.



3.35.2 Output Parameters

Table 43 Output Parameters

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

3.35.3 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)? ok
OperationSucceeded
```

```
(ok/cancel)? cancel
OperationFailed - Command interrupted
```

3.36 Restart

This command initializes a restart of the unit. Any established traffic connections are closed as part of the operation. The restart also clears PM data and terminates O&M traffic and Telnet/SSH sessions in progress. If an alarm 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.

3.36.1 Input Parameters

None.



3.36.2 Output Parameters

Table 44 Output Parameters

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

3.36.3 Example

```
OSmon> restart  
OperationSucceeded
```

3.37 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).

3.37.1 Input Parameters

Syntax: `uselocalsftp on/off`

Table 45 Input Parameters

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

3.37.2 Output Parameters

None.

3.37.3 Example

```
OSmon> uselocalsftp on
```



4 Basic CM Transaction

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

4.1 StartTransaction

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

4.1.1 Input Parameters

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

Table 46 Input Parameters

Parameter Name	Description
transactionId	<p>Unique transaction identifier.</p> <p>Valid value: a string of up to 10 characters.</p> <p>Refer to Section 2.4 on page 3 for a description of restricted keywords that cannot be used as value for this parameter.</p>
ossIpAddress (optional)	<p>The IP address of the SNMP manager to which notifications are sent.</p> <p>If used, this IP address is only used as a trap receiver while the transaction exists. When the command <code>endTransaction</code> is given, traps will no longer be sent to this address.</p> <p>If omitted, no notification is sent.</p>



4.1.2 Output Parameters

Table 47 Output Parameters

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

4.1.3 Example

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

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

4.2 CreateMO

This command creates a new MO instance. The command must be used within a basic CM transaction and does not support setting of attribute values. The creation of the new MO instance will take affect when a `commit` command is executed.

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

4.2.1 Input Parameters

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

Table 48 Input Parameters

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



4.2.2 Output Parameters

Table 49 Output Parameters

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

4.2.3 Example

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

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

4.3 DeleteMO

This command deletes a MO instance. The command must be used within a basic CM transaction. It is only possible to delete manually created MO instances and any child MO instances are automatically deleted. The deletion of the MO instance will take affect when a `commit` command is executed.

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

4.3.1 Input Parameters

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

Table 50 Input Parameters

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



4.3.2 Output Parameters

Table 51 Output Parameters

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

4.3.3 Example

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

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

4.4 GetMOAttribute

This command displays MO attribute values.

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

4.4.1 Input Parameters

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



Table 52 Input Parameters

Parameter Name	Description
transactionId (optional)	<p>Unique transaction identifier.</p> <p>Valid value: a string of up to 10 characters.</p> <p>Optional parameter when the command is executed within a basic CM transaction. The value returned will in this case be the value stored in the RAM which might not be the same as the currently used value. Changing MO attributes with the <code>setMOAttribute</code> command creates a temporary copy of the MO model in the RAM. The values in the RAM are used after the next <code>commit</code> command.</p> <p>This parameter is not used when the command is executed outside a basic CM transaction. The returned value will in this case be the currently used value stored in the flash memory.</p>
MO-DN (optional)	MO instance in a containment hierarchy that is read. Can be either a single MO instance or all instances of the specified MO.
-c (optional)	The -c flag is used when values from both normal attributes and counters should be displayed.
attributeId (optional)	Identifies the attribute to be returned.

4.4.2 Output Parameters

Table 53 Output Parameters

Parameter Name	Description
attributeIdOut	Values for the optional MO instance <i>attributeId</i> specified as the input parameter.
status	<p>Indicates either of the following:</p> <p><code>OperationSucceeded</code>.</p> <p><code>OperationFailed</code> - followed by the specified or unspecified reason.</p>

4.4.3 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 committed MO model is returned. A changed value will be in use after execution of the **commit** command.

4.4.4 Example 2

The two examples below show how to get the value of a specific attribute in all configured instances of a specific MO-DN.

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

```
OSmon> getMOAttribute STN=0,VLANGroup=all depLinkLayer
STN=0,VLANGroup=WAN; depLinkLayer= STN=0,EthernetInterfac
e=WAN;
STN=0,VLANGroup=LAN; depLinkLayer= STN=0,EthernetInterfa
ce=Port3;
OperationSucceeded
```

4.4.5 Example 3

In this example the optional *flag -c* is used to get normal attributes and counters. If the command is used without flag, 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= CXP102138_1;
STN=0; SW_PrimaryProductRevision= R2B04;
STN=0; SW_BackupProductNumber= CXP102138_1;
STN=0; SW_BackupProductRevision= R2B04;
STN=0; CPU_Load= 8;
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; systemClockPTPDomainNumber= -1;
STN=0; failedPMUploads= 0;
STN=0; userLabel= ;
STN=0; lastConfigChange= ;
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;
...
...
OperationSucceeded

```

4.4.6 Example 4

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

```

4.5 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 basic CM 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 about MO classes and their attributes/values, see Reference [4].

4.5.1 Input Parameters

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

Table 54 Input Parameters

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



4.5.2 Output Parameters

Table 55 Output Parameters

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

4.5.3 Example

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

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

4.6 CheckConsistency

This command makes a validation and consistency check of the complete MIB, resulting from configuration changes in the specific transaction. The command must be used within a basic CM transaction. The check is useful before a `commit` command is performed to verify that the configuration contains a complete set of MOs and attributes/values.

4.6.1 Input Parameters

Syntax: `checkConsistency <transactionId>`

Table 56 Input Parameters

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



4.6.2 Output Parameters

Table 57 Output Parameters

Parameter Name	Description
consistent	This parameter contains the result of the consistency check. The result is <code>consistent</code> or <code>notConsistent</code> .
status	Indicates either of the following: <code>OperationSucceeded</code> . <code>OperationFailed</code> - followed by the specified or unspecified reason.

4.6.3 Example 1

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

The *transactionId* is "trans1".

4.6.4 Example 2

```
OSmon> checkConsistency trans1  
notConsistent "IPInterface.primaryIP/SubnetMask:invalid  
IP"  
OperationSucceeded
```

The *transactionId* is "trans1".

4.7 Commit

This command commits all configuration requests in a basic CM transaction. When adding/deleting MO classes, and changing one or more attribute values, a temporary copy of the MO model is created that includes all the changes. The `commit` command is required to activate this temporary version of the MO model and make it the currently used and persistent MO model. The `checkConsistency` command can first be performed to check the validity and consistency of an updated configuration.

4.7.1 Input Parameters

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



Table 58 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 during the restart. If not set, and attributes requiring a restart are changed, the <code>commit</code> command is rejected and has to be reissued with this parameter.
autoRollback <autoRollbackTimer> (optional)	If set, the unit will restart on the old configuration if the command <code>endTransaction</code> has not been received within the specified time (<i>autoRollbackTimer</i>) in minutes.

4.7.2 Output Parameters

Table 59 Output Parameters

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

4.7.3 Example 1

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

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

4.7.4 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.



4.8 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.

4.8.1 Input Parameters

Syntax: `endTransaction <transactionId>`

Table 60 Input Parameters

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

4.8.2 Output Parameters

Table 61 Output Parameters

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

4.8.3 Example

```
OSmon> endTransaction trans1  
OperationSucceeded
```

The *transactionId* is "trans1".

4.9 GetContainment

This command retrieves all MO instances in a specific MO containment hierarchy below, and including, the specified *MO-DN*.

4.9.1 Input Parameters

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



Table 62 Input Parameters

Parameter Name	Description
transactionId (optional)	<p>Unique transaction identifier.</p> <p>Valid value: a string of up to 10 characters.</p> <p>Optional parameter when the command is executed within a basic CM transaction. The value returned will in this case be the value stored in the RAM which might not be the same as the currently used value. Changing MO attributes using the <code>setMOAttribute</code> command creates a temporary copy of the MO model in the RAM. The values in the RAM are used after the next <code>commit</code> command.</p> <p>This parameter is not used when the command is executed outside a basic CM transaction. The returned value will in this case be the currently used value stored in the flash memory.</p>
MO-DN	MO instance in a containment hierarchy.

4.9.2 Output Parameters

Table 63 Output Parameters

Parameter Name	Description
containment	All MO instances included in the specified containment hierarchy including the MO defined with the <code>MO-DN</code> parameter.
status	<p>Indicates either of the following:</p> <p><code>OperationSucceeded</code>.</p> <p><code>OperationFailed</code> - followed by the specified or unspecified reason.</p>

4.9.3 Example

```
OSmon> getContainment trans1 STN=0
```

```
STN=0;
STN=0,AutoIntegration=0;
STN=0,Equipment=0;
STN=0,EthernetInterface=AutoInt-1;
STN=0,EthernetInterface=AutoInt-2;
STN=0,IPInterface=AutoInt-1;
STN=0,IPInterface=AutoInt-2;
```



```

STN=0,MeasurementDefinition=0;
STN=0,RoutingTable=0;
STN=0,RoutingTable=0,IpRouteSys=AutoInt-1;
STN=0,RoutingTable=0,IpRouteSys=AutoInt-2;
STN=0,Synchronization=0;
STN=0,VLANGroup=AutoInt-1;
STN=0,VLANGroup=AutoInt-1,VLAN=AutoInt-1;
STN=0,VLANGroup=AutoInt-2;
STN=0,VLANGroup=AutoInt-2,VLAN=AutoInt=2;
OperationSucceeded

```

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

4.10 GetCounters

This command displays counters for the specified MO instance.

4.10.1 Input Parameters

Syntax: `getcounters <MO-DN>`

Table 64 Input Parameters

Parameter Name	Description
MO-DN	MO instance in a containment hierarchy to be read.

4.10.2 Output Parameters

Table 65 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).

4.10.3 Example

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



```

=====
Counter                | Value                | Change (2.6s)
=====
ifHCInOctets           | 785810115           | 1.17 Mbps
ifHCOctets             | 621870824           | 926.99 kbps
ifInErrors              | 0                    |
ifOutErrors            | 0                    |
ifHCInUcastPkts       | 654819              | +502
ifHCOctets             | 517464              | +394
ifInDiscards           | 0                    |
ifOutDiscards          | 0                    |
ifHCInBroadcastPkts   | 13354               | +2
ifHCOctets             | 13                  |
ifHCInMulticastPkts   | 234                 |
ifHCOctets             | 0                   |
ifInUnknownProtos     | 440                 |
globalFrameDiscards    | 0                   |
=====

```

4.11 GetTransactionId

This command returns the identity of an existing basic CM transaction.

4.11.1 Input Parameters

None.

4.11.2 Output Parameters

Table 66 Output Parameters

Parameter Name	Description
transactionId	Specifies the unique transaction identifier. If no transaction exists, this is indicated with --.
status	Indicates either of the following: OperationSucceeded. OperationFailed - followed by the specified or unspecified reason.

4.11.3 Example

```

OSmon> getTransactionId
OSmon session ID          transaction ID
           241             trans1
OperationSucceeded

```



The output parameter *transactionId* is "trans1".

4.12 GetTransactionStatus

This command returns the status for the specified transaction. The command must be used within a basic CM transaction.

4.12.1 Input Parameters

Syntax: `getTransactionStatus <transactionId>`

Table 67 Input Parameters

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

4.12.2 Output Parameters

Table 68 Output Parameters

Parameter Name	Description
transactionState	Possible values: "Idle", "CommitFailed", "CommitCompleted". The transaction state is "Idle" when the <code>startTransaction</code> command is performed but the <code>commit</code> command is not yet performed.
status	Indicates either of the following: OperationSucceeded. OperationFailed - followed by the specified or unspecified reason.

4.12.3 Example

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

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



5 Bulk CM Session

This chapter lists bulk CM session commands which are related to the download and upload of bulk CM XML configuration files, backup and restore of configuration, and 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.

5.1 StartSession

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

5.1.1 Input Parameters

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

Table 69 Input Parameters

Parameter Name	Description
sessionId	<p>Unique session identifier.</p> <p>Valid value: a string of up to 10 characters.</p> <p>Refer to Section 2.4 on page 3 for a description of restricted keywords that cannot be used as value for this parameter.</p>
ossIpAddress (optional)	<p>The IP address of the SNMP manager to which notifications are sent.</p> <p>If used, this IP address is only used as a trap receiver while the session exists. When the command <code>endSession</code> is given, traps will no longer be sent to this address.</p> <p>If omitted, no notification is sent.</p>



5.1.2 Output Parameters

Table 70 Output Parameters

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

5.1.3 Example

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

The *sessionId* is "bcm1".

The optional parameter *osslpAddress* is "192.168.59.62".

5.2 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.

5.2.1 Input Parameters

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

Table 71 Input Parameters

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



Parameter Name	Description
FileRef	<p>An SFTP URI, including authentication information, that defines from where the configuration file is retrieved and downloaded. The addressing of the file source must follow the syntax:</p> <pre>"sftp://username:password@<remotehost>/<filepath>"</pre> <p>where <remotehost> specifies either the IPv4 address ("a.b.c.d") or the IPv6 address in square brackets ("[a:b::c]") of the file source host, and <filepath> specifies the directory structure and filename.</p> <p>Note: The password must not contain a space or any of the following characters: @, %, ; or ".</p>
progressInterv (optional)	<p>The time (in seconds) after which the SIU must send notify session state changed notifications to the SNMP manager defined in command startSession.</p> <p>Valid values: 0 and 15–900.</p> <p>If set to "0" or omitted, only final notifications indicating change of state are sent.</p>

5.2.2 Output Parameters

Table 72 Output Parameters

Parameter Name	Description
status	<p>Indicates either of the following:</p> <p>OperationSucceeded.</p> <p>OperationFailed - followed by the specified or unspecified reason.</p>

5.2.3 Example

```
OSmon> download bcm1 sftp://jones:123abc@192.168.69.152/home/
jones/SIUCFG/siu.xml 15
OperationSucceeded
```

```
OSmon> download bcm1 sftp://jones:123abc@[2000:ffff:aaaa::abcd:1
234]/home/jones/SIUCFG/siu.xml 15
OperationSucceeded
```



The *sessionId* is "bcm1".

The *FileRef* is "sftp://jones:123abc@192.168.69.152/home/jones/SIUCFG/siu.xml" or "sftp://jones:123abc@[2000:ffff:aaaa::abcd:1234]/home/jones/SIUCFG/siu.xml".

The optional parameter *progressInterv* is "15" (seconds).

5.3 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.

5.3.1 Input Parameters

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

Table 73 Input Parameters

Parameter Name	Description
sessionId	Unique session identifier. Valid value: a string of up to 10 characters.
FileRef	An SFTP URI, including authentication information, that defines where to upload the configuration file. The addressing of the file source must follow the syntax: "sftp://username:password@<remotehost>/<filepath>" where <remotehost> specifies either the IPv4 address ("a.b.c.d") or the IPv6 address in square brackets ("[a:b::c]") of the file destination host, and <filepath> specifies the directory structure and filename. Note: The password must not contain a space or any of the following characters: @, %, ; or ".



Parameter Name	Description
fileContent	<p>Defines which data from the MIB that should be uploaded.</p> <p>Valid value:"CM"</p> <p>The value "CM" results in uploading of configuration data from all MO classes other than MO MeasuredMOClass and its child objects.</p>
progressInterv (optional)	<p>The time (in seconds) after which the SIU must send notify session state changed notifications to the SNMP manager defined in command startSession.</p> <p>Valid values: 0 and 15–900.</p> <p>If set to "0" or omitted, only final notifications indicating change of state are sent.</p>

5.3.2 Output Parameters

Table 74 Output Parameters

Parameter Name	Description
status	<p>Indicates either of the following:</p> <p>OperationSucceeded.</p> <p>OperationFailed - followed by the specified or unspecified reason.</p>

5.3.3 Example

```
OSmon> upload bcm1 sftp://jones:123abc@192.168.69.152/home/jones/SIUCFG/siu.xml CM 30
OperationSucceeded
```

```
OSmon> upload bcm1 sftp://jones:123abc@[2000:ffff:aaaa::abcd:1234]/home/jones/SIUCFG/siu.xml CM 30
OperationSucceeded
```

The *sessionId* is "bcm1".

The *FileRef* is "sftp://jones:123abc@192.168.69.152/home/jones/SIUCFG/siu.xml" or "sftp://jones:123abc@[2000:ffff:aaaa::abcd:1234]/home/jones/SIUCFG/siu.xml".

The *fileContent* is "CM".

The optional parameter *progressInterv* is "30".



5.4 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.

If the **-u** flag is added to the command, no password should be provided and the file will be uploaded as a clear text backup file.

Only MIM configuration is included into the backup (no PBOOT parameters or software). The configuration includes secure data (like passwords).

5.4.1 Input Parameters

Syntax: `backup <sessionId> <FileRef> <passwd>|-u [<progressInterv>]`

Table 75 Input Parameters

Parameter Name	Description
sessionId	Unique session identifier. Valid value: a string of up to 10 characters.
FileRef	An SFTP URI, including authentication information, that defines where to upload the configuration file. The addressing of the file destination must follow the syntax: "sftp://username:password@<remotehost>/<filepath>" where <remotehost> specifies either the IPv4 address ("a.b.c.d") or the IPv6 address in square brackets ("[a:b::c]") of the file destination host, and <filepath> specifies the directory structure and filename. Note: The password must not contain a space or any of the following characters: @, %, ; or ".
passwd	The password (1-20 characters) to be used for encryption of the configuration file. The same password is then used when performing a restore of the configuration. Refer to Section 2.4 on page 3 for a description of restricted keywords that cannot be used as value for this parameter.



Parameter Name	Description
-u	If the -u flag is used, the configuration file will not be encrypted before uploading. Cannot be used together with the <i>passwd</i> parameter. The -u flag must always be at the end of the command line.
progressInterv (optional)	The time (in seconds) after which the SIU must send notify session state changed notifications to the SNMP manager defined in command startSession . Valid values: 0 and 15–900. If set to "0" or omitted, only final notifications indicating change of state are sent.

5.4.2 Output Parameters

Table 76 Output Parameters

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

5.4.3 Example 1

The following examples will upload a clear text backup file:

```
OSmon> backup bcm1 sftp://jones:123abc@192.168.69.152/home/jones
/SIUCFG/siu.xml -u
OperationSucceeded
```

```
OSmon> backup bcm1 sftp://jones:123abc@[2000:ffff:aaaa::abcd:123
4]/home/jones/SIUCFG/siu.xml -u
OperationSucceeded
```

```
OSmon> backup bcm1 sftp://jones:123abc@192.168.69.152/home/jon
es/SIUCFG/siu.xml 30 -u
OperationSucceeded
```

The *sessionId* is "bcm1".

The *FileRef* is "sftp://jones:123abc@192.168.69.152/home/jones/SIUCFG/siu.xml" or "sftp://jones:123abc@[2000:ffff:aaaa::abcd:1234]/home/jones/SIUCFG/siu.xml"



The `-u` flag indicates that the backup file should not be encrypted before uploading.

The optional parameter `progressInterv` is "30".

5.4.4 Example 2

The following examples will upload an encrypted backup file:

```
OSmon> backup bcm1 sftp://jones:123abc@192.168.69.152/home/jones/SIUCFG/siu.xml admin
OperationSucceeded
```

```
OSmon> backup bcm1 sftp://jones:123abc@[2000:ffff:aaaa::abcd:1234]/home/jones/SIUCFG/siu.xml admin
OperationSucceeded
```

```
OSmon> backup bcm1 sftp://jones:123abc@192.168.69.152/home/jones/SIUCFG/siu.xml admin 30
OperationSucceeded
```

The `sessionId` is "bcm1".

The `FileRef` is "sftp://jones:123abc@192.168.69.152/home/jones/SIUCFG/siu.xml" or "sftp://jones:123abc@[2000:ffff:aaaa::abcd:1234]/home/jones/SIUCFG/siu.xml"

The `passwd` is "admin" and indicates that the backup file should be encrypted before uploading.

The optional parameter `progressInterv` is "30".

5.5 Restore

This command is used to restore a backup configuration file that first has been downloaded using the `download` command.

Note: A restore is manually executed in two steps; first downloading the configuration file and then restoring it using this command.

During restore, the downloaded backup configuration file is decrypted using the same password that was used when the file was uploaded (using the `backup` command) and then an attempt to activate it is done.

If the `-u` flag is added to the command, it is assumed that the backup configuration file was uploaded as a clear text file (not encrypted).

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.

5.5.1 Input Parameters

Syntax: `restore <sessionId> <passwd>|-u [-d]|[-c]`



Table 77 Input Parameters

Parameter Name	Description
sessionId	Unique session identifier. Valid value: a string of up to 10 characters.
passwd	The password to be used when the downloaded backup configuration file is being decrypted. This password is the same as the one used when the file was uploaded using the <code>backup</code> command.
-u	If the <code>-u</code> flag is used, the backup configuration file is assumed to be in clear text (not encrypted). Cannot be used together with the <code>passwd</code> parameter. The <code>-u</code> flag must always be at the end of the command line.
-d (optional)	If the <code>-d</code> flag (delayed) is set, the restored file is only stored persistently. The activation of the restored file will be performed when the SW session is activated (command <code>activateSW</code>) with the optional parameter <code>useNewConfiguration</code> 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.
-c (optional)	If the <code>-c</code> flag (checkOnly) is set, only a message containing configuration version and ability to restore on current SW will be printed.

5.5.2

Output Parameters

Table 78 Output Parameters

Parameter Name	Description
status	Indicates either of the following: <code>OperationSucceeded</code> . <code>OperationFailed</code> - followed by the specified or unspecified reason.



5.5.3 Example 1

The examples below will restore a clear text backup file.

```
OSmon> restore bcm1 -u  
OperationSucceeded
```

```
OSmon> restore bcm1 -c -u  
OperationSucceeded
```

```
OSmon> restore bcm1 -d -u  
OperationSucceeded
```

The *sessionId* is "bcm1".

The *-u* flag indicates that the backup configuration file is in clear text (not encrypted).

The optional *-c* flag is included to only get a message containing configuration version and the ability to restore on current configuration.

The optional *-d* flag is included if the activation of the restored file should be performed within a SW session (*activateSW*) with the optional parameter *useNewConfiguration* set.

5.5.4 Example 2

The examples below will restore an encrypted backup file.

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

```
OSmon> restore bcm1 admin -c  
OperationSucceeded
```

```
OSmon> restore bcm1 admin -d  
OperationSucceeded
```

The *sessionId* is "bcm1".

The *passwd* is "admin".

The optional *-c* flag is included to only get a message containing configuration version and the ability to restore on current configuration.

The optional *-d* flag is included if the activation of the restored file should be performed within a SW session (*activateSW*) with the optional parameter *useNewConfiguration* set.

5.6 Activate

This command activates a new bulk CM XML configuration file. A consistency check of the complete MIB, resulting from configuration changes in the specific session, is performed. Bulk CM session data is stored persistently. The command can cause different types of restarts depending on the restart type of the changed attributes.



In case any changed attributes require a restart, any established traffic connections are closed and the unit restarts. 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.

5.6.1 Input Parameters

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

Table 79 Input Parameters

Parameter Name	Description
sessionId	Unique session identifier. Valid value: a string of up to 10 characters.
delayed (optional)	If set, the configuration file is only stored persistently. The activation of the configuration file will be performed when the SW session is activated (using the command <code>activateSW</code>) with the optional parameter <code>useNewConfiguration</code> set. If not set, the configuration file will be activated immediately. The parameter is not possible to use in the same session as the parameter <code>autoRollback</code> . Note: Activation 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 the command <code>endSession</code> has been received within the specified time (<code>autoRollbackTimer</code>) in minutes.



5.6.2 Output Parameters

Table 80 Output Parameters

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

5.6.3 Example 1

```
OSmon> activate bcm1  
OperationSucceeded
```

5.6.4 Example 2

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

The *sessionId* is "bcm1"

The optional parameter *delayed* is included to ensure the activation of the configuration file should be performed within a SW session (`activateSW`) with the optional parameter *useNewConfiguration* set.

5.6.5 Example 3

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

The *sessionId* is "bcm1"

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

5.7 EndSession

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

5.7.1 Input Parameters

Syntax: `endsession <sessionId>`



Table 81 Input Parameters

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

5.7.2 Output Parameters

Table 82 Output Parameters

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

5.7.3 Example

```
OSmon> endSession bcm1
OperationSucceeded
```

The *sessionId* is "bcm1".

5.8 GetPendingConfiguration

This command is used to determine if there exists a pending bulk CM XML configuration file ready to be activated with the `activateSW` command (together with the downloaded software). If a file exists, a filename and MIM version is returned.

5.8.1 Input Parameters

None.

5.8.2 Output Parameters

Table 83 Output Parameters

Parameter Name	Description
filename	The filename of the bulk CM XML configuration file excluding directory information. If no file exists, this is indicated with --.



Parameter Name	Description
mimVersion	The MIM version from the bulk CM file header.
status	Indicates either of the following: OperationSucceeded. OperationFailed - followed by the specified or unspecified reason.

5.8.3 Example

```
OSmon> getPendingConfiguration  
siu82_CM.xml 5.11  
OperationSucceeded
```

The output parameter *filename* is "siu82_CM.xml".
The output parameter *mimVersion* is "5.11".

5.9 GetSessionIds

This command returns the identity of existing bulk CM sessions.

5.9.1 Input Parameters

None

5.9.2 Output Parameters

Table 84 Output Parameters

Parameter Name	Description
sessionIdList	This parameter shows a list of session Id(s) for existing bulk CM session(s). If no CM sessions exist, this is indicated with --.
status	Indicates either of the following: OperationSucceeded. OperationFailed - followed by the specified or unspecified reason.

5.9.3 Example

```
OSmon> getSessionIds  
bcm1  
OperationSucceeded
```



The output parameter *sessionIdList* is "bcm1".

5.10 GetSessionStatus

This command returns the status of the specified bulk CM session.

5.10.1 Input Parameters

Syntax: `getsessionstatus <sessionId>`

Table 85 Input Parameters

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

5.10.2 Output Parameters

Table 86 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: OperationSucceeded. OperationFailed - followed by the specified or unspecified reason.

5.10.3 Example

```
OSmon> getSessionStatus bcm1
DownloadInProgress
OperationSucceeded
```

The *sessionId* is "bcm1".

The output parameter *sessionState* is "DownloadInProgress".



5.11 AbortSessionOperation

This command interrupts the specified bulk CM session. Any downloaded data is discarded. No uploaded data is deleted.

5.11.1 Input Parameters

Syntax: `abortsessionoperation <sessionId>`

Table 87 Input Parameters

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

5.11.2 Output Parameters

Table 88 Output Parameters

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

5.11.3 Example

```
OSmon> abortSessionOperation bcm1  
OperationSucceeded
```

The *sessionId* is "bcm1".

6 Software Session

This chapter lists commands related to the downloading of software files and the activation of new software. The downloading and activation of software are performed in a software session.



6.1 StartSWSession

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

6.1.1 Input Parameters

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

Table 89 Input Parameters

Parameter Name	Description
sessionIdSW	<p>Unique session identifier.</p> <p>Valid value: a string of up to 10 characters.</p> <p>Refer to Section 2.4 on page 3 for a description of restricted keywords that cannot be used as value for this parameter.</p>
ossIpAddress (optional)	<p>The IP address of the SNMP manager to which notifications are sent.</p> <p>If used, this IP address is only used as a trap receiver while the SW session exists. When the command <code>endSWSession</code> is given, traps will no longer be sent to this address.</p> <p>If omitted, no notification is sent.</p>

6.1.2 Output Parameters

Table 90 Output Parameters

Parameter Name	Description
status	<p>Indicates either of the following:</p> <p><code>OperationSucceeded</code>.</p> <p><code>OperationFailed</code> - followed by the specified or unspecified reason.</p>

6.1.3 Example

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

The `sessionIdSW` is "s1".

The optional parameter `ossIpAddress` is "192.168.59.62".



6.2 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 software 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.

6.2.1 Input Parameters

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

Table 91 Input Parameters

Parameter Name	Description
sessionIdSW	Unique session identifier. Valid value: a string of up to 10 characters.
FileRef	An SFTP URI, including authentication information, that defines from where the software is retrieved and downloaded. The addressing of the file source must follow the syntax: "sftp://username:password@<remotehost>/<filepath>" where <remotehost> specifies either the IPv4 address ("a.b.c.d") or the IPv6 address in square brackets ("[a:b::c]") of the file source host, and <filepath> specifies the directory structure and filename. Note: The password must not contain a space or any of the following characters: @, %, ; or "."
progressInterv (optional)	The time (in seconds) after which the SIU must send notify SW session state changed notifications to the SNMP manager defined in command <code>startSWSession</code> . Valid values: 0 and 15–900. If set to "0" or omitted, no notifications are sent.



6.2.2 Output Parameters

Table 92 Output Parameters

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

6.2.3 Example

```
OSmon> downloadSW s1 sftp://jones:123abc@192.168.69.152/home/jones/SIUSW/siu.tar.gz 30
OperationSucceeded
```

```
OSmon> downloadSW s1 sftp://jones:123abc@[2000:ffff:aaaa::abcd:1234]/home/jones/SIUSW/siu.tar.gz 30
OperationSucceeded
```

The *sessionIdSW* is "s1".

The *FileRef* is "sftp://jones:123abc@192.168.69.152/home/jones/SIUSW/siu.tar.gz" or "sftp://jones:123abc@[2000:ffff:aaaa::abcd:1234]/home/jones/SIUSW/siu.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).

6.3 ActivateSW

This command closes any established traffic connections and restarts the unit with the latest downloaded software and optionally together with a downloaded bulk CM XML configuration file. 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.

6.3.1 Input Parameters

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



Table 93 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 XML configuration file will be activated. The requirement is that a bulk CM session exists in <i>ActivationInProgress</i> state and that it has been entered through the activate or restore command and with the parameter <i>delayed</i> set. If the requirement is not fulfilled, the 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 the command endSWSession has been received within the specified time (<i>autoRollbackTimer</i>) in minutes.

6.3.2 Output Parameters

Table 94 Output Parameters

Parameter Name	Description
status	Indicates either of the following: <i>OperationSucceeded</i> . <i>OperationFailed</i> - followed by the specified or unspecified reason.

6.3.3 Example 1

```
OSmon> activateSW s1  
OperationSucceeded
```

The *sessionIdSW* is "s1".

6.3.4 Example 2

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

The *sessionIdSW* is "s1".



The optional parameter *useNewConfiguration* is included to ensure that both the latest downloaded software and bulk CM XML configuration file should be activated.

6.3.5 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 the command **endSWSession** has been received before the *autoRollbackTimer*, with the value "10" (minutes), has expired.

6.4 EndSWSession

This command ends the specified software 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.

6.4.1 Input Parameters

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

Table 95 Input Parameters

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

6.4.2 Output Parameters

Table 96 Output Parameters

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

6.4.3 Example

```
OSmon> endSWSession s1
OperationSucceeded
```



The *sessionIdSW* is "s1".

6.5 GetSWSessionIds

This command returns the identity of an existing software session.

6.5.1 Input Parameters

None

6.5.2 Output Parameters

Table 97 Output Parameters

Parameter Name	Description
sessionIdList	This parameter shows a list of existing session Id(s). If no software session exists, this is indicated with --.
status	Indicates either of the following: OperationSucceeded. OperationFailed - followed by the specified or unspecified reason.

6.5.3 Example

```
OSmon> getSWSessionIds  
s1  
OperationSucceeded
```

The output parameter *sessionIdList* is "s1".

6.6 GetSWSessionStatus

This command returns the status of the specified software session.

6.6.1 Input Parameters

Syntax: `getSWSessionStatus <sessionIdSW>`



Table 98 Input Parameters

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

6.6.2 Output Parameters

Table 99 Output Parameters

Parameter Name	Description
sessionState	The state of the specified software session. Possible values: "Idle", "DownloadInProgress", "DownloadFailed", "DownloadCompleted", "ActivationInProgress", "ActivationFailed", "ActivationCompleted".
status	Indicates either of the following: OperationSucceeded. OperationFailed - followed by the specified or unspecified reason.

6.6.3 Example

```
OSmon> getSWSessionStatus s1
DownloadInProgress
OperationSucceeded
```

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

6.7 AbortSWSessionOperation

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

6.7.1 Input Parameters

Syntax: `abortSWSessionOperation <sessionIdSW>`



Table 100 Input Parameters

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

6.7.2 Output Parameters

Table 101 Output Parameters

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

6.7.3 Example

```
OSmon> abortSWSessionOperation s1  
OperationSucceeded
```

The *sessionIdSW* is "s1".

7 Alarm Subscriptions

This chapter lists commands related to subscriptions to alarms and notifications. Alarms are sent as SNMPv2c traps to SNMP manager(s) in the OSS. Subscriptions are stored persistently and survive restarts. The list of current alarms (displayed using the `getalarmlist` command) is removed at a restart of the unit, regardless of the restart cause, and any persistent faults will be regarded as new alarms after the restart.

7.1 Subscribe

This command defines the IP address of an SNMP manager and the time interval between heartbeat notifications. When the command is executed, alarms and notifications are sent to the specified SNMP manager. Up to 10 active subscriptions are supported, each assigned a unique identity by the



system, but there can only be one defined subscription for each IP address (*managerReference*).

7.1.1 Input Parameters

Syntax: `subscribe <managerReference> <timeTick>`

Table 102 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.

7.1.2 Output Parameters

Table 103 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: OperationSucceeded. OperationFailed - followed by the specified or unspecified reason.

7.1.3 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".



7.2 Unsubscribe

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

7.2.1 Input Parameters

Syntax: `unsubscribe <subscriptionId>`

Table 104 Input Parameters

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

7.2.2 Output Parameters

Table 105 Output Parameters

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

7.2.3 Example

```
OSmon> unsubscribe 1  
OperationSucceeded
```

The *subscriptionId* is "1".

7.3 GetSubscriptionIds

This command returns the identity of existing subscriptions.

7.3.1 Input Parameters

None.



7.3.2 Output Parameters

Table 106 Output Parameters

Parameter Name	Description
subscriptionIdSet	This parameter shows a list of existing subscription Id(s). If no subscriptions exist, this is indicated with --.
status	Indicates either of the following: OperationSucceeded. OperationFailed - followed by the specified or unspecified reason.

7.3.3 Example

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

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

7.4 GetSubscriptionStatus

This command returns the characteristics of the specified subscription.

7.4.1 Input Parameters

Syntax: `getSubscriptionStatus <subscriptionId>`

Table 107 Input Parameters

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



7.4.2 Output Parameters

Table 108 Output Parameters

Parameter Name	Description
subscriptionState	The state of the specified subscription. Possible values: notSuspended (active) or suspended.
timeTick	The time interval (in minutes) between heartbeat notifications. If the value is "0", heartbeat is turned off.
managerReference	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: OperationSucceeded. OperationFailed - followed by the specified or unspecified reason.

7.4.3 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".

7.5 SuspendSubscription

This command suspends the sending of alarms and notifications for the specified subscription.

7.5.1 Input Parameters

Syntax: `suspendSubscription <subscriptionId>`

Table 109 Input Parameters

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



7.5.2 Output Parameters

Table 110 Output Parameters

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

7.5.3 Example

```
OSmon> suspendSubscription 2
OperationSucceeded
```

The *subscriptionId* is "2".

7.6 ResumeSubscription

This command resumes a defined, but suspended, subscription.

7.6.1 Input Parameters

Syntax: `resumeSubscription <subscriptionId>`

Table 111 Input Parameters

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

7.6.2 Output Parameters

Table 112 Output Parameters

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



7.6.3 Example

```
OSmon> resumeSubscription 2
OperationSucceeded
```

The *subscriptionId* is "2".

8 Performance Management

This section lists basic Performance Management (PM) commands used to subscribe to PM data, that is all counters in all defined MO classes. The collected PM data is periodically saved and uploaded to a file repository.

For information about managing the PM file collection, see Reference [3].

8.1 SuspendPMMeasurements

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

8.1.1 Input Parameters

Syntax: `suspendPmMeasurements <PMMeasurementInstance>`

Table 113 Input Parameters

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

8.1.2 Output Parameters

Table 114 Output Parameters

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



8.1.3 Example

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

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

8.2 ResumePMMeasurements

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

8.2.1 Input Parameters

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

Table 115 Input Parameters

Parameter Name	Description
PMMeasurementInstance	Instance of PM measurement MO. The value is always "STN=0,MeasurementDefinition=0".
useContainerFile (optional)	If set, 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	<p>An SFTP URI, including authentication information, that defines where to upload the PM data. The addressing of the file destination must follow the syntax:</p> <pre>"sftp://username:password@<remotehost>/<filepath>"</pre> <p>where <remotehost> specifies either the IPv4 address ("a.b.c.d") or the IPv6 address in square brackets ("[a:b::c]") of the file destination host, and <filepath> specifies the directory structure and filename.</p> <p>Note: The password must not contain a space or any of the following characters: @, %, ; or ". Note: In addition to the <filepath>, a subdirectory named "PM" must exist at the file store destination.</p>



8.2.2 Output Parameters

Table 116 Output Parameters

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

8.2.3 Example

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

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

9 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 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*
- [6] *SIU 02 Installation and Basic Configuration*