



Bee-CUSP Convergent Signaling Platform operating guide



Notice

The information contained in this document is subject to change without notice.

Buzzinbees SAS makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Buzzinbees SAS shall not be liable for any errors contained herein, or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Copyright © 2009 Buzzinbees SAS[®], All Rights Reserved. Buzzinbees SAS liability, trademark and document use rules apply.

Printing History

First edition

September 2010

Trademarks

All product names, which may be mentioned in this document, are the property of the respective trademark or servicemark holders and are hereby acknowledged.

Table of contents

Preface.....	viii
1.2 Document structure	ix
Associated and reference documents.....	ix
Terms and acronyms	ix
Conventions	ix
1. Managing the Bee-CUSP CSP platform	10
1.1 Platform consistency	10
1.1.1 FEP processes.....	10
1.1.2 BEP processes	11
1.2 Management tools.....	11
1.3 Management objectives.....	12
1.4 Basic management tasks	12
1.5 Starting the platform.....	13
1.5.1 Starting the FEP,BEP and DIR.....	13
1.5.2 Starting the application copy processes.....	13
1.5.3 Verifying process connections.....	13
1.6 Stopping the platform	14
1.6.1 Partial stop and restart	14
1.6.2 Stopping a FEP	15
1.6.3 Stopping a BEP subsystem and its applications	15
1.6.4 Restarting	16
1.7 Checking process states.....	16
1.7.1 Managing the proxy (Bee-CUSP CSP Front End)	16
1.7.2 Logging syslog messages	17
1.8 Restoring a configuration	17
1.8.1 Reconfiguring individual FEPs.....	17
1.8.2 AutoReconfiguration actions on the FEP	17
1.8.3 Restoring the configuration of the BEP	18
1.8.4 AutoReconfiguration actions on the BEP	19
1.8.5 Restoring event and SCCP message reporting	20
1.8.6 What cannot be restored	20
1.8.7 Using proprietary management applications	20
1.8.8 Reconfiguring without AutoReconfiguration	21
1.9 Changing the platform transport protocol	21
1.10 Changing the AutoReconfiguration path	21
1.11 Changing the default GTT rules path.....	21
1.12 Setting the Director timeout.....	22
1.13 Removing an application	22
1.14 Updating device firmware	22
1.15 Controlling E1/T1 loopback.....	22
1.15.1 FRAMER mode.....	22
1.15.2 REMOTE mode.....	23
1.15.3 Implementation	23
1.15.4 Related S7MP commands.....	23
1.16 Notes on Bee-CUSP CSP behavior	23
1.16.1 Forced uninhibit compliance	24
1.16.2 Adjacent restart procedure and combined linkset.....	24
1.16.3 TFC handling by Bee-CUSP CSP from non-adjacent STP	24
1.16.4 Adjacent restart by linkset.....	25
1.16.5 Framing.....	26
2. Monitoring the platform.....	27
2.1 Status indication facility.....	27
2.1.1 Changing settings.....	28
2.1.2 Process status indications	28

2.1.3	Monitoring the IPsec	32
2.1.4	Inactive AS management	33
2.1.5	Traffic distribution and diversion in case of failure	33
2.1.6	PC-State and N-PC-State	36
2.1.7	N-COORD primitives	37
2.1.8	Bee-CUSP CSP Front End congestion management	37
2.1.9	Multiple point codes	41
2.2	SNMP agent	41
2.3	Black box facility	42
2.3.1	Reconfiguration	42
2.3.2	Platform startup to enable black box facility	42
2.3.3	Disabling automatic process restart	44
2.3.4	Disabling automatic process startup at reboot time	45
2.4	Incoming user traffic interceptor	45
2.4.1	Using the incoming user traffic interceptor	45
2.4.2	Activating the incoming user traffic interceptor	46
2.4.3	Filtering traffic	48
2.4.4	Filtering traffic mechanism	48
2.5	Recommendations when developing an MSU collector application	50
2.6	The logging and tracing framework	50
2.7	Failfast	51
3.	The Bee-CUSP CSP Front End SNMP agent MIB	52
3.1	Overview	52
3.1.1	SNMP overall architecture	54
3.1.2	Platform functionality	55
3.1.3	Implementation	55
3.1.4	The MIB	56
4.	Using the SNMP agent	69
4.3	Configuring the SNMP master agent	69
4.3.1	Configuring an automatic restart	69
4.3.1	Configuring the snmpd.conf file	69
4.3.2	Configuring the set of traps reported	71
4.4	Starting the SNMP agent	76
4.5	Starting the snmptrapd daemon	76
4.6	Monitoring a platform through the SNMP agent	77
4.6.3	Querying the SNMP agent through snmpget_requests	77
4.6.4	Interpreting the set of traps that are reported	78
4.7	Updating platforms monitored through the SNMP agent	79
4.8	Stopping the SNMP agent	79
5.	Using the S7 management program (S7MP)	80
5.9	Introduction	80
5.9.1	Defining the operating environment	80
5.10	Starting S7MP	81
5.10.1	Starting S7MP without an association ID	81
5.10.2	Starting S7MP for preserving context	81
5.11	Status indication display option	82
5.12	Restrictions	83
5.13	Command syntax	83
5.13.1	Entering commands	84
5.13.2	Command results	85
5.14	Exiting S7MP	86
5.15	S7MP help system	86
6.	S7MP command descriptions	87
6.1	ACTIVATE	87
6.2	ADD	88
6.3	ASSOCIATE	89
6.4	CLOSE_DIALOGS	92

6.5 CONNECT	93
6.6 COPY_ACTIVE	94
6.7 CREATE	94
6.8 DEACTIVATE	96
6.9 DELETE	97
6.10 DISABLE	98
6.11 DISCONNECT	99
6.12 DISEVT	100
6.13 DISMSG	101
6.14 DO	102
6.15 DUMP	104
6.16 ENABLE	107
6.17 ENAEVT	108
6.18 ENAMSG	110
6.19 END_ASSOCIATE	111
6.20 EXIT	112
6.21 GTT_RULE_CREATE	112
6.22 GTT_RULE_DELETE	115
6.23 GTT_RULE_MODIFY	116
6.24 GTT_RULE_SHALL	118
6.25 GTT_RULE_SHOW	119
6.26 HELP	120
6.27 HISTORY	121
6.28 INHIBIT	121
6.29 INIT_LINK	122
6.30 LINKTEST	123
6.31 LOAD	124
6.32 LOAD_STAND_BY	125
6.33 MODIFY	125
6.34 RECONFIGURE	126
6.35 REMOVE	128
6.36 REPEAT	130
6.37 RK_RULE_SHALL	131
6.38 RK_RULE_SHOW	132
6.39 SAVE_CONFIG	135
6.40 SENDMSG	137
6.41 SET	138
6.42 SHNEXT	142
6.43 SHNRESET	146
6.44 SHOW	147
6.45 SHRESET	149
6.46 SPAWN	155
6.47 SWITCH_TABLE	156
6.48 TEST_GTT	156
6.49 TRACE	157
6.50 UNINHIBIT	158
6.51 UNTRACE	159
6.52 WAIT	159

List of figures

7. Management architecture.....	11
8. Level status indication transitions	30
9. Front-end incoming traffic	39
10. Front-end outgoing traffic	40
11. The incoming user traffic interceptor	46
12. Generic information in the MIB	53
13. Agent-specific information in the MIB.....	0
14. SNMP overall architecture.....	53
15. Platform containment tree.....	58
16. fep containment tree.....	59
17. mtp2 containment tree.....	59
18. m2pa containment tree.....	0
19. saal containment tree.....	61
20. mtp3 containment tree.....	61
21. sccp containment tree.....	62
22. sua containment tree	63
23. nifsua containment tree.....	63
24. m3ep containment tree	64
25. tcap containment tree.....	65
26. bep containment tree.....	66
27. trapParam containment tree	66
28. agentInit containment tree	67
29. agent containment tree.....	68

List of tables

30.	
31. Traffic STATESHARE and LOADSHARING characteristic combinations.....	36
32. Traffic distribution	0
33. Incoming user traffic interceptor environment variables	47
34. Initial hierarchy of the MIB.....	55
35. Managed objects / entity mapping.....	57
36. S7MP command syntax	83
37. One-letter severity codes	85
38.	

Preface

This guide provides details on the basic operational functions of the Bee-CUSP Convergent Signaling Platform platform, as well as how to use the SNMP agent and the SS7 management program (S7MP) to run it.

Document structure

- **Bee-CUSP CSP functionality** covers basic platform operation and monitoring.
- **Using the SNMP agent** describes how to configure, start and stop the SNMP, and how to use it to monitor Bee-CUSP CSP platform.
- **Using S7MP** provides information on how to use the SS7 management program to configure, control and monitor the Bee-CUSP CSP platform.

Associated and reference documents

The Bee-CUSP CSP documentation set includes the following documents:

- *Bee-CUSP CSP release notes*
- *Bee-CUSP CSP product description*
- *Bee-CUSP CSP installation and configuration guide*
- *Bee-CUSP CSP operating guide*
- *Bee-CUSP CSP entity reference guide*
- *Bee-CUSP CSP application developer's guide*
- *Bee-CUSP CSP glossary*

The *Release notes* accompany the software kit and contain important, last-minute information that may not be documented elsewhere. You should read the *Release notes* before anything else in the documentation set.

Terms and acronyms

The *Bee-CUSP CSP Glossary* defines all the terms and acronyms used in this guide.

Conventions

The following conventions are used in this guide:

Code text	Is used for code examples, file names and directory paths, command names, routine names, and syntax.
<i>italicized text</i>	Is used for document titles, parameters, emphasized text, and replaceable text.
bold text	Indicates the first instance of terms being defined in text. In examples, it indicates user input.
<angle brackets>	Indicate generic variable names that must be substituted by real values or strings.
UPPERCASE	The Linux operating system differentiates between UPPERCASE and lowercase characters. Examples, syntax descriptions, routine definitions, and literal strings that appear in text must be entered exactly as shown.
lowercase	

Chapter 1. Managing the Bee-CUSP CSP platform

This chapter gives general advice on managing a Bee-CUSP CSP platform and describes a number of typical management tasks.

Management of a Bee-CUSP CSP platform can involve two areas of activity:

- Software development, including the construction of management application programs. The reference document *Bee-CUSP CSP application developer's guide* provides all the information needed for this activity.
- Hands-on management, through the application of S7MP commands directly to the platform to perform such activities as:
 - Starting or stopping processes or machines on the platform
 - Adding or removing processes or machines
 - Changing the characteristics of data links or link configurations to take into account changes in platform traffic
 - Applying new GTT routing rules
 - Changing default pathnames, including those to routing tables

Note



For information on platform monitoring, see [Monitoring the platform](#).

1.1 Platform consistency

Processing in the distributed architecture of a Bee-CUSP CSP platform relies on the fact that configuration information is contained in each of the Bee-CUSP CSP processes. In order for the platform to function correctly there must be consistency between the configuration databases maintained by the processes.

Platform management practices and activity, including the development of management applications, should always take into account this need to maintain consistency.

1.1.1 FEP processes

Since the FEP feature is distributed among several machines, the FEP processes on each machine must be configured in exactly the same way, that is:

- The same entities must be created on each FEP process;
- The status of the entities on each FEP must be consistent with each other.

It is possible to add a new FEP and configure it alone by downloading the configuration data from another FEP in the platform that is in the CONNECTED state. This mode of reconfiguration is referred to as dynamic reconfiguration.

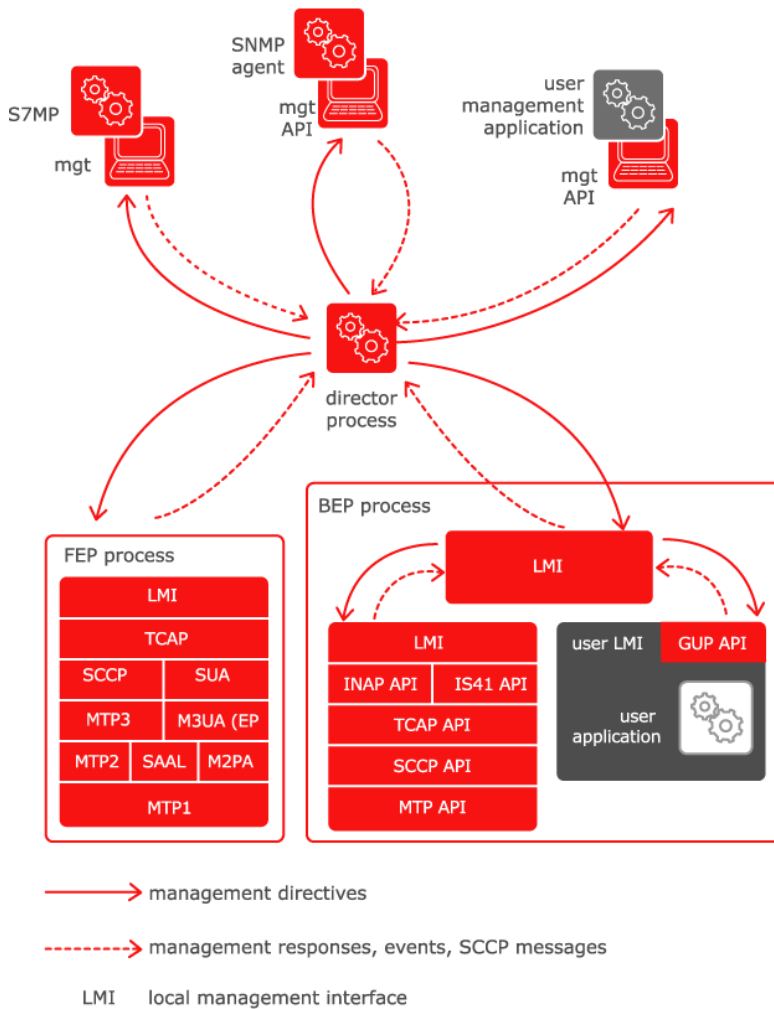
A more secure method to ensure consistency across the platform is to take advantage of the functionality offered by the *AutoReconfiguration* facility, see [Restoring a configuration](#). This will perform a reconfiguration of all FEPs on the platform, including a newly added one, from the same database, referred to as the static configuration database.

The requirement for this functionality is that the static configuration database be previously constructed by using the `SAVE_CONFIG` command in the S7MP utility. It is good practice to save the configuration in this way whenever changes are made.

1.1.2 BEP processes

The BEP feature is not distributed across several machines and each BEP is responsible for handling its own application processes only. If the BEP is stopped while outstanding application processes are still running, all the management entities related to these processes must be reconfigured. The BEP can be reconfigured using the *AutoReconfiguration* facility and the `RECONFIGURE S7MP` command, provided the configuration has previously been saved using the `SAVE_CONFIG S7MP` command.

Fig 1: Management architecture



1.2 Management tools

The tools used to manage a Bee-CUSP CSP platform are:

- The S7MP management application supplied, of which the commands and qualifiers are described in *Using the SS7 management program (S7MP)*
- Custom-built management applications, constructed using the guidelines described in the *Application developer's guide*.

1.3 Management objectives

During the working life of a Bee-CUSP CSP platform, the objectives of management generally will be to maintain the efficiency and smooth running of the platform. Over time, management may also involve the modification and adaptation of the platform to cope with change, such as:

- Changes in the structure of the external network;
- Changes in traffic patterns and volumes;
- Changes in management strategy.

Maintaining the efficiency of the platform will involve monitoring its performance, mostly through the regular, day-to-day monitoring of event activity for certain critical events.

Evolution of the platform to follow that of the communication environment may involve:

- Adding machines to, or removing machines from, the platform;
- Adding new links and communication devices to handle increases in traffic;
- Creating additional FEP and BEP processes to provide increased processing power;
- Changing the associations between links, linksets and drivers, and the FEP processes that manage them;
- Changing the way traffic is managed (loadsharing, for example);
- Changing protocol timers to tune the platform;
- Changing the characteristics of entity instances (deleting old ones and creating new ones);
- Changing routing rules and tables;
- Adding or deleting application copies.

All these activities are possible using the Bee-CUSP CSP tools that are provided in order to maintain an efficient and dynamic platform.

1.4 Basic management tasks

The following management tasks are described below:

- Starting the platform
- Stopping the platform
- Partial stop and restart
- Checking process states
- Restoring a configuration
- Changing the platform transport protocol
- Changing the AutoReconfiguration path
- Changing the default GTT rules path
- Setting the Director timeout
- Removing an application
- Updating device firmware
- Controlling E1/T1 loopback

Note



For information on platform monitoring, see [Monitoring the platform](#).

1.5 Starting the platform

When the configuration procedure is complete, the platform can be started using the `ss7startup` command to start the FEP, BEP and DIR subsystems and the application processes. The subsystems and processes must be started separately on each machine in the platform.

1.5.1 Starting the FEP, BEP and DIR

The general syntax of the command is:

```
ss7startup [<machine>]<platform_id> fep|bep|dir|<application_name>
```

The `ss7startup` command can be issued on any machine (the local machine) to operate on any other machine in the platform (the remote machine). When the command is entered without the machine name, the local machine is assumed by default.

1.5.1.1 Local operation

To start two FEPs and a DIR on platform a, the command entry would be:

```
ss7startup a fep1 fep2 dir
```

1.5.1.2 Remote operation

To start two FEPs on the remote machine `genepi`, on platform b, the command would be:

```
ss7startup genepi b fep1 fep2
```

1.5.2 Starting the application copy processes

Use `ss7startup`, as for the FEP, BEP or DIR, or execute the command file generated by the `ss7configure` utility.

```
/use/var/ss7/<platform_id>/bin/ss7_start_app_<application_name>_<platform_id>
```

1.5.3 Verifying process connections

Connections between Bee-CUSP CSP processes are established at process startup time and each process that starts tries to connect to other processes that are part of its configuration. A running Bee-CUSP CSP process that receives a connection request from another process responds to the request. Connections are made as follows:

- FEPs connect to all other FEPs, all BEPs, all DIRs and all applications
- BEPs connect to all FEPs, all DIRs and to local applications
- DIRs connect to all FEPs and all BEPs
- Applications connect to all FEPs and the local BEP

When processes are configured and running, check that they can reach the other processes. For each FEP and BEP process use `S7MP` to validate that:

- The `SUBS_LIST` attribute contains the type and name of each Bee-CUSP CSP process
- The `ASSO_LIST` attribute contains the name of each application copy

1.6 Stopping the platform

The platform can be stopped by using the `ss7shutdown` utility to stop the all subsystems on each machine in the platform.

```
ss7shutdown [<machine>]<platform_id> ss7_all
```

The `ss7shutdown` command can be issued on any machine (the local machine) to operate on any other machine in the platform (the remote machine). When the command is entered without the machine name, the local machine is assumed by default.

The SS7 network views this as a `clean` shutdown of the network signaling point.

1.6.1 Partial stop and restart

A partial stop of the platform followed by a restart may be required for several reasons, for example:

- Following a subsystem failure: In this case an event report is sent to the management application. When the problem is resolved, the subsystem is treated as a new one and is reintroduced and configured in the normal way.
- When adding a new Bee-CUSP CSP process, application copy or management entity: This allows a platform to be dynamically extended without shutting down the whole platform.
- By a management application or by a manual power-off, for whatever reason.

The tools used are the S7MP management program or the `ss7shutdown` command.

Suggestions for the FEP

- In the case of a FEP crash, restart the FEP and reconfigure it using the *AutoReconfiguration* feature
- In the case of FEP inconsistency, or if communication services are seen as congested:
 - If at least one FEP is running and connected, deactivate the links, restart the FEP and reconfigure it using *AutoReconfiguration*
 - If no FEP is running and connected, stop the service, deactivate the links, restart the FEP and reconfigure it
- In the case of inconsistency on several FEPs, restart the whole platform.

Suggestions for the BEP

- In the case of a BEP crash, restart the BEP and reconfigure it using the *AutoReconfiguration* feature
- In the case of BEP inconsistency (MTP3 database inconsistency), restart the BEP and reconfigure it using the *AutoReconfiguration* feature
- In the event of machine problems:
 - Stop any application copies
 - Stop the BEP
 - Solve the machine problem
 - Restart the BEP and its application copies, then reconfigure it using *AutoReconfiguration*

Suggestions for application copies

- Following an application crash, restart the application and reconfigure it using the *AutoReconfiguration* feature.
- Other application problems:
 - Use the `CLOSE_DIALOG` command to prevent the opening of new transactions
 - Restart the application and reconfigure it using the *AutoReconfiguration* feature.

Suggestions for the whole platform

- In the event of serious inconsistency problems on the platform:
 - Stop all application copies (`CLOSE_DIALOG`)
 - Deactivate all links
 - Stop all the Bee-CUSP CSP processes
 - Restart the processes and reconfigure the whole platform using the *AutoReconfiguration* feature.

1.6.2 Stopping a FEP

An individual FEP can be stopped with the `ss7shutdown` command:

```
>ss7shutdown [<machine>]<platform_id> fep
```

The machine name is needed if you are stopping a FEP on a remote machine. It is not needed if the FEP is on the local machine.

If the FEP is connected, it is advisable to first use S7MP to disconnect the FEP subsystem from the platform so that it is no longer operational:

```
S7MP>DISCONNECT fep <machine_name>
```

Remote SS7 nodes see the disconnection of the FEP as a number of *out_of_service* data links .

1.6.3 Stopping a BEP subsystem and its applications

If the BEP subsystem is connected, the S7MP commands required to stop BEP subsystem `padova` are as follows:

1. Close dialogs for all TCAP_BEP entities (application copies), for example:

```
CLOSE_DIALOGS BEP PADOVA TCAP_BEP 24
```



Note

The `DISABLE` command should not be issued until all dialogs in progress are complete, i.e. when all TCAP_BEP states are available.

2. Disable all application type entities (TCAP_BEP, SCCP_BEP, and MTP_BEP), for example:

```
DISABLE BEP PADOVA TCAP_BEP 36
```

3. Disconnect the BEP subsystem:

```
DISCONNECT BEP PADOVA
```

When a BEP subsystem is stopped, it is taken out of the Bee-CUSP CSP local configuration, but the signaling point remains operational. Stopping a BEP subsystem also affects the FEP subsystems, as the FEPs contain a list of application copies running on each BEP.

It is possible to subsequently reinsert the BEP subsystem.. If the BEP is not connected, use the `ss7shutdown` command to stop the BEP subsystem, for example:

```
ss7shutdown padova a bep
```

1.6.4 Restarting

Use the `ss7startup` utility, as described in [Starting the platform](#) to start the processes.

Use the *AutoReconfiguration* utility to restore or apply the management configuration, with the `RECONFIGURE S7MP` command. See also [Restoring a configuration](#).

1.7 Checking process states

Use the `ps` command to check the state of the processes that are started. The following example checks the state of the FEP, BEP, and DIR processes started on platform ID `a`.

- On a front-end node:

```
% ps -e|grep ss7_|grep -v grep
1137 ttyp2 I N 0:01.16 /usr/var/ss7/platform_y/bin/
ss7_dir_ansi_a.exe
3683 ttyp2 S N 0:02.23 /usr/var/ss7/platform_y/bin/
ss7_fep_ansi_a.exe
%
```

- On a back-end node:

```
% ps -e|grep ss7_|grep -v grep
1137 ttyp2 I N 0:01.16 /usr/var/ss7/platform_y/bin/
ss7_dir_ansi_a.exe
3786 ttyp2 I N 0:00.88 /usr/var/ss7/platform_y/bin/
ss7_bep_ansi_a.exe
%
```

Processes must be in either `S` `N` or `I` `N` state.

1.7.1 Managing the proxy (Bee-CUSP CSP Front End)

A single command, `ss7suaproxy`, covers all management related to the proxy. The `ss7suaproxy` command may take the following values:

- `ss7suaproxy startup`, used to launch the proxy once it has been provisioned
- `ss7suaproxy shutdown`, used to stop the proxy process
- `ss7suaproxy restart`, used to stop and restart the proxy process
- `ss7suaproxy reload`, used to shut down the SUA proxy and load the edited configuration file into the provisioning daemon, `gmfprovd`. It then restarts the SUA proxy.

Note



Before launching `ss7suaproxy reload`, ensure the provisioning daemon is up and running. If not, log

on as root and run:

```
# /sbin/service gmfprovd start
```

The `ss7suaproxy` command is also used for monitoring purposes:

- `ss7suaproxy showprov`, used to view the static configuration

- `ss7suaproxy showconf`, used to view the dynamic configuration and to monitor proxy operations (such as counters, ASP status).

1.7.2 Logging syslog messages

Bee-CUSP CSP Front End logs messages using the `syslog` utility. The `syslog` utility is automatically configured during installation and there are two types of message file:

- Operator messages `/var/log/ss7.log`
To view the messages online, enter:
`tail -f 'ss7syslogfile'`
- Process output
`/usr/var/ss7/<plaform_id>/log/<process>.err`

1.8 Restoring a configuration

The simplest, quickest and most secure method of restoring a configuration is to open a management session and issue the command

```
S7MP> RECONFIGURE
```

with no arguments. This invokes the *AutoReconfiguration* utility.

Provided the platform configuration has been previously saved using the `SAVE_CONFIG S7MP` command, this action restores the complete FEP and BEP entity tree structures from the static platform database of the SUBSYSTEM entity and guarantees the FEP consistency of the platform.

For general information on the *AutoReconfiguration* utility, refer to the *Bee-CUSP CSP product description*.

1.8.1 Reconfiguring individual FEPs

If you need to configure just one FEP subsystem, typically when you add a new FEP to an existing platform, you can do so by using the `RECONFIGURE S7M` Pcommand with the FEP as argument, and provided there is at least one other connected FEP in the platform.

```
S7MP>RECONFIGURE FEP ARUM
```

To reconfigure all FEPs that need to be reconfigured, use the asterisk (*) wildcard:

```
S7MP>RECONFIGURE FEP *
```

However, the recommended method is still to execute the `RECONFIGURE` command without arguments, as described in **Restoring a configuration**. This will ensure consistency between the FEPs in the platform.

1.8.2 AutoReconfiguration actions on the FEP

The `RECONFIGURE` command restores the complete configuration of the targeted `FEP_SUBSYSTEM` entity or entities and the child entities. The reconfiguration consists of:

- Creating the `FEP_SUBSYSTEM` and its child entities with their attribute values set as follows:
 - Characteristic attributes are set to the values previously saved
 - Counter attributes are set to 0, except for the `SS7_K_*_CREATIME` counters, which are set to the time when reconfiguration is performed
 - Event attributes are set to the `DISABLED` state
 - Status attributes are set to the `DISABLED` state

- Putting the data links into the state they held before *AutoReconfiguration* (INIT_LINK on the DATA_LINK)
- Activating or deactivating the linksets (ACTIVATE or DEACTIVATE on the LINKSET entity)
- Restoring SS7 links to the state they held before *AutoReconfiguration* (ACTIVATE or DEACTIVATE on the LINKSET entity; ACTIVATE, DEACTIVATE, or INHIBIT on the LINK entity)
- Enabling SCCP message reporting to applications (ENAMSG on the SCCP entity)
- Enabling event reporting (ENAEVT on appropriate entities with qualified events)
- Putting the SSNs in service (SENDMSG on the APPLICATION entity).

The INIT_LINK command is executed on a DATA_LINK during *AutoReconfiguration* if the data link is enabled.

The RECONFIGURE command can be applied to a named, non-connected FEP_SUBSYSTEM or it can be applied to all non-connected FEP_SUBSYSTEMs using the asterisk (*) character to replace the name of the FEP_SUBSYSTEM entity.

If there is no connected FEP in the platform, the FEP_SUBSYSTEM denti is reconfigured using the FEP permanent configuration file generated by the last SAVE_CONFIG command.

If there is at least one connected FEP in the platform, the FEP_SUBSYSTEM denti is reconfigured using a temporary file generated from the configuration of a selected connected FEP_SUBSYSTEM.

If an error occurs, one of the following values is returned:

- SS7_ENTITYEXISTS: The named FEP has already been created, but not connected
- SS7_NOTDONE: The named FEP is already connected; reason - LEVEL ALREADY CONFIGURED
- SS7_REJFAILED: The named FEP cannot read the permanent configuration file; reason - NO RECONF /EMPTY FILE Or: The permanent configuration file is corrupted and the FEP process is stopped to ensure consistency between the FEPs; reason - BAD RECONF FILE
- Return code of the request that could not be restored: An error occurred. A structure is returned that describes where the error occurred and the corresponding FEP process is stopped to ensure consistency between the FEPs.

If there is no connected FEP in the platform, all the FEP_SUBSYSTEM entities are reconfigured using the FEP permanent configuration file generated by the last SAVE_CONFIG command.

If there is at least one connected FEP in the platform, all the unconnected FEP_SUBSYSTEM entities in the platform are reconfigured using a temporary file generated from the configuration of a selected, connected FEP_SUBSYSTEM.

In either case, several FEPs can be targeted by this command, so there could be several output lists to consolidate.

Error returns are the same as in the previous example.

1.8.3 Restoring the configuration of the BEP

The RECONFIGURE command restores the named BEP_SUBSYSTEM and/or its appropriate local child entities to the state they were in before the interruption of the service. For example, the BEP_SUBSYSTEM entity is restored as CONNECTED, the MTP_BEP entity is restored as ENABLED, and the TCAP_BEP entity is restored as SUSPENDED. This command also restores the events that were enabled before the interruption of the service.

The RECONFIGURE command can only apply to the BEP_SUBSYSTEM and its child entities that refer to application copy processes that have been started.

If the named BEP is CONNECTED, only the entities referring to the API levels used by the local application copies that need reconfiguring will be reconfigured.

Note



The RECONFIGURE command is performed even if the MTP3 database has not been initialized. The CREATE and ENABLE commands applied to a TCAP_BEP entity are processed even when the MTP3 database is not initialized. See the `ss7_tcap_enable_indic_XXX` routine in the *Application developer's guide*.

1.8.4 AutoReconfiguration actions on the BEP

The reconfiguration of the BEP_SUBSYSTEM entity consists of:

- Creating the BEP_SUBSYSTEM with its attribute values set as follows:
 - Characteristic attributes are set to the values previously saved
 - Counter attributes are set to 0, except for the SS7_K_BEPS_CREATIME counter, which is set to the time when reconfiguration is performed
 - Event attributes are set to the DISABLED state
 - Status attributes are set to the DISABLED state
- Enabling the events that were enabled on the BEP_SUBSYSTEM entity
- Enabling the BEP_SUBSYSTEM entity
- Connecting the BEP_SUBSYSTEM entity

1.8.4.1 Reconfiguration of API level entities

The reconfiguration of entities relating to the API levels used by application copies consists of:

- Creating the appropriate BEP_SUBSYSTEM child entities with their attribute values set as follows:
 - Characteristic attributes are set to the values previously saved
 - Counter attributes are set to 0, except for the SS7_K_*_CREATIME counter, which is set to the time when reconfiguration is performed
 - Event attributes are set to the DISABLED state
 - Status attributes are set to the DISABLED state
- Enabling the events that were enabled on the BEP_SUBSYSTEM child entities
- Enabling the BEP_SUBSYSTEM child entities that were enabled
- Suspending the TCAP_BEP entities that were SUSPENDED.

Restores the management configuration commands required for any stopped or failed part of the congr BEP function, including the service applications that it hosts.

If an error occurs, one of the following values is returned:

SS7_ENTITYEXISTS	The named BEP has already been created, but not connected
SS7_REJFAILED	The named BEP cannot read the permanent configuration file; reason - NO RECONF/EMPTY FILE
Return code of the request that could not be restored	An error occurred during reconfiguration of the named BEP_SUBSYSTEM; a structure is returned that describes where the error occurred and the corresponding BEP process is stopped; a message is logged in the syslog file
SS7_PARTIALLY_COMPLETED	An error occurred during reconfiguration of an application type entity or one of their local child entities; a message is logged in the syslog file and the processing of the RECONFIGURE command continues
SS7_REJFAILED	The permanent configuration file is corrupted and the BEP process is stopped; reason - BAD RECONF FILE

1.8.5 Restoring event and SCCP message reporting

To achieve a full reconfiguration, i.e. one that includes the restoration of event and SCCP message reporting, it is necessary to start S7MP using the association ID argument (see [Using the SS7 management program \(S7MP\)](#)). This argument is required to ensure that the capability of restoring these reports is switched on and that, in the event of a subsequent disassociation and re-association between the application and the DIR, the management application reassociates with the same DIR function under the same reporting conditions as before the association was interrupted.

If the association ID argument is omitted at S7MP startup, event and SCCP message reporting cannot be restored when the management application reassociates after the DIR process restarts, the management application exits, or a Ctrl/C key sequence in the management application.

1.8.6 What cannot be restored

The *AutoReconfiguration* utility cannot restore the following cases:

- The values of the TRUNK entity characteristics SS7_K_TRNK_CGRBONSET or SS7_K_TRNK_CGRBABAT if they have been modified and are not the same as the default values of these attributes
- The SYSFAIL event on the FEP_SUBSYSTEM and the BEP_SUBSYSTEM entities

To restore the modified values of SS7_K_TRNK_CGRBONSET or SS7_K_TRNK_CGRBABAT after *AutoReconfiguration*, execute the MODIFY command manually:

```
MODIFY FEP * MTP2 TRUNK * /CGRBONSET=640 /CGRBABAT=896
```

Note that the values of CGRBONSET and CGRBABAT must be less than the value of RBSIZE.

To restore SYSFAIL events, execute the following commands manually:

```
ENAEVT FEP * /SYSFAIL ENAEVT BEP <bep_name> /SYSFAIL
```

1.8.7 Using proprietary management applications

See the *Application developer's guide* for the equivalent mechanism for your own management application, which is implemented through the `ss7_mgt_associate` routine.

The SAVE_CONFIG and the RECONFIGURE commands targeting the FEP function, or given with no specific targeted entity, must be performed in the same instance of the management application.

If you use your own management application:

- Saving the configuration is implemented through the `SS7_MGT_K_ACT_SAVE_CONFIG` action.
- *AutoReconfiguration* is implemented through the `SS7_MGT_K_ACT_RECONFIGURE` action.

See the *Application developer's guide* for further details.

1.8.8 Reconfiguring without AutoReconfiguration

If for some reason you are unable to use *AutoReconfiguration* you will have to carry out a manual reconfiguration. As a starting point for the manual configuration of the platform, you can use the generic script file that was produced on one of the FEPs in the initial configuration stage. This will reduce the amount of input you have to provide.

1.9 Changing the platform transport protocol

If you need to change the transport protocol, you must delete the platform configuration, then recreate a platform configuration with the new transport protocol settings.

1.10 Changing the AutoReconfiguration path

The *AutoReconfiguration* function relies on the existence of a permanent configuration file, saved by means of the `SAVE_CONFIG S7MP` command. When the platform configuration details are first saved, the directory structure created for the platform at that time includes a directory to contain the permanent configuration file. The default path for the *AutoReconfiguration* feature is `/usr/var/ss7/platform_<id>/save_config/ss7_save_config.dat`

If your management policy requires that the permanent configuration file be stored in a different directory you may change this default path in the following manner:

1. Start `ss7configure` and navigate to the Main menu.
2. Select item **5** - Modify the platform characteristics
3. On the Platform characteristics setup menu, select item **3** - Set the AUR path location
4. Enter the full path
5. Return to the Main menu, apply the configuration and exit.

1.11 Changing the default GTT rules path

The path to the directory containing the GTT rule files supplied in the kit is set by default to `/usr/var/ss7/platform_g/gtt` and it is not necessary to configure this if the default directory is to be used. You may however change this default path to point to an alternative set of GTT rules.

You should note carefully that the GTT rule databases of the machines in your platform are not aligned automatically. Therefore, you are responsible for maintaining their consistency.

For general information on GTT routing and rules, see the *Installation and configuration guide*.

The GTT rules default pathname is changed using `ss7configure`, through option 5 on the Main Menu (Modify the platform characteristics).

1. Start `ss7configure` with the platform ID as an argument, which displays the Main menu
2. Select item **5**- Modify the platform characteristics
3. Select item **1** -Set the directory for GTT files
4. Enter the new path
5. Save the configuration and exit (Item 6 on the menu)

1.12 Setting the Director timeout

The environment variable `SS7_MGT_TIMEOUT` sets the management request timeout whose default value is 6000 (that is, 1 minute in units of 0.01s). This timeout governs the interval that the Director waits before sending a timeout response to a management application when a FEP or BEP process does not reply in time to a management request.

The value set takes effect when the Director process is started. To change the default timeout value to, for example, 30 seconds, enter one of the following commands:

- In shell:

```
SS7_MGT_TIMEOUT=3000; export SS7_MGT_TIMEOUT
```

- In C shell:

```
% setenv SS7_MGT_TIMEOUT 3000
```

Then start or restart the Director process using the `ss7startup` command.

1.13 Removing an application

It is possible to remove an application without stopping the platform, by putting the APPLICATION entity *out_of_service*, for example:

```
SENDMSG FEP * SCCP APPLICATION 6 /NSTATE_REQ /USER_STATUS=OUT_OF_SERVICE
```

1.14 Updating device firmware

For information about updating device firmware, see the *Bee-CUSP CSP Front End installation and configuration guide*, under the heading 'Upgrading DNB board firmware'.

1.15 Controlling E1/T1 ILoopback

There are two loopback modes for E1/T1 connectivity, FRAMER and REMOTE, both of which are implemented at transceiver level.

1.15.1 FRAMER mode

In FRAMER mode, loopback is applied to check certain parts of the internal circuitry of the local DNB device. This is a transient mode, that is automatically removed when the tests are complete.

1.15.2 REMOTE mode

In REMOTE mode, loopback is applied on the local DNB device to enable the transmit/receive circuitry and the link itself to be checked from a remote machine. This mode is permanent and remains set until specifically disabled.

1.15.3 Implementation

Loopback on E1/T1 devices is implemented by setting the attribute values of an entity, the TRUNK entity. This is done by using the S7MP routine MODIFY. The TRUNK attribute is SS7_K_TRNK_LOOPBACK, which can be assigned the following values:

- SS7_K_TRNK_LB_NONE: no loopback mode is set;
- SS7_K_TRNK_LB_FRAMER: framer loopback mode is set (transient command);
- SS7_K_TRNK_LB_REMOTE: remote loopback mode is set (non-transient command);

To check the loopback FRAMER mode, the trunk events must be enabled before setting the trunk in loopback FRAMER mode. The event generates, to raise the test result:

event ID: SS7_K_TRNK_LFRAMER, with values: SS7_K_TRNK_LFRAMER_OK and SS7_K_TRNK_LFRAMER_KO.



Caution

In a daisy-chained configuration, remote loopback is not supported on the diverted trunk.

1.15.4 Related S7MP commands:

```
S7MP_A> MODIFY FEP FEP_NAME MTP2 TRUNK 1 /LOOPBACK=REMOTE
S7MP_A> MODIFY FEP FEP_NAME MTP2 TRUNK 1 /LOOP=NONE
S7MP_A> SHOW FEP FEP_NAME MTP2 TRUNK 1 /LOOP
S7MP_A> ENAEVT FEP FEP_NAME MTP2 TRUNK 1 /LFRAMER
S7MP_A> MODIFY FEP FEP_NAME MTP2 TRUNK 1 /LOOPBACK=FRAMER
```

A LOOPBACK related S7MP MODIFY command is rejected if:

- SS7_K_INVALID_ATT_VAL: the requested loopback mode is different from NONE, REMOTE or FRAMER.
- SS7_K_CONSTRAINT_VIOL: the trunk is not in the SS7_K_TRNK_AVAILABLE state.
- SS7_K_CONTEXT_INVALID: loopback is requested while the trunk is already in loopback mode.
- SS7_K_ASSOCIATED_LINK_ACTIVATED: at least one MTP3 link related to the trunk is ACTIVATED.
- SS7_K_RELATED_DTLK_NOT_ENABLED: framer is requested but no related MTP2 data link is ENABLED.

A trunk-related S7MP DISABLE command is not allowed if the trunk is currently in LOOPBACK mode (SS7_K_LOOPBACK_IN_PROGRESS). If the trunk is currently in LOOPBACK mode, a trunk-related MODIFY S7MP command of any dynamically settable characteristic is not allowed (SS7_K_CONTEXT_INVALID).

1.16 Notes on Bee-CUSP CSP behavior

This section describes various aspects of platform management behavior that may be useful.

1.16.1 Forced uninhibit compliance

Bee-CUSP CSP is compliant with *ANSI 96* and *ITU-T Q.704 1996 MTP3* (section 10.3.2) with regard to the action taken when receiving an uninhibiting request.

Both *ANSI 96* and *ITU-T 1996 (white book) recommendations* state that if, for any reason, an uninhibit signaling link message is not received in response to the force uninhibit message, a timer T13 expires. If this is the first expiry of T13 for this uninhibition attempt on this link, the procedure is restarted, including inspection of the status of the inhibited link. If the link is marked failed or blocked, or timer T13 has expired for the second time during uninhibition of this link, management is informed and the uninhibition is abandoned.

Bee-CUSP CSP behavior is as follows:

- For both *ANSI 96* and *ITU-T 1996 (white book) recommendations*, when the timer T13 has expired for the second time, the attempt to request the uninhibition is abandoned.
- For other recommendations (*ANSI 88*, *ITU-T blue book* and *red book*), the Bee-CUSP CSP behavior regarding this point remains as it has been in previous versions, i.e. when the timer T13 has expired for the second time, Bee-CUSP CSP marks the link state as uninhibited.

1.16.2 Adjacent restart procedure and combined linkset

This feature is provided only for the ANSI protocol version of Bee-CUSP CSP and satisfies the requirement that it should be possible to use both the adjacent signaling point restart procedure capability and the combined linkset capability in parallel.

1.16.2.1 How this feature works in Bee-CUSP CSP

The combined linkset is comprised of member linksets (at least two member linksets). Each member linkset applies its own adjacent restart procedure (as defined in the *ANSI 96 recommendations*) and it is only on completion of this procedure that the routing table of the combined linkset is updated in order to take into account the newly available member linkset.

This means that a remote destination which is reachable by a combined route (i.e. using a combined linkset) is only accessible when at least one member linkset completes its own adjacent restart procedure.

1.16.3 TFC handling by Bee-CUSP CSP from non-adjacent STP

This section explains how Bee-CUSP CSP handles this aspect of congestion management.

1.16.3.1 Behavior in previous Bee-CUSP CSP versions

While traffic is flowing in both directions, a TFC with a priority of 3 and corresponding to a remote destination known by Bee-CUSP CSP is sent from an STP unknown by the Bee-CUSP CSP platform to Bee-CUSP CSP. On receipt of such a TFC, Bee-CUSP CSP stops sending messages with priority levels of 0, 1 and 2 to the affected remote destination contained in the TFC.

Bee-CUSP CSP sets the T16 timer and returns a route set congestion test (RCT) message with a priority of 2 related to the congested destination. When no TFC is sent to Bee-CUSP CSP, timer T16 expires and an RCT with a priority of one related to the congested destination is sent.

Priority one messages are now restored from Bee-CUSP CSP to the congested destination. Again no TFC is sent to Bee-CUSP CSP, timer T16 expires and messages with a priority of zero are sent from Bee-CUSP CSP to the congested destination. At any time, the congestion state should be displayed when querying the destination using the S7MP management tool.

1.16.3.2 Preferred operation

All TFC messages received by Bee-CUSP CSP should be issued from existing STPs even if those STPs are not defined as adjacent point codes at the Bee-CUSP CSP platform configuration level. Note that it is the users' responsibility to ensure the consistency of the network configuration to avoid end nodes sending TFC messages to the Bee-CUSP CSP stack.

The affected destination enclosed in the TFC message should be a remote destination declared at the Bee-CUSP CSP platform configuration level. If it is not, Bee-CUSP CSP will discard the TFC message, a management event will be raised and a counter will be incremented.

For the combined linkset capability, RCT messages will be sent on all the links which make up the combined route. In other words, the RCT message will have the SLS code rotated as regular traffic and will be load-shared across all available links.

1.16.3.3 Current TFC handling

A new logical name has been defined within Bee-CUSP CSP to implement the TFC handling feature. This logical name will allow Bee-CUSP CSP to handle the situation where a remote STP, unknown to it, sends a TFC related to a remote destination which Bee-CUSP CSP should know.

The logical name to accept TFC messages is:

`SS7_$HANDLE_TFC_FROM_REMOTE_STP.`

The value of this logical name can be TRUE or FALSE. TRUE means that DECss7 will accept TFC initiated by a remote STP. FALSE means that DECss7 will discard TFC initiated by a remote STP.

1.16.4 Adjacent restart by linkset

In previous versions of Bee-CUSP CSP, the management characteristic used to select the adjacent signaling point restart procedure to apply was defined at the MTP3 entity level (i.e. selectable at the time of creation of the MTP3 entity).

1.16.4.1 The current situation

Using the S7MP management tool, it is only possible to issue the following commands:

```
CREATE FEP * MTP3 /ADJ_SP_RESTART=procedure
```

where `procedure` is ANSI88 or ANSI96 for the ANSI stack ITU-T RED, WHITE, BLUE books for the ITU/CHINA stack.

In the current version of Bee-CUSP CSP this management characteristic is implemented at the LINKSET entity level, but for backward compatibility the ADJ_SP_RESTART characteristic must remain as it is at present. The value of this characteristic is the default value for the linkset characteristic. This LADJ_SP_RESTART characteristic could be modified at.

For example, it is possible to issue the following commands:

```
CREATE FEP * MTP3 /ADJ_SP_RESTART=procedure1
CREATE FEP * MTP3 LINKSET 1 /ADJPC=2 /RULEID=2
CREATE FEP * MTP3 LINKSET 2 /ADJPC=3 /RULEID=3 /LADJ_SP_RESTART=procedure2
```

where `procedure` is ANSI88 or ANSI96 for the ANSI stack ITU-T RED, WHITE, BLUE books for the ITU/CHINA stack.

Bee-CUSP Convergent Signaling Platform operating guide

In this case, LINKSET 1 operates in the procedure1 mode, because the LADJ_SP_RESTART characteristic is not specified when the linkset is created, and so the mode specified at the MTP3 entity creation is used.

LINKSET 2 operates in the procedure2 mode. This mode may be different from that of procedure1, but it should be a valid value corresponding to the Bee-CUSP CSP installed stack.

This option would not be possible when creating a combined linkset. For example, using the S7MP management tool, the following command would be rejected:

```
CREATE FEP * MTP3 LINKSET 3/COMBINED/RULEID=5/LADJ_SP_RESTART=procedure
```

This should work regardless of whether the linksets are part of a combined linkset or not. The adjacent restart procedure for the combined linkset is controlled by the underlying linksets.

1.16.5 Framing

1.16.5.1 The background

In previous versions of Bee-CUSP CSP, at T1 trunk creation, the line coding depended on the line speed:

for 56kbit/s, it was AMI; for 64kbit/s, it was B8ZS. The preference was that it should be possible to have AMI or B8ZS framing for 56kbit/s. AMI line coding was not possible for 64kbit/s.

AMI line coding forces the eighth bit of each timeslot to 1, to ensure one density (hence 56K). However it should be possible to force the eighth bit of each timeslot to 1 (56K like AMI) and still code a 0 with B8ZS.

1.16.5.2 The present situation

For the current version of Bee-CUSP CSP a new characteristic attribute on the TRUNK entity (settable only at creation) has been added: SS7_K_TRNK_T1LCODING, and this has three possible values:

- SS7_K_TRNK_LC_DEFAULT (default value = today F/W behavior)
- SS7_K_TRNK_LC_AMI (for AMI line coding)
- SS7_K_TRNK_LC_B8ZS (for B8ZS line coding)

Examples of S7MP commands handling the T1LCODING attribute:

```
S7MP_A> CREATE FEP FEP_NAME MTP2 TRUNK 2 /DEVNAME=PT37B /FEPNAME=VERLEN /T1C=64
_S7MP_A> /T1LCODING=B8ZS
S7MP_A> CREATE FEP FEP_NAME MTP2 TRUNK 2 /DEVNAME=PT37B /FEPNAME=VERLEN /T1C=56
_S7MP_A> /T1L=B8ZS
S7MP_A> CREATE FEP FEP_NAME MTP2 TRUNK 2 /DEVNAME=PT37B /FEPNAME=VERLEN /T1L=AMI
S7MP_A> SHOW FEP * MTP2 TRUNK 2 /T1C /T1L /TYP
```

1.16.5.3 Unchannelized

For Unchannelized firmware, T1LINESPEED and T1LINCODIND are not settable:

- Line speed is set to 1.536
- Line coding is set to B8ZS

Chapter 2. Monitoring the platform

Monitoring is an essential part of platform management and Bee-CUSP CSP offers both S7MP commands and a variety of facilities to do this. These include:

- The status indication facility;
- The SNMP agent;
- The black box facility;
- The incoming user traffic interceptor;
- The logging and tracing framework, which includes logging, tracing, and the common trace facility (CTF);
- Failfast.

2.1 Status indication facility

The status indication facility includes process status indications (PSIs) for the DIR, FEP, BEP, and service application processes, and level status indications (LSIs) for the API levels used by the service applications.

When you start S7MP the value of the `s7mp_proc_status_indic` environment variable determines whether the Status Indication facility is switched on for PSIs only or whether it is on or off for all Status Indications, including LSIs. If this environment variable is not defined, the value of the `PROC_STATUS_INDIC` qualifier of the `ASSOCIATE` command is read as ON, which means that only PSI reporting is activated.

The possible values of the `s7mp_proc_status_indic` environment variable and of the `PROC_STATUS_INDIC` qualifier of the `ASSOCIATE` command are:

- ON to enable reporting of PSIs only
- OFF to disable reporting of PSIs and LSIs
- ALL to enable reporting of PSIs and LSIs

There is no option to enable only LSIs to be reported.

Changing settings

To change the reporting setting for status indications you must end the current association between S7MP and the DIR (`END_ASSOCIATE` command). Then you can reassociate S7MP with the DIR using the `ASSOCIATE` command and the appropriate setting for status indication reporting.

`ASSOCIATE /PROC_STATUS_INDIC=ON|OFF|ALL`

If you give the `ASSOCIATE` command without the `PROC_STATUS_INDIC` qualifier, the reassociation will be performed taking the value of the `s7mp_proc_status_indic` environment variable to set the status indication reporting. If this environment variable is not defined, the value ON is taken by default. For more information on environment variables, see 'Environment variables' in the *Installation and configuration guide*. See also the Entity reference guide for more information.

When status indication reporting is switched on (PSIs only or PSIs and LSIs), the management application receives DIR status indications even if the association with the DIR fails, but does not receive indications from the targeted FEP processes until the association with the DIR is successful.

The DIR may be unreachable when the management application starts up, in which case a value of `SS7_NOSS7MGT` is returned. The management application has to wait for the DIR to become reachable before it can associate successfully and receive the targeted process and level status indication reports.

In *AutoReconfiguration* of the BEP function, LSIs are extremely useful for indicating which API level entities have been reconfigured. Status Indications (SIs) provide the management application with information on the

status of Bee-CUSP CSP processes, user-written service application processes, and the API levels used by the service applications.

When you start S7MP the value of the `s7mp_proc_status_indic` environment variable determines whether the Status Indication facility is switched on for PSIs only or whether it is on or off for all Status Indications, including LSIs. If this environment variable is not defined, the value of the `PROC_STATUS_INDIC` qualifier of the `ASSOCIATE` command is read as ON, which means that only PSI reporting is activated.

The possible values of the `s7mp_proc_status_indic` environment variable and of the `PROC_STATUS_INDIC` qualifier of the `ASSOCIATE` command are:

- ON to enable reporting of PSIs only;
- OFF to disable reporting of PSIs and LSIs;
- ALL to enable reporting of PSIs and LSIs.

There is no option to enable LSIs only to be reported.

2.1.1 Changing settings

To change the reporting setting for status indications you must end the current association between S7MP and the DIR (`END_ASSOCIATE` command). Then you can reassociate S7MP with the DIR using the `ASSOCIATE` command and the appropriate setting for status indication reporting.

```
ASSOCIATE /PROC_STATUS_INDIC=ON|OFF|ALL
```

If you give the `ASSOCIATE` command without the `PROC_STATUS_INDIC` qualifier, the re-association will be performed taking the value of the `s7mp_proc_status_indic` environment variable to set the status indication reporting. If this environment variable is not defined, the value ON is taken by default. See the *S7MP Reference Guide* for more information.

When Status Indication reporting is switched on (PSIs only or PSIs and LSIs), the management application receives DIR Status Indications even if the association with the DIR fails, but does not receive indications from the targeted FEP, BEP, or service application processes, until the association with the DIR is successful.

The DIR may be unreachable when the management application starts up, in which case a return value of `SS7_NOSS7MGT` is given. The management application has to wait for the DIR to become reachable before it can associate successfully and receive the targeted Process and Level Status Indication reports.

2.1.2 Process status indications

Process status indications (PSIs) provide information on the current state of the DIR, BEP, FEP and service application processes.

A reachable FEP or BEP process indicates that the corresponding entity is ready to receive management configuration commands (`CREATE`, `ENABLE`, and `CONNECT`). A reachable FEP process also indicates that all the child entities of the corresponding `FEP_SUBSYSTEM` entity are ready to receive management commands. A connected FEP or BEP process indicates that the MTP3 database has been loaded and initialized following the `CONNECT` command on the entity.

If the DIR process is running when S7MP starts and S7MP associates successfully, the management application receives a reachable DIR indication followed by the indications showing the status of other monitored processes and then the last process indication (LPI) of the DIR indicating that there are no more responses to come.

If S7MP receives an unreachable DIR indication after a previously successful association, all subsequent attempts to establish an association will fail until S7MP receives a reachable DIR indication.

Reachable process status indications (DIR, FEP, BEP)

S7MP receives a reachable DIR indication when:

- A DIR process starts (at least one association performed)
- The DIR is already running on the Bee-CUSP CSP platform and the association is successful.

S7MP receives a reachable FEP or BEP indication when:

- The FEP or BEP process starts after a successful association
- A process for an unconnected FEP or BEP subsystem is running on the Bee-CUSP CSP platform, S7MP starts and the association is successful.

Connected process status indications (FEP)

S7MP receives a connected FEP or BEP indication when:

- A FEP_SUBSYSTEM is connected by a management command and the MTP3 database is initialized and loaded
- A BEP_SUBSYSTEM is connected by a management command and there is at least one connected FEP in the platform, so that the MTP3 database is initialized and loaded;
- A process representing a connected FEP or BEP subsystem is already running on the platform, S7MP starts and the association is successful

Reachable process status indications (service applications)

S7MP receives a reachable user (service) application indication when:

- The application process starts (one management association has been completed)
- The application process is already running on the Bee-CUSP CSP platform and the association is successful
- The application process is already running on the Bee-CUSP CSP platform and the BEP process restarts after a failure.

Unreachable process status indications

S7MP receives an unreachable process status indication when:

- A Bee-CUSP CSP process fails
- A BEP or DIR process fails on which another Bee-CUSP CSP process depends. If a BEP process fails, the dependent service applications are no longer visible. Likewise, if a DIR process fails, the FEP and BEP are no longer visible.

Last process indications

When the DIR or BEP restarts and the association is successful, the last process indication (LPI) is sent to inform S7MP that no more PSIs will arrive from any of the processes. For example, if the DIR fails and then restarts and S7MP successfully reassociates with it, S7MP receives PSIs only for the FEPs and BEPs that are reachable when the DIR restarts.

Applying this logic to a sample platform configuration, if all the functions are reachable when S7MP is started, S7MP receives:

1. Reachable PSI for the DIR
2. Reachable PSI for FEP *denti*
3. Reachable PSI for FEP *celebre*
4. Reachable PSI for BEP *congre*
5. Last process indication from the DIR

If the DIR fails and then restarts, the following sequence might occur:

1. Unreachable PSI for the DIR
2. Reachable PSI for the DIR
3. S7MP reassociates with the DIR
4. Reachable PSI for the DIR
5. Reachable PSI for FEP *celebre*
6. Reachable PSI for BEP *congre*
7. Last process indication from the DIR

The absence of a reachable PSI for FEP *denti* up until the arrival of the LPI from the DIR makes it possible to deduce that FEP *denti* has failed during the period that the DIR was unavailable. This illustrates the real utility of the LPIs: they make it possible to determine which processes are not reachable for any reason.

The last process indication marks the end of the startup phase of the DIR and BEP processes or the end of a failure phase of a BEP process.

2.1.3 Level status indications

Level status indications (LSIs) refer to the entities representing the API levels used by service applications. Level status indications notify the management application when there is a change in the status of the API levels used by a service application.

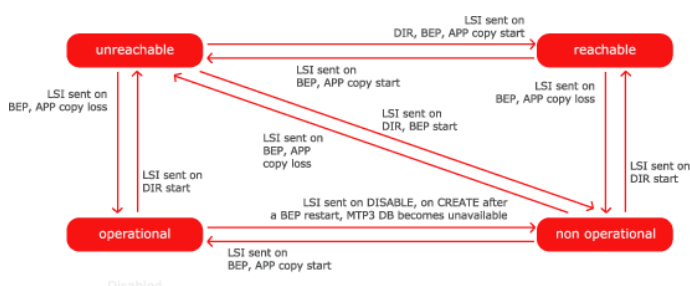
A service application can use one or more of the following API levels:

- MTP represented by the MTP_BEP entity;
- SCCP represented by the SCCP_BEP entity;
- TCAP represented by the TCAP_BEP entity.

S7MP can receive the level status indications raised by the API levels if the value of the `s7mp_proc_status_indic` environment variable or of the `PROC_STATUS_INDIC` qualifier of the `ASSOCIATE` command is set to `ALL` to enable reporting of PSIs and LSIs and S7MP performs a successful association with the DIR.

The entities representing the API levels used by a reachable service application may or may not be configured when S7MP, the BEP, or the DIR start or restart.

Fig 2: Level status indication transitions



Reachable

An API level has been defined for a service application and the corresponding entity is ready to receive management commands.

Unreachable

An API level has been defined for a service application but the corresponding entity is not ready to receive management commands.

Operational

An API level entity has been created and enabled and the MTP3 database is initialized and available.

Non-operational

An API level entity has been:

- Created but disabled;
- Created and enabled, but the MTP3 database is not available;
- Created but disabled and the MTP3 database is not available.

Note that the MTP3 database becomes unavailable when the last connected FEP or the BEP fails and the MTP3 database is no longer authoritative. The MTP3 database also becomes unavailable when the BEP with which it is associated is being initialized and the MTP3 database is no longer authoritative.

No LSI is raised if an event occurs that does not change the status of a given API level.

The figure below summarizes the transitions that trigger level status indications.

2.1.3.1 Reachable LSI

The management application receives a reachable LSI when:

- The status of a given API level changes from *unreachable* to *reachable*;
- The status of a given API level changes from *non-operational* to *reachable*. The following events trigger these transitions:
 - The first time a service application calls an `enable_indications` routine for an API level it uses;
 - A service application is running with a given API level, the management application performs a successful association, but the API level entity has not been created;
 - A service application is running with a given API level, the associated BEP restarts, but the API level entity has not been created;
 - A service application is running with a given API level, the associated DIR restarts, but the API level entity has not been created;
 - A service application restarts;
 - An API level entity is successfully deleted by a management command.

2.1.3.2 Unreachable LSI

The management application receives an unreachable LSI when:

- The status of an API level changes from *reachable* to *unreachable*;
- The status of an API level changes from *operational* to *unreachable*;
- The status of an API level changes from *non-operational* to *unreachable*.

The following events trigger these transitions:

- A service application fails;
- The associated BEP function fails.

2.1.3.3 Operational LSI

The management application receives an operational LSI when:

- The status of an API level changes from *non-operational* to *operational*;
- The status of an API level changes from *unreachable* to *operational*.

The following events trigger these transitions:

- An API level entity is successfully enabled by a management command and the MTP3 database is available;
- A service application is running with an API level for which the corresponding entity has been created and enabled, the MTP3 database is available and the management application performs a successful association;

- A service application is running with an API level for which the corresponding entity has been created and enabled and the MTP3 database of the BEP hosting the service application completes its initialization and becomes available;
- A service application is running with an API level for which the corresponding entity has been created and enabled, the first FEP restarts and the MTP3 database becomes available;
- A service application is running with an API level for which the corresponding entity has been created and enabled, the MTP3 database is available, the associated DIR restarts and the management application performs a successful association.

2.1.3.4 Non-operational LSI

The management application receives a non-operational LSI when:

- The status of an API level changes from *operational* to *non-operational*;
- The status of an API level changes from *unreachable* to *non-operational*;
- The status of an API level changes from *reachable* to *non-operational*.

The following events trigger these transitions:

- An API level entity is successfully disabled by a management command;
- An entity is recreated that represents an existing API level in operational status after the restart of the BEP;
- A service application is running with an API level for which the corresponding entity has been created and disabled and the management application performs a successful association;
- A service application is running with an API level for which the corresponding entity has been created and disabled and the associated BEP restarts;
- A service application is running with an API level for which the corresponding entity has been created and disabled, the associated DIR restarts, and the management application performs a successful association;
- A service application is running with an API level for which the corresponding entity has been created and enabled and the local BEP restarts;
- A service application is running with an API level for which the corresponding entity has been created and enabled and the last remaining FEP fails, causing the MTP3 database to become unavailable;

Note



The MTP3 database becomes unavailable when the last connected FEP fails and the MTP3 database is no longer authoritative. The MTP3 database also becomes unavailable when the BEP with which it is associated is being initialized and the MTP3 database is no longer authoritative.

- A service application is running with an API level for which the corresponding entity has been created and enabled when the associated DIR restarts, but the MTP3 database is unavailable.

The following transition does not raise an LSI:

- The status of an API level changes from *reachable* to *non-operational*.

The following event triggers this transition:

- An API level entity is successfully created by a management command.

2.1.4 Monitoring the IPsec

The procedure described in this section provides steps to test the IPsec connection and to dump the SA database (to check the algorithm in use).

1. To test the IPsec connection, run the command to view the network packets being transferred between the hosts and verify that they are encrypted via IPsec:

```
tcpdump -i ethx
```

Each packet must include an authentication header (AH) and must be shown as ESP (encapsulating security payload), i.e. encrypted packet. For example:

```
17:13:20.617872 ipstg1 > ipstg2: \
AH(spi=0x0aaa749f,seq=0x335): ESP(spi=0x0ec0441e,seq=0x335) (DF)
```

2. To dump the SA database and check the algorithm in use, type the command `setkey -D`. For example:

```
[root@pro18 etc]# setkey -D
10.0.0.1 10.0.0.1
esp mode=transport spi=171722014(0x0a3c451e) reqid=0(0x00000000)
E: 3des-cbc 0ca0e933 27c662b7 f306bf35 d48d6633
A: hmac-sha1 8ef41257 3d8b49ff f3ebfbb8 2d4844d0 72eec9a4
seq=0x00000000 replay=4 flags=0x00000000 state=mature
created: Feb 6 10:40:54 2008 current: Feb 6 10:42:42 2008
diff: 108(s) hard: 3600(s) soft: 2880(s)
last: hard: 0(s) soft: 0(s)
current: 0(bytes) hard: 0(bytes) soft: 0(bytes)
allocated: 0 hard: 0 soft: 0
sadb_seq=7 pid=22290 refcnt=0
10.0.0.1 10.0.0.2
ah mode=transport spi=38429521(0x024a6351) reqid=0(0x00000000)
A: hmac-sha1 9d43dd69 33561368 1726bedf ba49d0b1 9f38eb62
seq=0x00000000 replay=4 flags=0x00000000 state=mature
created: Feb 6 10:40:54 2008 current: Feb 6 10:42:42 2008
diff: 108(s) hard: 3600(s) soft: 2880(s)
last: hard: 0(s) soft: 0(s)
current: 0(bytes) hard: 0(bytes) soft: 0(bytes)
allocated: 0 hard: 0 soft: 0
sadb_seq=6 pid=22290 refcnt=0
```

2.1.5 Inactive AS management

The SG is based on the SCP mechanism, so if an AS becomes unavailable the following processing occurs:

- No TFP are sent on the SS7 side.
- SSP(s) (with PC and SSN destination values) are sent to all PCs present in the concerned PC list of the AS, if the AS is described with an RK containing a PC+SSN (and only in this case).

2.1.6 Traffic distribution and diversion in case of failure

Once an inbound message has been matched to a routing key, the SGP knows the AS identifier. The SGP must then select an active ASP/route for the AS and forward the message to the AS. ASP/route selection also depends on the routing delivery mode as well as the message type and content.

There are three routing delivery modes on the IP side: round-robin, loadsharing on calling party address, and TID (transaction identifier of dialog).

- Round-robin mode loadshares the traffic equally among all available routes.
- Loadsharing on calling party address mode selects the route according to the calling party address.

- TID mode is for continuing message TCAP only. It selects the route according to the TID field.

Note



Round-robin and loadsharing on calling party address cannot be mixed; they are mutually exclusive. TID mode must be associated with another mode.

At management configuration time, you can set the routing delivery mode by defining both the STATESHARE characteristic and the LOADSHARING characteristic of the application (DESTINATION entity).

Set the STATESHARE characteristic to:

- YES, for round-robin delivery loadsharing. (In this case, further settings are required on the LOADSHARING characteristic, see below.)
- NO, for TID delivery.

The STATESHARE characteristic combines with the LOADSHARING characteristic settings to define the delivery mode you want to implement, either round-robin or loadsharing on calling party address.

2.1.6.1 Loadsharing delivery

Loadsharing is a method used by Bee-CUSP CSP Front End to select an ASP among all the ASPs that can reach the SUA AS. Bee-CUSP CSP Front End offers three loadsharing mechanisms:

- round-robin delivery, altered by weights
- loadsharing on SCCP calling party address delivery
- loadsharing on B-Address (BADDR) delivery.

These mechanisms are detailed below.

Round-robin delivery

Round-robin is the default mode. The round-robin approach is a load-share routing with a weighting attached to each route. Round-robin routing is appropriate for either SCCP applications or TCAP applications that do not require dialog context maintenance. For more information, refer to the Traffic distribution table below.

Loadsharing on SCCP calling party address delivery

This feature implements a loadsharing algorithm used to select the ASP depending on the calling party address content. That is, once an AS has been selected a first time for a given value in its calling address field, each time a message arrives with the same calling party address, it will be automatically routed to the same ASP (within the given AS). The calculated route range depends on the route weight.

The calling party address content used to calculate the route value is GTAI. If there is no GTAI present, it uses the calling PC or MTP3 originator PC. This value is compared with the route range.

- If the route is available, the associated ASP/route is elected.
- If the route is momentarily unavailable, its traffic will be dispatched among other available routes.

Loadsharing on B-Address delivery

Loadsharing on B-Address (BADDR), also referred to as binding on B-Address, applies the same principle and the same algorithm as *loadsharing on SCCP calling party address* deliver above, except that it is based on the B-Address instead of the calling party address. In this case the loadsharing mechanism computes an CRC-32c value from the B-Address message field.

For further information, see 'Traffic handling' in the Bee-CUSP CSP installation and configuration guide.

Allocating a weight to a route — ASP weight

When allocating a weight to a route, you can set either a positive or a negative value. Routes set with a positive (and non-zero) weight are used in priority. However, routes set with a negative weight will become active if all other routes become unavailable. In this case, if an AS is composed of several negatively-weighted ASPs, the traffic is distributed among these ASPs according to the absolute value of their weight. Using the ASPs' positive and negative weight values, you can configure the routes as either master or backup, as detailed in the examples below.



Note

A zero weight is permitted and can be used on a temporary basis. A route with a weight of zero is excluded from the routing table. No message is sent to the corresponding ASP.

Example

AS1 is composed of:

- ASP1 => weight 1 (active)
- ASP2 => weight -1 (active)

In this case, ASP1 handles 100% of the traffic intended for AS1.

Example

AS1 is composed of:

- ASP1 => weight 1 (down)
- ASP2 => weight -1 (active)

In this case, ASP2 handles 100% of the traffic intended for AS1.

Example

AS1 is composed of:

- ASP1 => weight 1 (down)
- ASP2 => weight -1 (active)
- ASP3 => weight -3 (active)

In this case, the traffic intended for AS1 is shared between ASP2 and ASP3 as follows: ASP2 handles 25% of the traffic and ASP3 handles the remaining 75% of the traffic.

2.1.6.2 TID delivery

The TID approach applies to TCAP applications that must maintain dialog context. It is used to manage TCAP dialogs and route TCAP messages of the same dialog to the same application copy (DESTINATION entity) via the same route (ASP), and in-sequence.

Messages *starting* (BEGIN) a TCAP dialogue are routed using the round-robin or loadsharing on calling party address algorithm, while messages *continuing* (CONTINUE) or *ending* (END) a TCAP dialog are routed using the TID routing delivery mode. The TID field is extracted and then used to route the message to the appropriate ASP. The relevant part of message TID is compared to the TID range for which all ASPs are registered.

- If one of them matches, the associated ASP is selected.

- If the TID of the message (the relevant part of the message TID) is not registered for any ASP, the dialogue is aborted by sending back a P-ABORT/REJECT message.

In the case of a distributed platform, the routing mechanism favors the local SCTP association. However, if none is available, the message is sent to another SGP, if available. A mechanism avoids mis-sequencing. For more information, refer to the 'Traffic distribution' table below.

2.1.6.3 Traffic distribution

Table 1: STATESHARE and LOADSHARING characteristic combinations

STATESHARE characteristic	LOADSHARING characteristic	
	RRW setting*	CGPA setting
YES setting	Round-robin delivery applies.	Loadsharing on SCCP calling party address applies.
NO setting	For SCCP or TCAP (BEGIN only), round-robin delivery applies. For all other TCAP messages (CONTINUE, END) TID delivery applies.	For SCCP or TCAP (BEGIN only), loadsharing on SCCP calling party address applies. For all other TCAP messages (CONTINUE, END) TID delivery applies.

*For further details on traffic distribution applicable with this setting, refer to the table below.

Table 2: Traffic distribution

Type of message	SGP has (at least) one active ASP	SGP has no active ASP at all
Class 0 Begin, Query, Unidirectional	Use loadsharing capability.*	Reply with a UDTS/XUDTS with unequipped user return cause if return on error QOS is set.
Class 0 Continue, End, Abort	If application (Destination) is in stateshare: Use loadsharing capability.* If application (Destination) is not in state-share: Use TID routing, depending on the ASP TID range that serves the concerned AS.	Reply with a UDTS/XUDTS with unequipped user return cause if return on error QOS is set.
Class 1 Begin, Query	Use loadsharing capability.*	Reply with a UDTS/XUDTS with unequipped user return cause.
Class 1 Continue, End	If application (Destination) is in stateshare: Use loadsharing capability.* If application (Destination) is not in state-share: Use TID routing, depending on the ASP TID range that serves the concerned AS.	Reply with a UDTS/XUDTS with unequipped user return cause if return on error QOS is set.
Class 1 Unidirectional	Use loadsharing capability.*	Reply with a UDTS/XUDTS with unequipped user return cause if return on error QOS is set.

* Use loadsharing capability: either loadsharing on calling party address (CGPA) or round-robin (RRW) with weight distribution between active ASPs that serve the AS concerned.

2.1.7 PC-State and N-PC-State

PC-State and N-PC-State are forwarded to all active application servers.

2.1.8 N-COORD primitives

Bee-CUSP CSP Front End does not support the N-COORD primitives.

2.1.9 Bee-CUSP CSP Front End congestion management

2.1.9.1 Congestion management at the SUA level

Congestion in transmission (from SG to AS)

There are three possible causes of congestion:

1. Congestion of the SUA buffer pool. A buffer has to be reserved in the pool every time a message is to be sent to an AS. It is released once SCTP notifies SUA that the message has been sent.
2. Congestion of the SCTP ASSOCIATION transmit buffer. The association has a queue of messages that are waiting to be sent (in fact, pointers to buffers in the pool). The queue grows if the SCTP window is closed.
3. Congestion of the remote AS. A SCON message is sent by an ASP to Bee-CUSP CSP Front End to indicate that the remote AS is in a state of congestion.

Congestion of the SUA buffer pool

The SUA management entity defines the following characteristics, events, and counters:

- The SUA pool size (NBBUFF),
- The SUA pool congestion thresholds (CGONSET, CGABAT, CGDISC),
- The current SUA pool congestion and discard level (CONGLEVEL, DISCLEVEL),
- The priority of messages indicating whether they can be sent when the DISCLEVEL threshold is reached (ASPSMPRIO, ASPTMPRIO, ASPRKMPRIO, ASPMGMPRIO, ASPSSNMPRIO),
- The current number of free buffers in the SUA pool (NBFREEBUFF),
- Congestion events (CONG, NOCONG, DISCCONG),
- The counters of discarded messages according to their class (NB DATADISCG, NBASPTMDISCG, NBASPSMDISCG, NBRKMDISCG, NBMGMTDISCG, NBSSNMDISCG),
- The counters of messages exchanged with the SCCP layer (NBUDTSRCV, NBUDTSNT, NBXUDTSRCV, NBXUDTSNT, NBUDTRCV, NBUDTSNT, NBXUDTRCV, NBXUDTSNT),
- The global counter of discarded messages NBMSGDISCG.

When the CGONSET threshold is crossed:

- The CONG event is raised,
- CONGLEVEL takes the value 1,
- The message is sent anyway.
- When the CGABAT threshold is crossed:
- The NOCONG event is raised,
- CONGLEVEL takes the value 0.

When the CGDISC threshold is crossed:

- DISCLEVEL takes the value 1 (back to 0 when it is crossed in the other direction),
- DATA messages are discarded,
- ASPTM, ASPSM, RKM, MGMT and SSNM messages with a priority characteristic of 0 are discarded, others are sent,
- The discard counter corresponding to the message class is incremented (if the message is discarded)
- The global counter for discarded messages (NBMSGDISCG) is incremented (if the message is discarded).

When there are no more buffers in the pool:

- All messages are discarded,
- The discard counter corresponding to the message class is incremented,
- The global counter for discarded messages (NBMSGDISCG) is incremented,
- In the case of a DATA message from the SS7 side, a (X)UDTS is sent back according to the message QOS.

Congestion of the SCTP ASSOCIATION transmit buffer

The ASSOCIATION management entity defines the following characteristics, events and counters:

- The current congestion level of the association transmit buffer CGTBLEVEL,
- Congestion events (CONG, NOCONG),
- The counter of discarded messages NBMSGDISCG.

In addition, the SUA management entity defines characteristics that are common to all associations:

- The number of buffers in association transmit buffers CGTBNBBUFF.
- The congestion thresholds for association transmit buffers CGTBONSET, CGTBABAT.

When the GTBONSET threshold is crossed, a flow control procedure is performed as follows:

The message is inserted in the queue and the thread is put in the sleep state in order to give time to the SCTP layer to empty the queue. The CONG event is raised.

When the CGTBABAT threshold is crossed, the flow control procedure ceases. The NOCONG event is raised.

When the transmit buffer is full, the message is discarded. The thread is also put in the sleep state. The counters NBMSGDISCG of the ASSOCIATION entity and NBMSGDISCG of SUA entity are incremented.

Congestion of the remote AS

No particular action is taken if the ROUTE / DESTINATION becomes congested, for example, when a SCON with PC and SSN is received (the SCCP layer does not manage SSC). Only the counters NBSCONRCV of the affected ROUTE and DESTINATION are incremented.

Congestion in reception (from AS to SG)

When the SCTP layer receives a message, it is copied to the ASSOCIATION reception buffer and stays there during its processing by the SUA layer. The buffer is then released and the `read()` function is re-entered.

In the case of a DATA message, it must be forwarded to the SCCP layer through the FEP. At that point, a FEP free buffer is reserved and the message is copied into it. If the FEP is congested (see FEP_SUBSYSTEM characteristics), the DATA is discarded and the SCCP layer raises a PC-STATE indication. SUA then sends back a SCON (with the PC of the called address, no SSN) to the AS.

2.1.9.2 Congestion management at the M3EP level (M3UA endpoint capability)

Remote congestion

Remote congestion can be due to destination-specific path congestion (interpretation of TFC) or to an adjacent signaling gateway or IPSP. This kind of congestion must be stored on a destination route basis for SCON.

If the congestion is due to the SG (or IPSP), it affects all the destinations that use this SG (or IPSP) as a route. When this SG (or IPSP) sends a SCON message to the platform, the affected PC in the SCON must be the destination PC. The PC of the SG (or IPSP) **must not** be used because the upper layer does not necessarily know of it.

If the same level of congestion affects all delivered destinations, the SG can send a SCON message with a generic, wildcarded PC (the mask is 24 for ANSI or any other value for ITU and China).

A SCON message received on an M3UA endpoint (M3EP) association is converted into an MTP_STATUS (PC_STATE) indication about the affected PC, which is passed on to the upper layer. In the case of national congestion mode, the level of congestion must be consolidated between all routes for a destination.

In the case of international congestion mode handling, where only a SCON message is returned, no state has to be maintained in the M3UA endpoint layer. In case of national congestion mode handling, a congestion message is valid unless contradicted by another SCON message or if a timer expires. The ASP periodically sends a DAUD message when a remote SS7 PC is known to be congested.

Local congestion

The following are the possible causes of local congestion:

- An association becomes congested after SCTP stops reading (TX queue).
- The global pool of free M3UA buffers reaches a low level.
- The upper layer cannot provide a free buffer to transfer messages.

Bee-CUSP CSP Front End incoming traffic

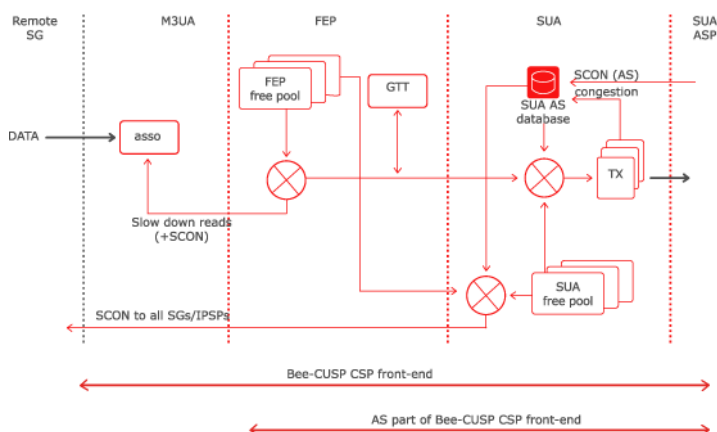
The figure below illustrates the congestion model used for M3UA endpoint (M3EP) incoming messages from a remote SG (or IPSP). The received SCTP message is stored locally in a static buffer, then, after a consistency check, it is delivered to the upper common level 3 layer.

- In the case of congestion of the FEP free pool of buffer, the read on the incoming association is delayed for a time. In the NATIONAL option, this timer value depends on the level of congestion.
- If the priority of the message is lower than the Bee-CUSP CSP Front End internal level of congestion, the message is discarded and a SCON may be sent to the remote SG or IPSP. The sending of the SCON is conditioned by an environment variable.
- To limit the number of broadcasts, these SCON messages are sent every N received messages, N being greater than the equivalent in MTP3, and N being specified via the management configuration with a default value (see the CGSTATN characteristic of the M3EP layer).

If a buffer is available to deliver the message to the upper layer, it is then scheduled as an SCCP message. When the user message data is handled by the SCCP layer, the GT translator can be invoked to determine the final addressee.

- In the case of congestion of the SUA path, which comes from the SUA free pool or AS state, the message is silently discarded. A congestion indication received by the SUA level itself (upon receipt of a SCON coming from an SUA ASP) is simply discarded.

Fig 3: Front end incoming traffic



Bee-CUSP Convergent Signaling Platform operating guide

The ASP never receives DAUD messages for a congestion audit; there is no need for the M3UA endpoint layer to keep any information about the upper layer's state of congestion.

Bee-CUSP CSP Front End outgoing traffic

The figure below illustrates the congestion model used for M3UA outgoing messages to a remote SG or IPSP.

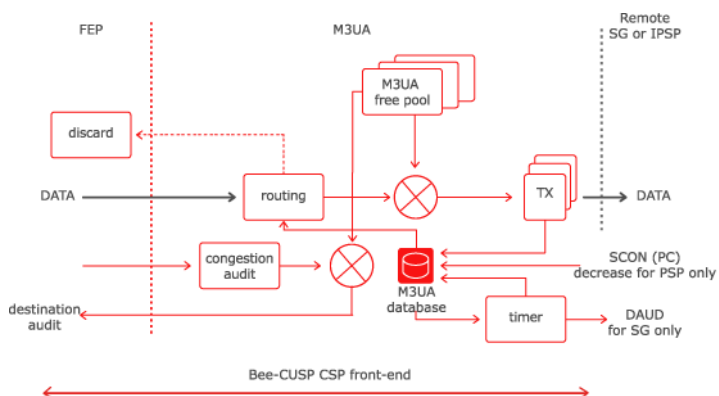
When a message is eligible to be sent by the M3UA endpoint layer, the `send` procedure is invoked. After a further availability check, message priority is compared to the *used path congestion level*. This means that if a message for a given SLS is sent on an association that is not congested, the message will be sent even if another association on the same route is congested. The test for the congestion level of outgoing messages is performed according to the SLS value.

You can assign a very high priority to SCCP management messages when congestion occurs for a destination, which allows you to use *reserved* resources.

A consolidated congestion state for the destination is used to answer audits from the upper layer. This test does not depend on the SLS value but represents the worst congestion value of all the possible paths for the distributed platform. A destination congestion state arises if one of the following parameters reaches its maximum level:

- congestion state of an outgoing association (i.e. the outgoing association with the worst value)
- congestion state for the M3UA global free pool of buffers
- congestion state of the remote SG or IPSP
- congestion state of the FEP free pool

Fig 4: Front end outgoing traffic



When a SCON message is received from a remote SG or IPSP, an indication is sent to the upper layer (SUA). This indication includes the local congestion level, which can be above the level given in the SCON message (MTP_STATUS indication).

Note that an individual association congestion can induce congestion for all the destinations that may use it. An association congestion leads to the highest congestion level state in the NATIONAL option. In this case, the global route is affected for the consolidation of the remote destination XXX and the maximum for local associations.

2.1.9.3 Multi-ASP

Given that the Bee-CUSP CSP Front End platform is distributed amongst several processes, there is a need to distribute congestion handling across all its processes so that any audit coming from the upper layer can return a consistent answer for a destination, whichever process is audited.

For congestion audits from the upper layer, in a well-balanced and stable platform, the congestion level for a destination is the same for each ASP. However, if the platform is not well-balanced or unstable, the value may be different from one ASP to another.

SSNM indication distribution

The remote SG **must** send SSNM messages to all the ASPs in the Bee-CUSP CSP Front End platform. If one of the ASPs is not UP, its congestion state may be inconsistent compared to that of the other ASPs in the platform. A broadcast mechanism between ASPs guarantees platform consistency.

SCTP association congestion distribution

Individual association congestion can affect several destinations that use it. Once an association is congested, it is reported to all routes and then, if it is used in the routing table, to all the destinations that may use it.

Association congestion handling can be executed by both the SG and IPSP objects. The broadcast between ASPs is limited to the consolidated congestion state transitions. Two consecutive association transitions only trigger a single update message between ASPs.

2.1.10 Multiple point codes

Multiple point codes allow a Bee-CUSP CSP Front End platform to process incoming SS7 traffic for different MTP3 point codes. This is used to identify SIGTRAN application servers with point codes that are different from those of the Bee-CUSP CSP Front End platform.

2.1.10.1 Configuring alias point codes

To create an alias point code, enter the `S7MP` command:

```
ADD FEP * /ALIAS_PC_LIST=(X,Y)
```

S7MP commands for managing alias PC lists

To update the alias PC list, you can only use the `ADD` and `REMOVE` commands. These two directives are provided by the `FEP_SUBSYSTEM` entity. You can also use the `SHOW` command to display the current list of alias point codes supported by the platform.

Using the `S7MP` management program, you can enter the following commands:

- `ADD FEP * /ALIAS_PC_LIST=(X,Y)`
- `REMOVE FEP * /ALIAS_PC_LIST=(X,Y)`
- `SHOW FEP * /ALIAS_PC_LIST`

Example - Creating an alias

```
S7MP> ADD FEP * /ALIAS_PC_LIST=(700, 701)
S7MP> ADD FEP * /ALIAS_PC_LIST=703
```

Example - Deleting an alias

```
S7MP> REMOVE FEP * /ALIAS_PC_LIST=(700, 701)
S7MP> REMOVE FEP * /ALIAS_PC_LIST=703
```

2.2 SNMP agent

The SNMP agent enables a network management application, such as OpenView, to monitor Bee-CUSP CSP platforms. The simple network management protocol (SNMP) runs on the transport user datagram protocol

(UDP). It enables the manager to communicate with the SNMP agent to interrogate the Bee-CUSP CSP management information base (MIB).

The objects in the MIB are mapped onto the Bee-CUSP CSP entities and specified entity attributes, so that there is a correlation between a MIB object and a Bee-CUSP CSP constant. The MIB provides sets of tables to structure the information obtained through this mapping process from the Bee-CUSP CSP management database.

The SNMP agent translates SNMP Get or GetNext requests from the Manager into Bee-CUSP CSP SHOW directives for the Director. When the Director responds, the information is translated by the agent into an SNMP response. If the Director message is an event, it is translated into the corresponding SNMP trap.

For further information on monitoring using SNMP, see [Using the SNMP agent](#).

2.3 Black box facility

The black box facility continuously monitors the state of the FEP and SNMP, together with BEP, DIR processes, and:

- Following a reboot, automatically starts the processes of the last `platform_id` that was started by the `ss7startup` command (without the `nobootstart` option)
- Following a process crash on a running platform, automatically attempts to restart the process or processes that crashed

Depending on certain conditions, the black box facility also automatically reconfigures the platform.

2.3.1 Reconfiguration

After starting or restarting a machine the black box facility invokes the last saved configuration file on the platform, and the BEP uses the configuration file stored in:

```
/usr/var/ss7/platform_<id>/save_config/  
SS7_SAVE_CONFIG_BEP_[machine]_[platform_id].DAT
```

For a static reconfiguration, the utility relies on the fact that the platform was previously been configured and that the configuration was saved using the `SAVE_CONFIG S7MP` command.

Note



To restore event reporting, the S7MP tool must be restarted with a session id that identifies the connection between the S7MP management interface and the Director process. See the example in the section below.

Note



Up to 26 platforms can be configured. To avoid overloading system resources, the automatic restart of processes is performed for only one `<platform_id>` at a time, and this is the last `<platform_id>` to be started with `ss7startup`.

2.3.2 Platform startup procedure to enable black box facility

To enable the black box facility, the platform must be configured and started in the following manner:

1. Create the platform configuration using the `ss7configure` tool.

This defines the processes and device (DNB), and configures the SS7 and IP parts; that is, the MTP2, MTP3 and SCCP levels, and the M2PA, M3EP and SUA levels.

The FEP configuration file generated by `ss7configure` is saved in

```
/usr/var/ss7/platform_<Id>/s7mp/fep_gen_<Id>.s7mp
```

The BEP configuration file generated by `ss7configure` is saved in

```
/usr/var/ss7/platform_<Id>/s7mp/bep_gen_<Id>.s7mp
```



Note

The mandatory layers, that is, BEP, FEP, MTP2, MTP3, SCCP and SUA, must be created and enabled prior to platform connection.

2. Start the platform with the command:

On the FEP machine:

```
ss7startup [<machine>] <platform_id> fep dir
```

On the BEP machine:

```
ss7startup [<machine>] <platform_id> bep dir
```

Example where platform id = **d** and **pro20** BEP machine with application **app88**, **pro21** FEP machine:

```
ss7startup pro21 d fep dir
ss7startup pro20 d bep app88 dir
```

3. Open an S7MP session with a <platform id> and, if required, a <session id>:

```
setenv FEP_NAME '*'
setenv BEP_NAME 'pro20'
s7mp<platform_id> [<session_id>]
```

Example where platform id = **d** and BEP machine is **pro20**:

```
s7mpd
```

(The <session_id> is optional.)

4. Apply the configuration with the DO command.

Example where platform id = **d**

```
S7MP_D>do /usr/var/ss7/platform_d/s7mp/fep_gen_d.s7mp
S7MP_D>do /usr/var/ss7/platform_d/s7mp/bep_gen_d.s7mp
S7MP_D>do /usr/var/ss7/platform_d/s7mp/app_app88_pro20_d.s7mp
```

5. Optional: If you want to make additional changes to your configuration:
 - Modify the configuration again using S7MP.

- Save the configuration:

```
S7MP_D>save_config
```

The FEP configuration file generated by the `SAVE_CONFIG` command is saved in

```
/usr/var/ss7/platform_<Id>/save_config/SS7_SAVE_CONFIG.DAT
```

for each machine in the platform.

The BEP configuration file generated by the `SAVE_CONFIG` command is saved in

```
/usr/var/ss7/platform_<Id>/save_config/  
SS7_SAVE_CONFIG_BEP_[machine]_[platform_id].DAT
```

for each BEP machine in the platform.

6. Optional: If you want to monitor the platform with SNMP, start the SNMP agent on a FEP machine with `ss7startup`

```
ss7startup [<machine>] <platform_Id> snmp
```

Example where platform id = **d**:

```
ss7startup d snmp
```

In subsequent reboots or restarts the black box facility will be effective.

2.3.3 Disabling automatic process restart

The black box facility is enabled by default. However if the facility needs to be disabled for individual processes while the platform is running, for example during debugging or changing hardware, it can be done as follows:

1. Set the environment variable `SS7_PROCESS_AUTO_RESTART` to `OFF`.

By default the variable `SS7_PROCESS_AUTO_RESTART` is set to `ON`. To change the value of the variable you need to make an entry in the `ss7_parameters` file, in the `/usr/var/ss7` or `/usr/var/ss7/platform_<platform_id>/config` directory of the required BEP or FEP machine. You may need to create the file itself.

Add the following sequence to set the variable:

```
#!/bin/sh  
SS7_PROCESS_AUTO_RESTART=OFF  
export SS7_PROCESS_AUTO_RESTART
```

Auto-restart of the BEP or FEP, DIR, application and SNMP agent processes is disabled after the platform has been stopped and restarted (using `ss7shutdown` and `ss7startup`).

For more information on environment variables, see the *Installation and configuration guide*.

2. Stop the processes using `ss7shutdown <platform_Id> <process>`

When the BEP or FEP, DIR, application or SNMP agent processes are stopped using `ss7shutdown`, the black box facility is disabled and these processes will not be automatically restarted if they crash.

To restart the processes, you must use the `ss7startup` utility (see next step).

3. Restart the processes using `ss7startup <platform_Id> <process>`

To restore the black box function:

1. Remove the entries in the `ss7_parameters` file that were added to disable the facility.
2. Stop the processes using `ss7shutdown <platform_Id> <process>`
3. Restart the processes using `ss7startup <platform_Id> <process>`

2.3.4 Disabling automatic process startup at reboot time

Since the black box facility is turned ON by default, it automatically starts the Bee-CUSP CSP processes when a machine is rebooted. If this behavior is not required, automatic process startup can be disabled using the `ss7shutdown` command, as follows:

```
ss7shutdown [<machine>] <platform_id> ss7_all nobootstart
```

The `nobootstart` parameter prevents the started `<platform_id>` from being available at reboot time, so there is effectively no platform identified to restart processes on.

2.4 Incoming user traffic interceptor

The incoming user traffic interceptor feature allows you to capture incoming SS7 traffic according to a set of predefined filters. The captured traffic is duplicated and sent through a UDP socket to an external application called 'MSU collector'. The incoming user traffic interceptor enables you to gather detailed statistics, perform checks on byte lists, generate call detail records (CDRs), etc. It neither lowers the routing capabilities nor the performance of Bee-CUSP CSP.

The MSU collector is not delivered with Bee-CUSP CSP. It is your responsibility to develop this application. For a test application, contact your Buzzinbees SAS representative.

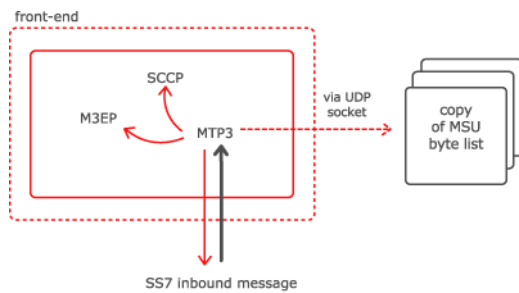
For troubleshooting information regarding the incoming user traffic interceptor, see the *Bee-CUSP CSP troubleshooting guide*.

2.4.1 Using the incoming user traffic interceptor

The incoming user traffic interceptor can be activated for both ANSI and ITU-T (not for CHINA), on messages coming from SS7 connectivity (NSL, HSL/ATM, HSL/MTP2 or M2PA layers) and targeted at any direction (either level 2 or M3EP). It is controlled and activated through environment variables read at FE process startup (see [Activating the incoming user traffic interceptor](#)).

The figure below shows the general architecture of the incoming user traffic interceptor.

Fig 5: The incoming user traffic interceptor



Traffic is intercepted at the MTP3 level and copied to the MSU collector according to filtering rules based on MTP OPC, DPC and SI parameters (for more information on filtering, see **Filtering traffic**). Only user messages are captured, network management messages are not captured.

The MSU collector can be co-located with one of the FE processes of Bee-CUSP CSP or run on a separate server. The MSU collector receives intercepted MSUs through a UDP socket. The UDP protocol is datagram-based.

Packets are sent individually. They have defined boundaries, which guarantee that there is neither a split, nor a merge in data streams. SS7-segmented packets are sent as they are, no reassembly is performed before they are sent to the MSU collector application.

The message received by the MSU collector is composed of the MTP MSU SIF byte list, according to MTP recommendations.

ITU-T example

```
43
03 C0 02 20
09 00 03 0A 0E 07 12 06 08 12 01 55 11 04 43 0B 00 07 08 62 04
02 00 00 38
```

This byte list can be interpreted as follows (the figures are in hexadecimal format):

```
SIO = 43
Routing label = 03 C0 02 20
Payload = 09 00 03 0A 0E 07 12 06 08 12 01 55 11 04 43 0B 00 07 08 62
06 48 04 02 00 00 38 (SCCP packet)
```

Limitations

The following limitations apply and mean that you cannot entirely rely on the data collected using the incoming user traffic interceptor.

- Traffic is captured **before** the actual routing is processed. As a consequence, the MSU byte lists sent to the MSU collector include all the traffic intercepted at this stage and can contain messages discarded during the actual routing process, either due to destination unavailability or congestion.
- The UDP socket used to receive intercepted MSUs is in non-blocking mode. As a result, if the MSU collector is unavailable or too slow, or if the network is too slow, some byte lists may be lost since **no buffering or retransmission** is available.

2.4.2 Activating the incoming user traffic interceptor

To enable the incoming user traffic interceptor, go to the `ss7_parameters` file and set the following environment variables as detailed in the table below.

Once the environment variables are properly set, the FE process reads and interprets the filter file each time it is launched. As soon as a valid configuration file is read, the incoming user traffic interceptor feature is active and starts sending intercepted traffic to the UDP socket.

Table 3: Incoming user traffic interceptor environment variables

Environment variable name	Setting	Comments
SS7_MTP3_SPY_ACTIVE	SS7_MTP3_SPY_ACTIVE=ON or SS7_MTP3_SPY_ACTIVE=OFF	<p>If set to ON, the incoming user traffic interceptor is activated.</p> <p>If set to OFF or not set or set to any other string, the feature is not activated.</p>
SS7_MTP3_SPY_PORT_UDP	SS7_MTP3_SPY_PORT_UDP=<UDP port value>	<p>The MSU collector UDP port value is mandatory (in decimal format). If not set, the feature is not activated and no warning message is logged.</p> <div>  <p>Note</p> <p>This is a unidirectional connection, there is no message from the MSU collector to Bee-CUSP CSP.</p> </div>
SS7_MTP3_SPY_FILTER_FILE	SS7_MTP3_SPY_FILTER_FILE=<Filter file>	<p>The path and filename of the filter file that contains the filtering rules. For more information on filtering, see Filtering traffic.</p> <p>For instance,</p> <pre>SS7_MTP3_SPY_FILTER_FILE=/usr/ var/ss7/platform_<id>/ mtp3_spy_filter.dat</pre>
SS7_MTP3_SPY_HOSTNAME	SS7_MTP3_SPY_HOSTNAME=<Server name>	<p>Used to specify the hostname of the MSU collector server. While optional, it is recommended to set this variable even when the MSU collector is co-located with the Bee-CUSP CSP FE process in order to have a unique <code>ss7_parameters</code> file across a multiserver Bee-CUSP CSP platform. For instance:</p> <pre>SS7_MTP3_SPY_HOSTNAME=localhost SS7_MTP3_SPY_HOSTNAME=mozart</pre> <p>When this environment variable is used, the hostname must also be declared in the <code>/etc/hosts</code> file of each server belonging to the Bee-CUSP CSP platform and running this feature.</p>

2.4.3 Filtering traffic

A filtering mechanism is applied to select which MSU is sent to the MSU collector application. The filters uses rules defined in a specific filter file that you need to create. The environment variable `SS7_MTP3_SPY_FILTER_FILE` contains the filter file name and its location (see [Activating the incoming user traffic interceptor](#) for more information).

2.4.3.1 Creating valid filtering rules

To create valid filtering rules, the following conditions apply:

- each filtering rule should be a set of DPC, OPC, and SI
- each filtering rule must include at least one of these parameters: DPC, OPC, SI. If one parameter is missing, a wildcard is used instead.
- values must be entered in decimal format
- parameters must be within the following ranges:
 - DPC and OPC: between 0 and 16777215
 - SI: any value other than 0, 1 and 2 (only user part messages)
- 2.x can handle up to 500 rules
- use # to comment a filtering rule

The filter file is parsed at FE process startup and each time the command `load fep * mtp3` is run. If syntax errors are detected, these are shown in the filtering log `/usr/var/ss7/platform_<platform_id>/log/fep_<platform_id>.log`. For more information on the syntax of error output in the log file, see the *Bee-CUSP CSP troubleshooting guide*.

2.4.4 Filtering traffic mechanism

The set of filtering rules included in the filter file is parsed starting from the first line to the last line. When an MTP3 MSU matches at least one filtering rule, it is sent to the MSU collector. If one of the parameters is not specified in a given rule, a wildcard mechanism is applied for this parameter.

For instance, if the SI field is not present in a filtering rule, all kind of user traffic parameters matching the specified OPC and/or DPC ones are sent to the MSU collector, regardless of the SI (see the example below # All kind of user traffic from MTP3 OPC=2222 to MTP3 DPC=1111):

Rules file

```
# SCCP traffic from MTP3 OPC=123 to MTP3 DPC=3456
dpc=3456;opc=123;si=3;
# all ISUP traffic whatever the originator and the recipient
si=5;
# SCCP traffic coming from MTP3 OPC=45
opc=45;si=3;
# TUP traffic only
si=4;
# All kind of user traffic from MTP3 OPC=2222 to MTP3 DPC=1111
dpc=1111;opc=2222;
```

2.4.4.1 Notes on applying the filter file to the configuration

Filter file consistency throughout the platform

On a multiple front end platform, each FE process has its own configuration file and filter file. The system administrator must guarantee that all FE processes share the same configuration and filter file (either using NFS mount or by physically copying the configuration and filter files).

No filtering is applied if...

If the `SS7_MTP3_SPY_FILTER_FILE` environment variable is not set or if the specified filter file does not exist, all MTP3 MSU byte lists are sent to the MSU collector.

No MTP3 MSU byte list is sent if...

If the filter file is unreadable, empty or contains a syntax error, no MTP3 MSU byte list is sent.

2.4.4.2 Modifying and applying the modified filter file

You can modify the filtering rules without stopping and restarting any FE process. To reload a modified filter file on all FE processes dynamically, run the following S7MP command:

```
S7MP> LOAD FEP * MTP3
```

Note

The `S7MP LOAD` command must target all the FE processes (using the `*`). The same command targeting a particular FE process is not supported. Assuming the environment variables are properly set, the FE process reads and interprets the filter file each time it is launched. As a consequence, the self-reconfiguration commands `SAVE` and `RECONFIGURE` are not needed for a FE process to read again the filter file at startup time.

2.4.4.3 Filtering logs

For general monitoring of the filter file, you can retrieve the filtering log in `/usr/var/ss7/platform_<platform_id>/log/fep_<platform_id>.log`.

Filtering logs

```
Info: Using /usr/var/ss7/ss7_parameters file
%SS7-I-RULE_LOADING, Start loading MTP3 rules file - V3.0
%SS7-I-FILEOPEN, File opened, start loading
%SS7-I-RULE_CHECK, RULE 0
%SS7-I-RULE_CHECK, RULE 1
%SS7-I-RULE_CHECK, RULE 3
%SS7-I-RULE_CHECK, RULE 4
%SS7-I-RULE_END, End of loading phase
SS7_MTP3_SPY_ACTIVE is set to ON
SS7_MTP3_SPY_PORT_UDP is set to 4444
SS7_MTP3_SPY_HOSTNAME is set to cepe
SS7_MTP3_SPY Filter file is set to /usr/var/ss7/platform_i/mtp3_spy_filter.dat
SS7_MTP3_SPY Filter file parsing is OK
INFORMATION: SCCP_SLS8_BITS=0 (SLS 8 bits feature activated)
SS7-I-RECONF: 11/07/2008-10:08:01-Fep auto reconfiguration not activated
```

If the filter file contains syntax errors, the log includes:

```
SS7_MTP3_SPY Filter file parsing is KO

SS7_MTP3_SPY Create Filter Table : KO
```

2.5 Recommendations when developing an MSU collector application

When developing an MSU collector application, the control and operation of the MSU collector should not be performed by Bee-CUSP CSP. Consequently, the MSU collector application returns the state of Bee-CUSP CSP (for instance, if an Bee-CUSP CSP FE process restart is in progress or is about to start) and what actions are appropriate, such as re-initializing the MSU collector. On the other hand, Bee-CUSP CSP is never directly informed of the current state of the MSU collector.

2.6 The logging and tracing framework

Comprehensive logging and tracing facilities are provided for platform monitoring and troubleshooting.

- **Logging:** The logging mechanism records platform events, providing a means of capturing specific platform activities and unusual but significant events. Logs are created that allow platform operators to monitor the functioning of the platform and to later check what has happened in the case of platform malfunction.
Logs can be used to generate SNMP traps in order to provide notifications of certain platform conditions. See **Using the SNMP agent**.
- **Tracing:** The tracing mechanism makes it possible to troubleshoot problems in real-time during platform operation. The traces produced complement the platform logs; their purpose is to provide more detail about what is happening on the platform.
Tracing can be implemented on the trace points of the CTF (common trace facility); trace points are places where information is collected and analyzed on specific exchanges between components of the platform. Tracing makes it possible to record network protocol or internal messages as they pass a trace point. These CTF traces can be retrieved using the CTF utility.

Logging and tracing are fully described in the *Bee-CUSP CSP troubleshooting guide*.

Logging

Logs are the main way to obtain information about the functioning of the platform. All events that affect the ability of the platform to perform its telecommunications function should be logged in order to allow the operator or administrator to take appropriate action.

The logging is based on the standard logging and tracing mechanism, `nettl`. This provides information about system network activity such as state changes, errors and connections. Events are assigned log severity levels but the logs are passed to the system without severity filtering, allowing the user to choose which severity levels are displayed.

For information on how to set up and use logs, refer to the *Bee-CUSP CSP troubleshooting guide*.

Tracing

The tracing mechanism allows real time problem troubleshooting during platform operation. Traces complement logs by providing more detail on events that occur on the platform. Tracing can be used to record both normal and abnormal platform activities, provided that the data traced is useful (so as not to unduly affect platform performance).

Tracing is based on the standard logging and tracing mechanism, `nettl`. The tracing part of the tool takes snapshots of inbound and outbound packets going through the network, as well as loopback and header information. All the data related to a process or component is captured, and then user-defined filters are applied. Traces are categorized by trace level and functional area, which allows for them to be filtered accordingly.

For information on setting up and using traces, see the *Bee-CUSP CSP troubleshooting guide*.

**Caution**

Tracing adversely affects platform performance and must only be used at the request of Buzzinbees SAS support.

CTF facility

The common trace facility (CTF) is a utility that allows users to collect and display information about specific protocol exchanges between systems in a network. This information can be used for troubleshooting. For further information, refer to the *Bee-CUSP CSP troubleshooting guide*.

2.7 Failfast

For further information see the *Bee-CUSP CSP product description*.

Chapter 3. The Bee-CUSP CSP Front End SNMP agent MIB

This section describes the overall structure of the Bee-CUSP CSP Front End management information base (MIB) and correlates it with the relevant Bee-CUSP CSP Front End entity attributes. It also provides the containment tree for each entity in the MIB tree. This document assumes that you are familiar with the simple network management protocol (SNMP), its conventions, and the typical structure of MIB containment and registration trees. In accordance with the SNMP protocol, the Bee-CUSP CSP Front End MIB uses a single tree structure for both the containment and the registration tree.

If you are using the Bee-CUSP CSP Front End with the Bee-CUSP CSP back-end product, you must run the SNMP agent from a front-end machine (i.e. from Bee-CUSP CSP Front End).

3.1 Overview

The MIB is situated in the Enterprise branch of the MIB containment tree. Each MIB object is described using the ASN.1 macro object type defined in the structure of management information (SMI). The SNMP SMI considers an object to be a model object type and the identification of an instance is not specified in the SMI.

The macro object type allows the definition of the following:

- Syntax
- Access (Inaccessible, Read-only)
- Status (Mandatory, Optional, Obsolete, or Depreciated)
- The object Identifier in the registration tree. The default value and the index are defined to identify an instance.

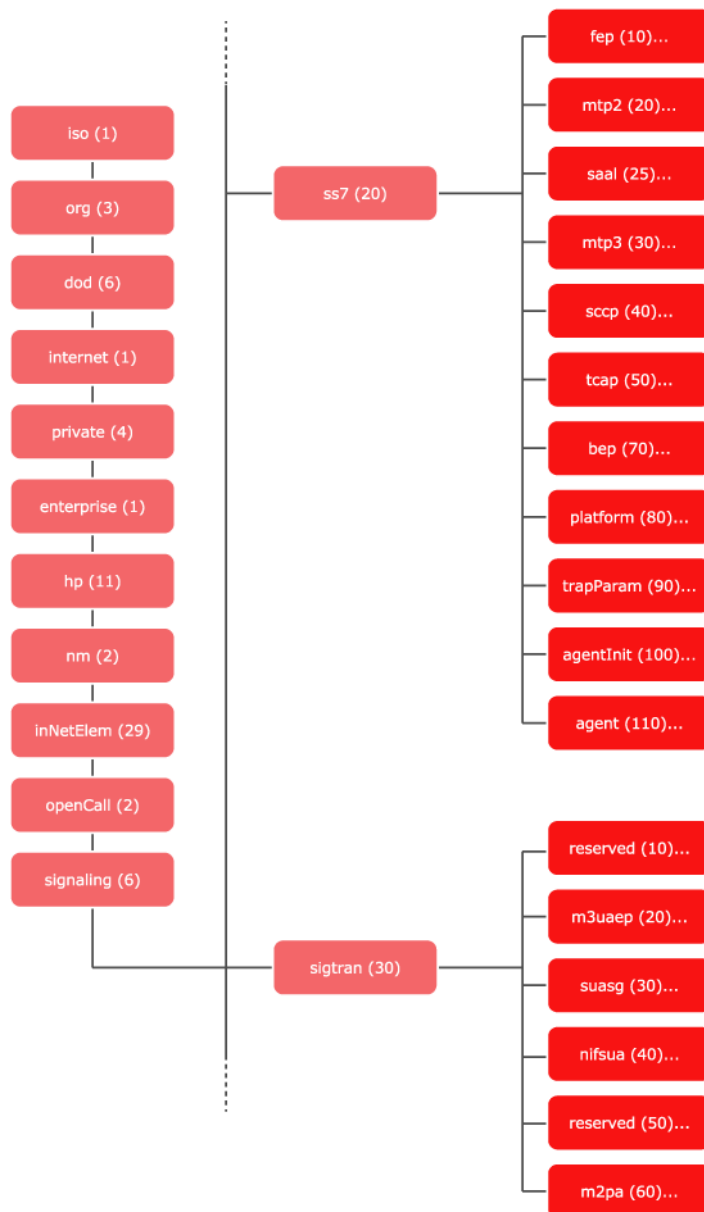
There is no inheritance between MIB objects. Not all ASN.1 syntax types are permitted in MIB definitions and accessible objects can only have a restricted syntax type, whereas inaccessible objects are sequences or sets of accessible objects.

The SNMP agent enables a network management application, such as HP OpenView, to monitor platforms. SNMP itself runs on the transport user datagram protocol (UDP) and enables the manager to communicate with the SNMP agent and interrogate the MIB.

The objects in the MIB are mapped to the entities and specified entity attributes, so that there is a correlation between a MIB object and a constant. The MIB provides sets of tables to structure the information obtained through this mapping process and the SNMP agent translates `SNMP Get` or `GetNext` requests from the manager into `SHOW` directives for the Director. When the Director responds, the information is translated by the agent into an SNMP response. If the Director message is an event, it is translated into the corresponding SNMP trap.

The next two figures show generic platform information and agent-specific information.

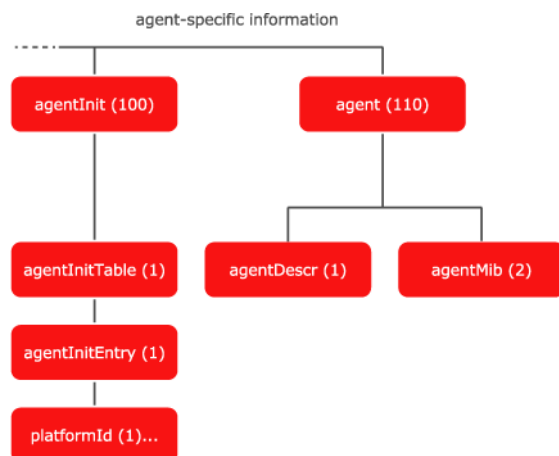
Fig 6: Generic information in the MIB

**Note**

These entities are fully described in the `~ss7/lib/HPOC-STG-<entity name>.mib` files.

`tcap (50)` and `bep (70)` are only applicable if you are using Bee-CUSP CSP Front End with the Bee-CUSP CSP back-end product.

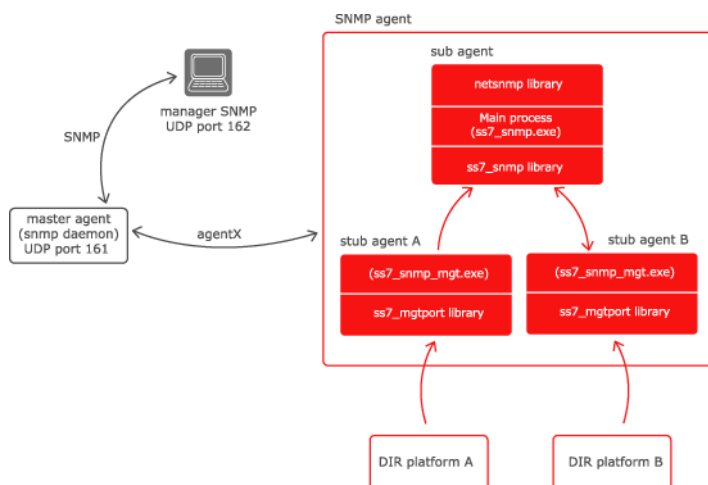
Fig 7: Agent-specific information in the MIB



3.1.1 SNMP overall architecture

The figure below shows the relationship between the SNMP agent, the SNMP subagent, and the SNMP stubs.

Fig 8: SNMP overall architecture



The **SNMP subagent** is the main process. There is a single SNMP subagent and one or more **stub processes** running on the server. The SNMP subagent starts a stub process for each managed platform.

A stub process is a relay between the SNMP subagent and a specific DIR. Upon reception of an SNMP request, the SNMP subagent sends a directive to the relevant stub, which forwards it to the corresponding DIR. The stub process sends back responses, events or PSIs received from the DIR to the SNMP subagent, which translates it into SNMP responses or traps.

The `ss7_snmp` library, linked to the SNMP subagent, provides a communication mechanism with the stub processes, management functions (show, associate, enable, and so on) and other facilities (timer handling, MIB translation, and so on). The SNMP subagent runs as user `root` while each stub process runs as user `ss7`.

When the SNMP subagent is started, it reads the `~ss7/lib/HPOC-STG-SS7.mib` file to map SS7 entities with SNMP objects and register the MIB structure to the `net-snmp` library.

3.1.2 Platform functionality

The SNMP agent can monitor one Bee-CUSP CSP Front End platform. One part of the SNMP agent, the association manager, starts a new process (stub) to allow the agent to be associated with a running DIR process that has been started in the Bee-CUSP CSP Front End platform that you want to monitor through SNMP. This implies that the DIR was started with the `ss7startup` command.

3.1.3 Implementation

The SNMP agent used by Bee-CUSP CSP Front End is based on the UCD (University College Davis) SNMP agent supplied with the Linux and Proliant support package (PSP). The architecture of the SNMP agent requires a master agent process and a subagent process that communicate through the distributed protocol interface (DPI), version 2 (RFC 1592). The master agent manages MIB II. The subagent handles a specific MIB that represents a part of the overall MIB. Bee-CUSP CSP Front End provides one subagent and one specific MIB. is based on the UCD (University College Davis) SNMP agent supplied with Linux and Proliant Support Package (PSP). The architecture of the SNMP Agent requires a Master Agent process and a subagent process that communicate through the Distributed Protocol Interface (DPI), version 2 (RFC 1592). The Master Agent manages MIB II. The subagent handles a specific MIB that represents a part of the overall MIB. Bee-CUSP CSP Front End provides one subagent and one specific MIB.

The table below details the initial hierarchy of the MIB.

If you are using your Bee-CUSP CSP Front End with the Bee-CUSP CSP back-end product, two initial hierarchies are supported: the **default** hierarchy, which applies to OpenCall platforms, and an alternate hierarchy, which applies to DEC platforms.

To swap from the default hierarchy to the alternate hierarchy, you need to set an environment variable when starting the SNMP agent.

Table 4: Initial hierarchy of the MIB

Default initial hierarchy	Alternate initial hierarchy (for DEC platforms used with the Bee-CUSP CSP product)
iso (1)	Information only
org (3)	org (3)
dod (6)	dod (6)
internet (1)	internet (1)
private (4)	private (4)
enterprise (1)	enterprise (1)
hp (11)	dec (36)
nm (2)	ema (2)
inNetElem (29)	mib-extension (18)
openCall (2)	in (43)
signaling (6)	decss7 (1)
ss7 (20)	

The UCD master agent runs as a daemon called `snmpd`, which must be started automatically by Linux. For more information on the Linux UCD master agent, see the man pages for `snmpd` and `snmpd.conf`. The subagent is a UCD layer that talks to `snmpd`.

Note



In the case of mixed Linux/Tru64 configurations (applicable to Bee-CUSP CSP only), the SNMP agent **must** be started on a Linux machine. The agent will then handle requests for Tru64 machines from the Linux machine.

3.1.4 The MIB

3.1.4.1 MIB objects

The Bee-CUSP CSP Front End MIB is built to monitor one or more Bee-CUSP CSP Front End platforms. Only the attributes needed for efficient monitoring of a platform or platforms from a local and network point of view are extracted from the Bee-CUSP CSP Front End entities.

In addition, some inaccessible MIB objects that match Bee-CUSP CSP Front End entities are added. These managed objects are:

- `fep`, which corresponds to the Bee-CUSP CSP Front End FEP_SUBSYSTEM entity for managing the front-end process of Bee-CUSP CSP.
- `mtp2`, which corresponds to the Bee-CUSP CSP Front End MTP2 entity for managing the MTP2 level of the SS7 stack
- `saal`, which corresponds to the Bee-CUSP CSP Front End SAAL entity for managing the SAAL level of the SS7 stack
- `mtp3`, which corresponds to the Bee-CUSP CSP Front End MTP3 entity for managing the MTP3 level of the SS7 stack
- `sccp`, which corresponds to the Bee-CUSP CSP Front End SCCP entity for managing the SCCP level of the SS7 stack
- `m2pa`, which corresponds to the Bee-CUSP CSP Front End M2PA entity for managing the M2PA level of the SIGTRAN stack
- `m3uaep`, which corresponds to the Bee-CUSP CSP Front End M3UAEP entity for managing the M3UA level of the SIGTRAN stack
- `suasg`, which corresponds to the Bee-CUSP CSP Front End SUASG entity for managing the SUA level of the SIGTRAN stack
- `nifsua`, which corresponds to the Bee-CUSP CSP Front End NIFSUA entity for managing the interface layer between the SS7 and SIGTRAN stacks
- `tcap*`, which corresponds to the TCAP_FEP entity for managing the distribution part of the SS7 TCAP level in an Bee-CUSP CSP Front End platform
- `bep*`, which corresponds to the BEP_SUBSYSTEM entity for managing the back-end process of Bee-CUSP CSP.

The next level of containment objects has no direct correlation with Bee-CUSP CSP Front End entities, but exists to implement the normal structure of the MIB. These are the table objects, one table for each Bee-CUSP CSP Front End entity that is mapped in the MIB. To allow for network element management, the managed objects listed in the table below have been added. These are mapped onto Bee-CUSP CSP Front End entities.

Table 5: Managed objects / entity mapping

MIB object	Mapped to Bee-CUSP CSP Front End entity
fepEntry	FEP_SUBSYSTEM
mtp2Entry	MTP2
saalEntry	SAAL
hsslEntry	HSSL
atrEntry	ATM_TRUNK
mtp3Entry	MTP3
sccpEntry	SCCP
applicationTable*	APPLICATION
dtlkEntry	DATA_LINK
trunkEntry	TRUNK
destinationEntry	DESTINATION
routeEntry	ROUTE
linksetEntry	LINKSET
linkEntry	LINK
m2paEntry	M2PA
m3uaepEntry	M3EP
suasgEntry	SUA
nifsua	NIFSUA
tcapTable*	TCAP
bepTable*	BEP
mtpbepTable*	MTP_BEP
sccpbepTable*	SCCP_BEP
tcapbepTable*	TCAP_BEP

All MIB objects marked with an asterisk (*) are only available if you are using your Bee-CUSP CSP Front End with the Bee-CUSP CSP back-end product.

There are additional MIB objects that are not associated with the Bee-CUSP CSP Front End entities, but required for managing Bee-CUSP CSP Front End platforms. These are the following:

- `platform` to provide the MIB structure for managing individually identified Bee-CUSP CSP Front End platforms, with its subordinate objects:
 - `platformTable`
 - `platformEntry`
- `trapParam` for managing SNMP traps
- `agentInit` to provide the MIB structure for initializing and re-initializing the Bee-CUSP CSP Front End subagent for a given Bee-CUSP CSP Front End platform, with its subordinate objects:
 - `agentInitTable`
 - `agentInitEntry`

- agent to provide Bee-CUSP CSP Front End SNMP agent-specific information and version number, with its subordinate objects:
 - agentDescr
 - agentMib

3.1.4.2 Registration tree objects

The remaining objects in the Bee-CUSP CSP Front End MIB are indexed on a specific object ID, or on a specific series of object IDs. The SNMP commands `Get` and `GetNext` operate on these objects, most of which represent a particular Bee-CUSP CSP Front End entity attribute. Each accessible object in the registration tree can be thought of as a leaf. The leaves of the registration tree structured as part of the MIB contain the Bee-CUSP CSP Front End MIB mapped onto Bee-CUSP CSP Front End management information. The object identifier (OID) that represents a specific leaf must be indexed on the `platformId` and appropriate additional index objects, depending on the information to be accessed. See the *Entity reference guide* for details of the Bee-CUSP CSP Front End MIB registration tree.

The following sections provide details of the indexed objects. Each chapter contains tables listing a set of objects that are dependent on a specific inaccessible object, and also lists their corresponding table and entry objects from the containment tree. The object ID or IDs against which each object is indexed are given in each table.

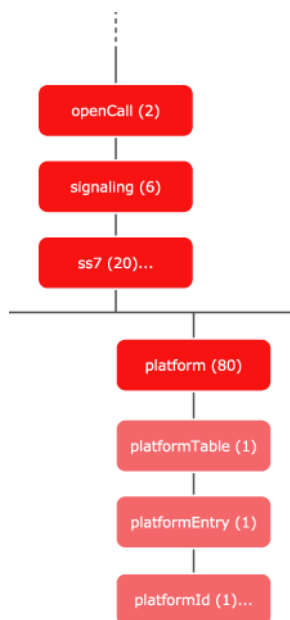
3.1.4.3 MIB structure

This section details the MIB structure of each Bee-CUSP CSP Front End entity.

MIB structure (platform)

The following figure shows the indexed objects in the MIB structure which are dependent on the `platform` inaccessible object.

Fig 9: Platform containment tree



Note

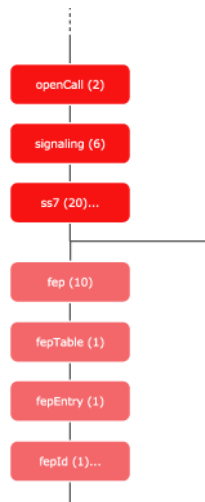


This entity is fully described in the `~ss7/lib/HPOC-STG-SS7.mib` file.

MIB structure (fep)

The following figure shows the indexed objects in the MIB structure which are dependent on the `fep` inaccessible object.

Fig 10: fep containment tree

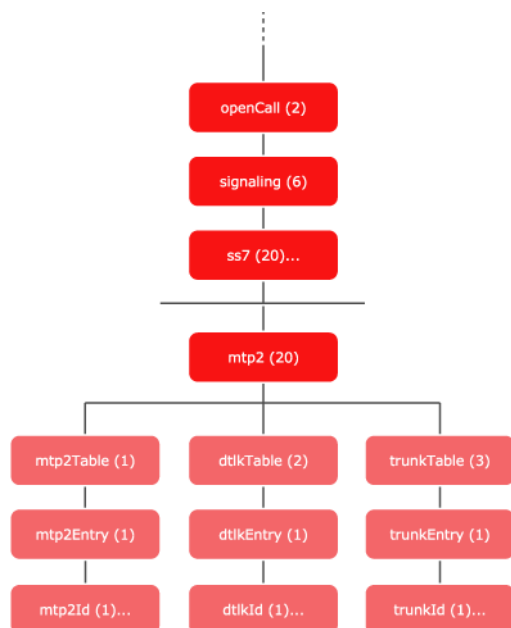
**Note**

This entity is fully described in the `~ss7/lib/HPOC-STG-SS7.mib` file.

MIB structure (mtp2)

The following figure shows the indexed objects in the MIB structure dependent on the `mtp2` inaccessible object.

Fig 11: mtp2 containment tree



Note

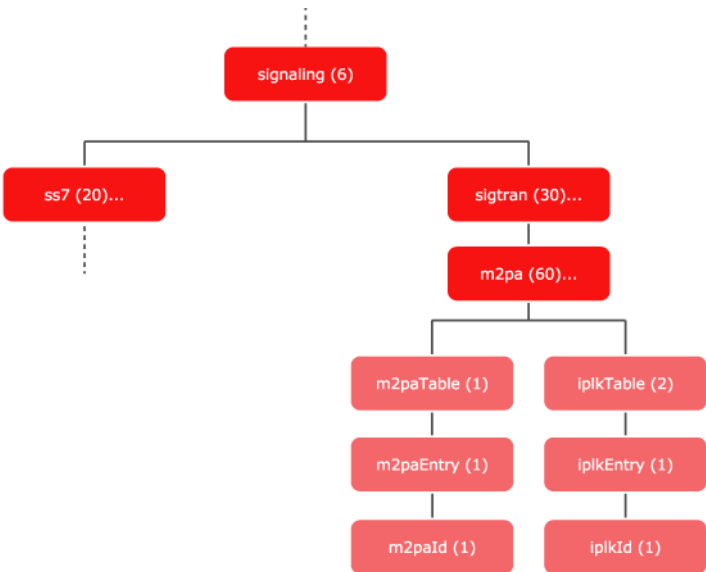


This entity is fully described in the `~ss7/lib/HPOC-STG-SS7.mib` file.

MIB structure (m2pa)

The following figure shows the indexed objects in the MIB structure dependent on the `m2pa` inaccessible object.

Fig 12: m2pa containment tree



Note

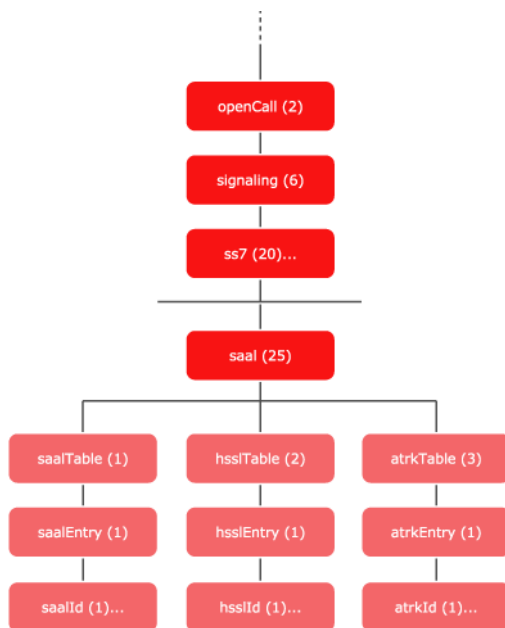


This entity is fully described in the `~ss7/lib/HPOC-STG-M2PA.mib` file.

MIB structure (saal)

The following figure shows the indexed objects in the MIB structure dependent on the `saal` inaccessible object.

Fig 13: saal containment tree

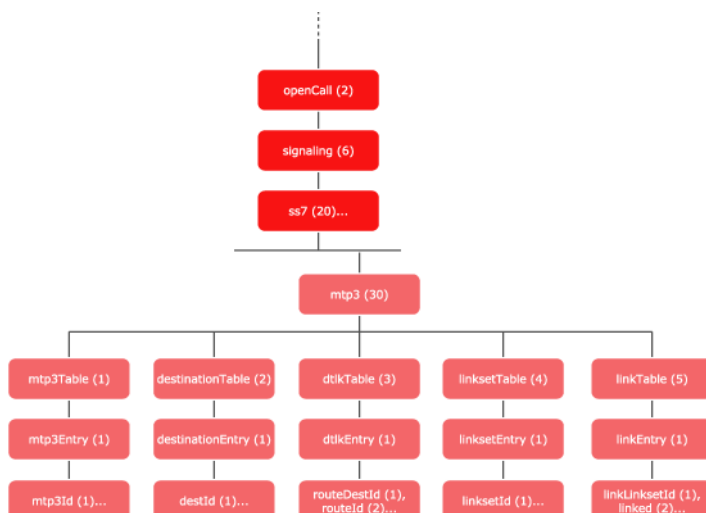
**Note**

This entity is fully described in the `~ss7/lib/HPOC-STG-SS7.mib` file.

MIB structure (mtp3)

The following figure shows the indexed objects in the MIB structure dependent on the mtp3 inaccessible object.

Fig 14: mtp3 containment tree



Note

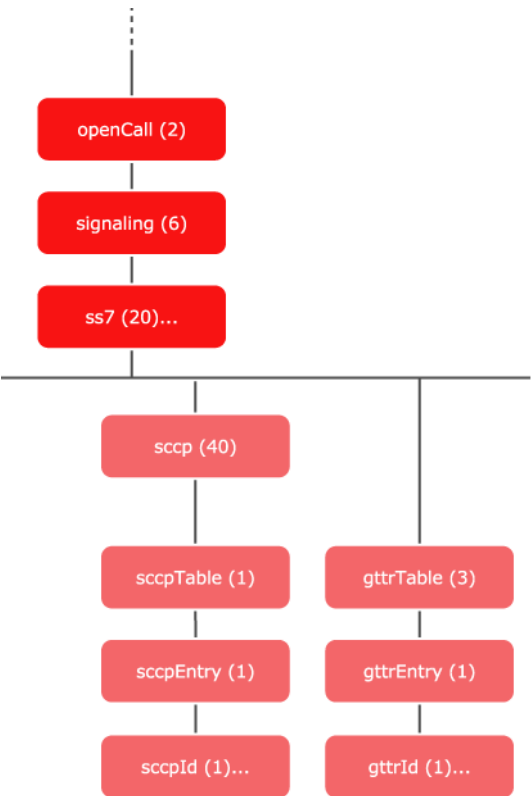


This entity is fully described in the `~ss7/lib/HPOC-STG-SS7.mib` file.

MIB structure (sccp)

The following figure shows the indexed objects in the MIB structure dependent on the `sccp` inaccessible object.

Fig 15: sccp containment tree



Note



This entity is fully described in the `~ss7/lib/HPOC-STG-SS7.mib` file.

MIB structure (sua)

The following figure shows the indexed objects in the MIB structure dependent on the `sua` inaccessible object.

Fig 16: sua containment tree

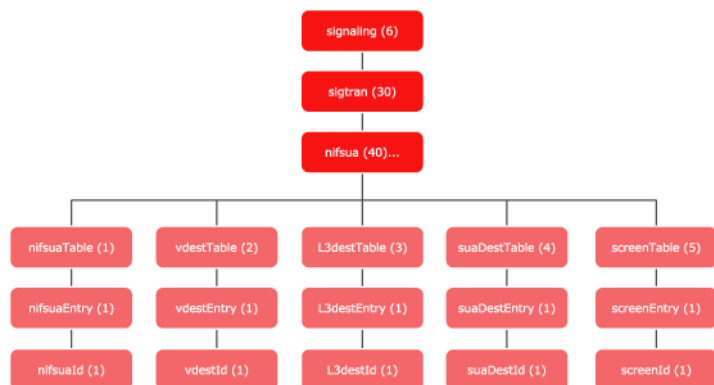
**Note**

This entity is fully described in the `~ss7/lib/HPOC-STG-SUASG.mib` file.

MIB structure (nifsua)

The following figure shows the indexed objects in the MIB structure dependent on the `nifsua` inaccessible object.

Fig 17: nifsua containment tree

**Note**

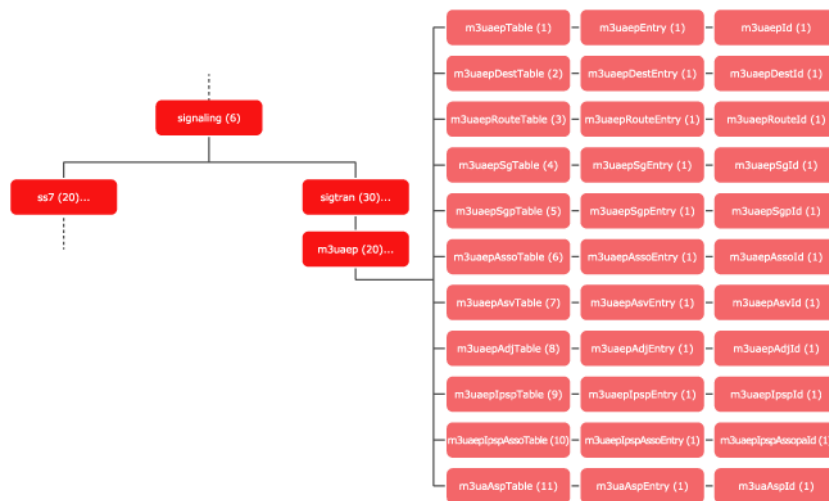
This entity is fully described in the `~ss7/lib/HPOC-STG-NIFSUA.mib` file.

MIB structure (m3ep)

The following figure shows the indexed objects in the MIB structure dependent on the `m3ep` inaccessible object.

Bee-CUSP Convergent Signaling Platform operating guide

Fig 18: m3ep containment tree



Note

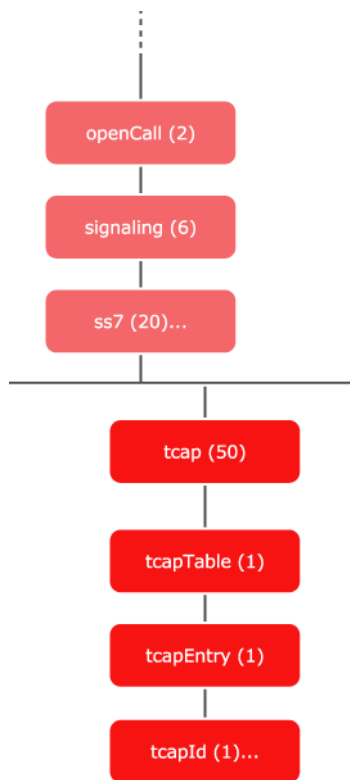
This entity is fully described in the `~ss7/lib/HPOC-STG-M3UAEP.mib` file.

MIB Structure (tcap)

This section describes the indexed objects in the MIB structure that are dependent on the inaccessible `tcap` object.

The `tcap` object is only available if you are using your Bee-CUSP CSP Front End with the Bee-CUSP CSP back-end product.

Fig 19: tcap containment tree

**Note**

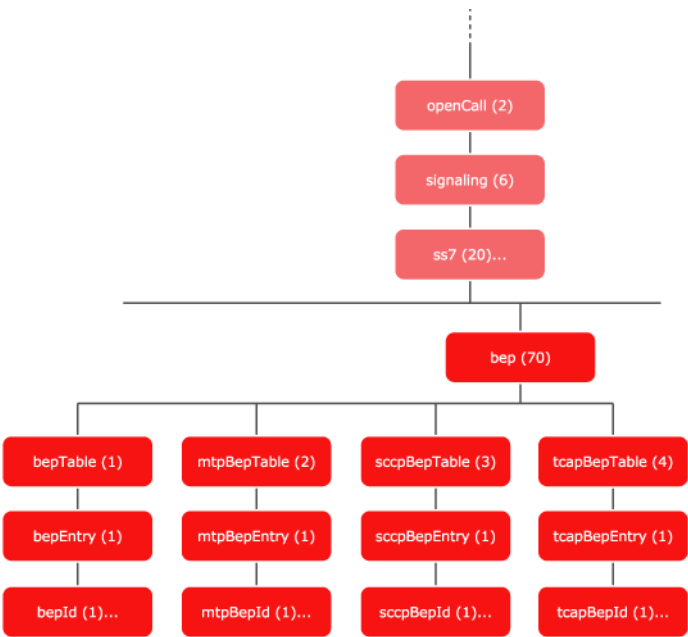
This entity is fully described in the `~ss7/lib/HPOC-STG-SS7.mib` file.

MIB Structure (bep)

This section describes the indexed objects in the MIB structure that are dependent on the inaccessible SNMP `bep` object.

The `bep` object is only available if you are using your Bee-CUSP CSP Front End with the Bee-CUSP CSP back-end product.

Fig 20: bep containment tree



Note

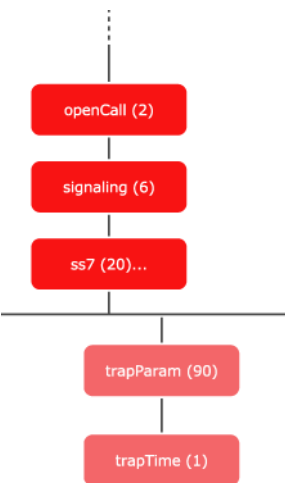


This entity is fully described in the `~ss7/lib/HPOC-STG-SS7.mib` file.

MIB structure (trapParam)

The following figure shows the indexed objects in the MIB structure dependent on the `trapParam` inaccessible object.

Fig 21: trapParam containment tree



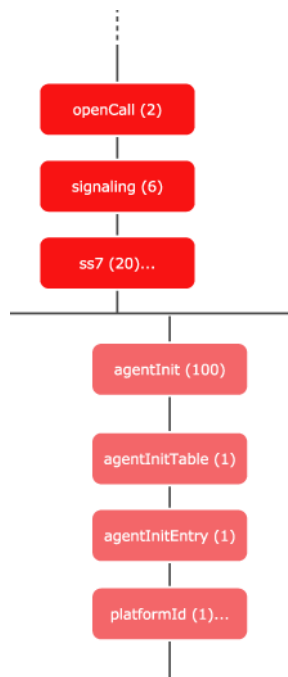
Note

This entity is fully described in the `~ss7/lib/HPOC-STG-SS7.mib` file.

MIB structure (agentInit)

The following figure shows the indexed objects in the MIB structure dependent on the `agentInit` inaccessible object.

Fig 22: agentInit containment tree

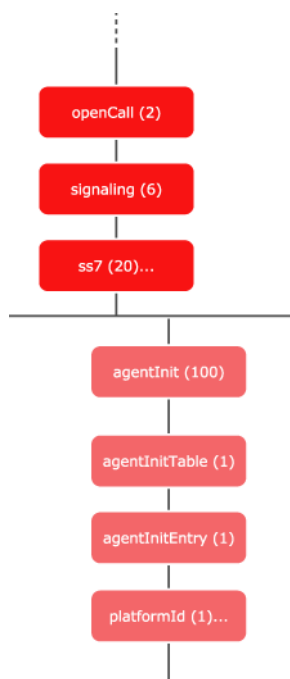
**Note**

This entity is fully described in the `~ss7/lib/HPOC-STG-SS7.mib` file.

MIB structure (agent)

The following figure shows the objects in the MIB structure dependent on the `agent` inaccessible object.

Fig 23: agent containment tree



Note



This entity is fully described in the `~ss7/lib/HPOC-STG-SS7.mib` file.

Chapter 4. Using the SNMP agent

This chapter provides information on how to configure, start and stop the SNMP agent. It also details how to use the SNMP agent to monitor a Bee-CUSP CSP platform.

4.1 Configuring the SNMP master agent

You need to configure the `snmpd` daemon to:

- Allow automatic restart at boot time
- Turn on AgentX support and allow the reporting of traps to a specific manager

You can then configure the set of traps that are reported.

4.1.1 Configuring an automatic restart

You must ensure that the `snmpd` daemon is automatically started on the server that runs the Bee-CUSP CSP SNMP agent.

You must also ensure that the `snmpd` daemon is restarted at reboot time. If it is not running, start it manually by running the following command as user `root`:

```
/etc/init.d/snmpd start
```

4.1.2 Configuring the `snmpd.conf` file

Configure the NetSNMP master agent in the `/etc/snmp/snmpd.conf` file to:

- Enable AgentX support (step 1.1)
- Define the IP-STG MIB view
- Configure the IP address of the SNMP managers that will receive the SNMP traps (step 1.3).

The `/etc/snmp/snmpd.conf` file is automatically filled at kit installation with AgentX support enabled and with definition of IP-STG MIB view .

Procedure

1. Edit the `/etc/snmp/snmpd.conf` file as follows (modify as user `root`):

1. Add the command `master agentx` to turn on AgentX master agent support.

2. Add the IP-STG MIB view:

```
view systemview .1.3.6.1.4.1.11.2.29.2.6 3.
```

3. The Bee-CUSP CSP SNMP agent uses the `public` community for reporting traps. You must identify which server will receive the SNMP traps to ensure that the Bee-CUSP CSP SNMP agent reports to a given IP address.

For example, the following command causes all traps reported by the Bee-CUSP CSP SNMP agent to be reported to IP address 16.36.5.22:

Bee-CUSP Convergent Signaling Platform operating guide

```
trap2sink 16.36.5.22 public
```

Note



If the trap receiver server is the server hosting Bee-CUSP CSP (i.e. the trap receiver is started on the Bee-CUSP CSP server) you need to add the following line in the `/etc/hosts.allow` file:

```
snmptrapd: ALL
```

Note



On the trap receiver server, add the following line in the `/etc/snmp/snmptrapd.conf` file :

```
authCommunity log,execute,net public
```

For more information on the notification receiver configuration file, see the man page of `snmptrapd.conf`.

2. Restart the snmpd daemon: `/etc/init.d/snmpd restart`

3. Run the `ss7startup` command for the platform ID that the SNMP agent will monitor:

```
ss7startup [<machine>] <platform_Id> snmp
```

Example where platform id = d:

```
ss7startup d snmp
```

See **Starting the SNMP agent** for details of using the `ss7startup` command to start Bee-CUSP CSP SNMP agent processes.

Example of snmpd.conf file

```
[root@dingo snmp]# grep -v "#" /etc/snmp/snmpd.conf
dlmod cmaX /usr/lib/libcmaX.so
rwcommunity private 127.0.0.1
rocommunity public 127.0.0.1
syscontact Root <root@localhost> (configure /etc/snmp/snmp.local.conf)
syslocation Unknown (edit /etc/snmp/snmpd.conf)
com2sec notConfigUser default public
com2sec6 notConfigUser default public
group notConfigGroup v1 notConfigUser
group notConfigGroup v2c notConfigUser
view systemview included .1.3.6.1.2.1.1
view systemview included .1.3.6.1.2.1.25.1.1
access notConfigGroup "" any noauth exact systemview none none
pass .1.3.6.1.4.1.4413.4.1 /usr/bin/ucd5820stat
master agentx
view systemview included .1.3.6.1.4.1.11.2.29.2.6
```

Notes on secured SNMP configuration

Secured SNMP is the only SNMP configuration supported and does not need to be specified. The following lines are automatically added in `/etc/sudoers` at the time of kit installation:

```
# BEGIN-Privilege needed by IP-STG to use snmp agent
%ocadmin, ocadmin <hostname> = NOPASSWD: /usr/bin/ss7_snmp.exe *, /usr/var/ss7/
bin/ss7_snmp_stop.exe *
%ocadmin, ss7 <hostname> = NOPASSWD: /usr/bin/ss7_snmp.exe *, /usr/var/ss7/bin/
ss7_snmp_stop.exe *
# END-Privilege
```

Once these lines are added, the SS7 SNMP agent will automatically be run with `sudo`. Using `sudo` before running any executable ensures that security checks are performed.

4.1.3 Configuring the set of traps that are reported

You can configure the Bee-CUSP CSP SNMP agent dynamically to enable or disable SNMP traps generated for an internal reason or because the agent receives an Bee-CUSP CSP event. This is done by loading the trap filtering file `ss7_trap_filtering.dat`. (The trap filtering file is delivered in the kit with all traps set to on.) You do not need to stop the Bee-CUSP CSP SNMP agent in order to load the file.

The pathname of the trap filtering file is:

```
/usr/var/ss7/ss7_trap_filtering.dat
```

This file is on read-only but under account `ss7`, it is possible to enable writing to modify it. To configure or reconfigure the set of traps reported by the Bee-CUSP CSP SNMP agent:

1. Modify the trap filtering file, `ss7_trap_filtering.dat`, as required. The mapping between the `trapid` and the trap received by the `trapreceiver` is as follows.
 - a. To disable a trap, you first need to check the trap received in the `trapreceiver` log and make a note of the text describing the trap.
 - b. Run a search for this text in the file `~ss7/lib/decss7_mib.profile`. The text must be assigned to the cause attribute of an `EVENT REPORT` structure.
 - c. In the file `ss7_trap_filtering.dat` set the value of the `trap_id` attribute in this structure to `OFF`.

2. Restart the Bee-CUSP CSP SNMP agent:

```
ss7startup [<machine>] <platform_Id> snmp
```

or, if the Bee-CUSP CSP SNMP agent is already running, reload the file using the `AgentInit` object from the Bee-CUSP CSP MIB:

```
/usr/bin/snmpset -v 2c -m /usr/var/ss7/lib/decss7.mib -t 20 -r 0 -c public
localhost .1.3.6.1.4.1.11.2.29.2.6.20.100.1.1.2.1 i 2
```

where `<IndexOfPlatform>` is a number (`a= 1, b=2,... z=26`), the SNMP platform index that corresponds to the Bee-CUSP CSP platform ID.

The internal trap numbers shown in the first column of the file *must not be modified*. The comment in the third column of the file indicates the name of the trap. An example of the trap filtering file template follows, with all traps set to be reported (value = ON).

```
//*****
//
//          **** COPYRIGHT NOTICE ****
//
```

Bee-CUSP Convergent Signaling Platform operating guide

```
//
// Copyright (c) Digital Equipment Corporation, 1992
//
// All Rights Reserved. Unpublished rights reserved under the
// copyright laws of the United States.
//
// The software contained on this media is proprietary to and
// embodies the confidential technology of Digital Equipment
// Corporation. Possession, use, duplication or dissemination
// of the software and media is authorized only pursuant to a
// valid written license from Digital Equipment Corporation.
//
//
// The information in this software is subject to change without notice
// and should not be construed as a commitment by DIGITAL EQUIPMENT
// CORPORATION.
//
// DIGITAL assumes no responsibility for the use or reliability of its
// software on equipment which is not supplied by DIGITAL.
//
//
//*****
//*****
//
//
// FACILITY:          Extensible SNMP agent for DECss7/Isup
//
// ENVIRONMENT:       LINUX
//
// MODULE NAME:       ss7_trap_filtering.dat
//
// DESCRIPTION:
//
// AUTHORS:
//
// CREATION DATE:    22/03/1999
//
//*****
// Internal Traps
<1>      <ON> // agentInitStarted
<2>      <ON> // agentInitSuccessful
<3>      <ON> // agentStopped
<4>      <ON> // agentAssoFailed
<5>      <ON> // agentDisassoFailed
<6>      <ON> // agentTrapFailed
<7>      <ON> // agentTrapSuccessful
<8>      <ON> // agentPlatformInconsistency

// PSI Traps
<103>    <ON> // fepProcessDown
<104>    <ON> // fepProcessUp
<105>    <ON> // dirProcessDown
<106>    <ON> // dirProcessUp
<107>    <ON> // appProcessDown
<108>    <ON> // appProcessUp
<110>    <ON> // fepProcessConnected
<111>    <ON> // levelAppOperational
```



```

<112>    <ON> // levelAppNonOperational
<113>    <ON> // levelAppDown
<114>    <ON> // levelAppUp
<115>    <ON> // acfServerDown
<116>    <ON> // acfServerUp

// SCCP Traps
<201>    <ON> // nstateIndUserInServ
<202>    <ON> // nstateIndUserOutOfServ

// Event Traps
<301>    <ON> // fepCongestionOn
<302>    <ON> // fepCongestionOff
<303>    <ON> // fepStorageError
<304>    <ON> // fepLogFileFull
<305>    <ON> // fepConnectLostWithSubsys
<306>    <ON> // fepAssoLostWithAppICopy
<307>    <ON> // fepCongestionHigh

<501>    <ON> // dtlkLinkFailAllReasons
<502>    <ON> // dtlkExcessDelayAck
<503>    <ON> // dtlkExcessErrorRate
<504>    <ON> // dtlkExcessRemoteCong
<505>    <ON> // dtlkCongestionStart
<506>    <ON> // dtlkInitialAlignFailure
<507>    <ON> // dtlkStartRemProcOutage
<508>    <ON> // dtlkEndRemProcOutage
<509>    <ON> // dtlkLinkInService
<510>    <ON> // dtlkCongestionEnd
<511>    <ON> // dtlkCongestionDiscard
<512>    <ON> // dtlkDriverInfo
<513>    <ON> // dtlkLoadRcvIn1
<514>    <ON> // dtlkLoadRcvOut1
<515>    <ON> // dtlkLoadRcvIn2
<516>    <ON> // dtlkLoadRcvOut2
<517>    <ON> // dtlkLoadRcvIn3
<518>    <ON> // dtlkLoadRcvOut3
<519>    <ON> // dtlkLoadRcvIn4
<520>    <ON> // dtlkLoadRcvOut4
<521>    <ON> // dtlkLoadTrnIn1
<522>    <ON> // dtlkLoadTrnOut1
<523>    <ON> // dtlkLoadTrnIn2
<524>    <ON> // dtlkLoadTrnOut2
<525>    <ON> // dtlkLoadTrnIn3
<526>    <ON> // dtlkLoadTrnOut3
<527>    <ON> // dtlkLoadTrnIn4
<528>    <ON> // dtlkLoadTrnOut4
<601>    <ON> // trunkConnectionLost
<602>    <ON> // trunkCriticalAlarm
<603>    <ON> // trunkConnectionRecovered
<604>    <ON> // trunkCriticalAlarmOff
<605>    <ON> // trunkLFrmer
<606>    <ON> // trunkRdirCriticalAlarm
<607>    <ON> // trunkRdirCriticalAlarmOff

<2401>   <ON> // saalSoftwareError
<2402>   <ON> // saalLevelFail

```

Bee-CUSP Convergent Signaling Platform operating guide

```
<2501>    <ON> // hsslAlignmentEV
<2502>    <ON> // hsslInService
<2503>    <ON> // hsslOutOfService
<2504>    <ON> // hsslCongDetected
<2505>    <ON> // hsslCongCeased
<2506>    <ON> // hsslSscopRecovered
<2507>    <ON> // hsslPduTransmitted
<2508>    <ON> // hsslUnitDataRcv
<2509>    <ON> // hsslSscopRelease
<2510>    <ON> // hsslRemRelease
<2511>    <ON> // hsslLocalRelease
<2512>    <ON> // hsslStartProving
<2513>    <ON> // hsslStopProving
<2514>    <ON> // hsslUnsolicitedPdu
<2515>    <ON> // hsslUnsuccRetransEV
<2516>    <ON> // hsslOtherError
<2517>    <ON> // hsslSdLossEV
<2518>    <ON> // hsslCreditCondEV
<2519>    <ON> // hsslSscopUnitData
<2520>    <ON> // hsslLpOutage
<2521>    <ON> // hsslLpRecovered

<2601>    <ON> // atrkTkCntLoses
<2602>    <ON> // atrkTkCntRecovered
<2603>    <ON> // atrkCriticalAlarm
<2604>    <ON> // atrkClearedAlarm

<701>     <ON> // mtp3OrigMsdDiscardInVpC
<702>     <ON> // mtp3NoMoreCongestion
<703>     <ON> // mtp3CongestionStarted
<704>     <ON> // mtp3TfmTcmRejected

<801>     <ON> // destStartOfInaccess
<802>     <ON> // destCongested
<803>     <ON> // destEndOfInaccess
<804>     <ON> // destNoMoreCongestion
<805>     <ON> // destMsdDiscardedInaccPc
<806>     <ON> // destTempoDestCreated
<807>     <ON> // destTempoDestDeleted
<808>     <ON> // destRuleModif

<901>     <ON> // routeRestricted
<902>     <ON> // routeProhibited
<903>     <ON> // routeAllowed
<904>     <ON> // routeDisabled

<1001>    <ON> // linksetStartOfUnavail
<1002>    <ON> // linksetEndOfUnavail
<1003>    <ON> // linksetRuleModif

<1101>    <ON> // linkMsgDiscarded
<1102>    <ON> // linkT2TimeOut
<1103>    <ON> // linkT5TimeOut
<1104>    <ON> // linkSlInhibited
<1105>    <ON> // linkSlUninhibited
<1106>    <ON> // linkInhibitDenied
```

```

<1107>    <ON> // linkUninhibitNotPossible
<1108>    <ON> // linkLocalAutoChangeOver
<1109>    <ON> // linkSlRestoration
<1110>    <ON> // linkLfuSent
<1111>    <ON> // linkLfuReceived
<1112>    <ON> // linkSltSuccess
<1113>    <ON> // linkSltFailed
<1114>    <ON> // linkInhibReqTimeOut
<1115>    <ON> // linkUninhibReqTimeOut
<1116>    <ON> // linkRemLinkInhibited
<1117>    <ON> // linkRemLinkUninhibited
<1118>    <ON> // linkCongestedInTb
<1119>    <ON> // linkNotCongestedInTb
<1120>    <ON> // linkUpuSent
<1121>    <ON> // linkUpuReceived

<1201>    <ON> // sccpRoutingFailure
<1202>    <ON> // sccpSyntaxErrors
<1203>    <ON> // sccpGttFailure
<1204>    <ON> // sccpMgmtReqDiscarded
<1205>    <ON> // sccpSegmentationError
<1206>    <ON> // sccpReassemblyError
<1207>    <ON> // sccpRemoteProblem
<1208>    <ON> // sccpTcapErr

<2201>    <ON> // inapSrfFailure
<2202>    <ON> // inapSrfTsrftimeout
<2203>    <ON> // inapSrfErrorCancel
<2204>    <ON> // inapSrfErrorOpRcv
<2205>    <ON> // inapSrfErrorOnARI

<2301>    <ON> // is41Failure
<2302>    <ON> // is41SoftwareError
<2303>    <ON> // is41CongestionOn
<2304>    <ON> // is41CongestionOff

<3101>    <ON> // m3uaLevelFail
<3102>    <ON> // m3uaCong
<3103>    <ON> // m3uaNoCong

<3201>    <ON> // m3uaSgpAcceptConn
<3202>    <ON> // m3uaSgpRefusedConn
<3203>    <ON> // m3uaSgpLevelFail
<3204>    <ON> // m3uaSgpActiveEv

<3401>    <ON> // m3uaAssociationNotEstablished
<3402>    <ON> // m3uaAssociationDownEv
<3403>    <ON> // m3uaAssociationUpEv
<3404>    <ON> // m3uaAssociationNoCong
<3405>    <ON> // m3uaAssociationCong
<3406>    <ON> // m3uaAssociationAsptmRcv
<3407>    <ON> // m3uaAssociationAsptmSnd
<3408>    <ON> // m3uaAssociationAspsmRcv
<3409>    <ON> // m3uaAssociationAspsmSnd
<3410>    <ON> // m3uaAssociationRkmRcv
<3411>    <ON> // m3uaAssociationRkmSnd
<3412>    <ON> // m3uaAssociationMgmtRcv

```

```
<3413>    <ON> // m3uaAssociationMgmtSnd
<3414>    <ON> // m3uaAssociationSsnmRcv
<3415>    <ON> // m3uaAssociationSsnmSnd

<3501>    <ON> // m3uaDestinationDownEv
<3502>    <ON> // m3uaDestinationPendingEv
<3503>    <ON> // m3uaDestinationInactEv
<3504>    <ON> // m3uaDestinationActEv
<3505>    <OFF> // m3uaDestinationRemoteCongEv (ANSI only)
<3506>    <OFF> // m3uaDestinationRemoteNoCongEv (ANSI only)
<3507>    <ON> // m3uaDestinationLocalCongEv
<3508>    <ON> // m3uaDestinationLocalNoCongEv

<3601>    <ON> // m3uaRouteDownEv
<3602>    <ON> // m3uaRouteInactEv
<3603>    <ON> // m3uaRouteActEv
<3701>    <ON> // nifScreenRuleMatch
```

4.2 Starting the SNMP agent

The Bee-CUSP CSP SNMP agent should only be started after the full management configuration has been applied to one of the Bee-CUSP CSP platforms that it is to monitor, and Bee-CUSP CSP processes have been started for this platform.

Run the following command for the platform ID that the SNMP agent is to monitor:

```
ss7startup [<machine>] <platform_Id> snmp
```

When the SNMP agent is started, it enables the events mapped to the traps specified in the MIB for all entities that have been created. If the SNMP agent is started before the platform has been fully configured, or if new entities are created after the SNMP agent is started, only inconsistencies will be detected in the new entities during subsequent monitoring. In this case, the SNMP agent should be restarted with the `ss7startup` command.



Caution

In the case of multi-platform configuration on a given server, all FE processes on all platforms must be configured and their configuration saved using the `S7MP SAVE_CONFIG` command before starting the SNMP agent (using the `ss7startup` command). If not, this could result in inconsistencies, related to the SNMP agent in the MIB.

4.3 Starting the snmptrapd daemon

In order to receive and log the SNMP trap messages reported by the SNMP agent, you must ensure that the `snmptrapd` daemon is automatically started on the server on which the SNMP agent runs.

You must also ensure that the `snmptrapd` daemon is restarted at reboot time. If it is not running, start it manually by running:

```
/etc/init.d/snmptrapd start
```

The MIB module to load must be added within the `OPTIONS` defined in the `/etc/snmp/snmptrapd.options` file:

```
OPTIONS="... -m /usr/var/ss7/lib/decss7.mib"
```

The traps are sent by default in the directory `/var/log/messages`. You

can redirect the SNMP traps to the file `snmp_trap.log` within the `OPTIONS` defined in the `/etc/snmp/snmptrapd.options` file:

```
OPTIONS="... -Lf /usr/var/ss7/platform_x/snmp_trap.log"
```

For more information on the `snmptrapd` daemon, see the man page for `snmptrapd`.

4.4 Monitoring a platform through the SNMP agent

You are responsible for synchronizing the running of all applications on platforms under your control. These applications include the SNMP agent.

The SNMP `Get` and `GetNext` commands can be sent using the Linux `snmpget` and `snmpgetnext` commands. These commands are used to query platform information through the SNMP agent. The SNMP agent reports asynchronous information through traps that can be displayed using the `snmptrapd` daemon. The `snmpget` and `snmpgetnext` commands and the use of the `snmptrapd` daemon are documented in the Linux man pages.

The SNMP `Get` or `GetNext` command is translated into an Bee-CUSP CSP `SHOW` directive for the application managing a specific platform. (These directives are documented under the `S7MP SHOW` command, with or without the `SHOWWILD` mode set to on, in the `S7MP` section of this document.

4.4.1 Querying the SNMP agent through `snmpget_requests`

The following examples show the use of the `snmpget` tool supplied with Linux for querying the SNMP agent. In these examples the SNMP agent runs on the server `padova` in platform `a`.

Examples

Sample `snmpget`

```
/usr/bin/snmpget -t 10 -v 2c -m /usr/var/ss7/lib/decss7.mib -c public localhost
.1.3.6.1.4.1.11.2.29.2.6.20.30.2.1.6.1.14
```

This SNMP `Get` request corresponds to the `S7MP` command:

```
SHOW FEP * MTP3 DESTINATION 14 /STATUS
```

Sample `snmpget`

```
/usr/bin/snmpget -t 10 -v 2c -m decss7.mib -m /usr/var/ss7/lib/decss7.mib -c
public localhost
.1.3.6.1.4.1.11.2.29.2.6.20.30.3.1.2.1.14.22
```

This SNMP `Get` request corresponds to the `S7MP` command:

```
SHOW FEP * MTP3 DESTINATION 14 ROUTE 22 /ID
```

where the following mapping applies to the trailing figures:

- 1 is the index for the Bee-CUSP CSP platform ID and corresponds to platform `a`. Platform IDs are indexed in ascending order: the lowest index number corresponds to the first platform ID in alphabetical order, so 1 identifies platform `a`, 2 identifies platform `b`, and so on.

- 14 represents the DESTINATION entity instance ID
- 22 represents the ROUTE entity instance ID

Where Bee-CUSP CSP requests can be distributed, i.e. where they target child entities of the FEP_SUBSYSTEM, the SNMP Get and GetNext commands are mapped to the corresponding Bee-CUSP CSP directives, making use of the global view feature. Therefore, the FEP_SUBSYSTEM ID is not explicitly named, but is replaced by the asterisk (*) character in the S7MP command. Consequently, the SNMP command does not index the FEP_SUBSYSTEM ID.

4.4.2 Interpreting the set of traps that are reported

This section describes the way traps are raised according to their definition in the MIB. Following is an example detailing the definition of the m3uaLevelFail trap type:

Example

```
m3uaLevelFail TRAP-TYPE
    ENTERPRISE m3uasg
    VARIABLES { trapTime, trapEntityName, trapCause,
                trapSeverity, trapType, trapText, trapPlatformId,
                m3uaPlatformId, trapSpecificProblem }
    DESCRIPTION "Error causes the program to stop"
    MANAGEDOBJECT-TABLE: m3uaTable
    SEVERITY-DEFAULT: Critical
    SPROBLEM-VARIABLE: trapSpecificProblem"
    ::= 3101
```

- The ENTERPRISE clause determines under which section of the MIB the trap is registered: m3uasg traps are located under (signaling.sigtran.)m3uasg. All other traps (including PSI, internal and SS7 traps) are located under (signaling.)ss7.
- The VARIABLES clause defines the sequence of MIB objects contained in each instance of the trap type.

Some MIB objects are common to all traps while others are specific to a given trap type. Common objects are prefixed with trap (for instance, trapTime, trapType). These MIB objects are defined in the trapParam MIB table, which is located under (signaling.)ss7. Remaining objects are trap-specific and are defined in their corresponding table. In the example above, m3uaPlatformId is defined in the m3uasgTable, located under (signaling.sigtran.)m3uasg.

- The DESCRIPTION clause provides a textual definition of the trap type.
- The trap value (3101 in the example above) is assigned to each trap type under the registration authority defined in the ENTERPRISE clause.

The snmpTrapOID, which is the concatenation of the ENTERPRISE OID and the trap value, is present in each trap instance.

Below is an example of the way the trap receiver will display an instance of the m3uaLevelFail trap type.

Example of m3uaLevelFail trap type output

```
Mar 11 11:23:44 polka snmptrapd[22145]: polka.vbe.cpqcorp.net [16.188.157.27]:
Trap SNMPv2-MIB::sysUpTime.0 =
SMI::enterprises.11.2.29.2.6.20.90.3 = STRING: "FEP_SUBSYSTEM
POLKA M3UA", SNMPv2-SMI::enterprises.11.2.29.2.6.20.90.6 =
STRING: "Error causes the program to stop", SNMPv2-
SMI::enterprises.11.2.29.2.6.20.90.7 = INTEGER: 8, SNMPv2-
SMI::enterprises.11.2.29.2.6.20.90.2 = INTEGER: 0, SNMPv2-
SMI::enterprises.11.2.29.2.6.20.90.99 = STRING: "Y", SNMPv2-
```

```
SMI::enterprises.11.2.29.2.6.30.10.1.1.99 = INTEGER: 25,
SNMPv2-SMI::enterprises.11.2.29.2.6.20.90.9 = INTEGER: 3101
```

The trap instance contains the following information:

- The `snmpTrapOID`:

```
SNMPv2-MIB::snmpTrapOID.0 = OID: SNMPv2-
SMI::enterprises.11.2.29.2.6.30.10.0.3101
```

- The MIB objects defined in the `VARIABLE` clause (both common and specific):

```
SNMPv2-SMI::enterprises.11.2.29.2.6.20.90.1 =
STRING: "11-Mar- 2005 11:23:44.00", SNMPv2-SMI::enterprises.11.2.29.2.6.20.90.3
=
STRING: "FEP_SUBSYSTEM POLKA M3UA", SNMPv2-
SMI::enterprises.11.2.29.2.6.20.90.6 =
STRING: "Error causes the program to stop",
SNMPv2-SMI::enterprises.11.2.29.2.6.20.90.7 =
INTEGER: 8, SNMPv2-SMI::enterprises.11.2.29.2.6.20.90.2 =
INTEGER: 0, SNMPv2-SMI::enterprises.11.2.29.2.6.20.90.99 =
STRING: "Y", SNMPv2-SMI::enterprises.11.2.29.2.6.30.10.1.1.99 =
INTEGER: 25, SNMPv2-SMI::enterprises.11.2.29.2.6.20.90.9 =
INTEGER: 3101
```

where the trap-specific object appears in bold, i.e.

```
SNMPv2-SMI::enterprises.11.2.29.2.6.30.10.1.1.99 = INTEGER: 25
```

Note



In this example, `snmpTrapOID` contains a next-to-last 'o' sub-identifier (between the enterprise OID and the trap value) which is automatically added by the `net-snmp` API to map SNMP v1 traps to SNMP v2 notifications.

4.5 Updating platforms monitored through the SNMP agent

When you update a platform monitored through the SNMP agent, you must re-initialize the agent by restarting it. The agent cannot detect when any aspect of the configuration is modified. It will only detect inconsistencies against known configurations during subsequent monitoring. To prevent inconsistencies, we recommend you restart the SNMP agent using the `ss7startup` command.

4.6 Stopping the SNMP agent

Use the following command to stop all instances of the Bee-CUSP CSP SNMP agent:

```
ss7shutdown snmp or ss7shutdown [<machine>] <platform_Id> snmp
```

Chapter 5. Using S7MP

5.1 Introduction

The SS7 management program (S7MP) is a utility used for configuring, controlling and monitoring an Bee-CUSP CSP platform. This section provides reference material relating to S7MP and describes the format and syntax of S7MP commands. You should use it in conjunction with the Bee-CUSP CSP operating guide and Bee-CUSP CSP entity reference guide to configure, control, and manage an Bee-CUSP CSP platform.

5.1.1 Defining the operating environment

You can setup your operating environment by defining certain environment variables in the C shell. S7MP can be enabled in verbose mode and batch files can be used to simplify platform configuration tasks.

5.1.1.1 Verbose mode

In verbose mode, all the output returned in response to S7MP commands is displayed. If verbose mode is not enabled, responses are only displayed in the case of errors.

Verbose mode can be turned ON or OFF with the **SET** command after S7MP has been started, for example:

```
SET MODE /VERBOSE=ON
```

```
SET MODE /VERBOSE=OFF
```

See the section on the **SET** command for more details.

5.1.1.2 Environment variables

Environment variables can be used in S7MP commands instead of names in command files or when commands are entered interactively. These must be defined *before* the platform is started and *before* the S7MP command file is run. For example, the following C shell command defines the environment variable FEP1 as the FE process subsystem (FEP_SUBSYSTEM) on a server called VENISE:

```
% setenv FEP1 VENISE
```

You can also use environment variables to provide attribute values (whatever the attribute qualifiers may be). The following C shell command defines the environment variable:

- SCP_OPC as the generic name for the OPC value:

```
% setenv SCP_OPC 1
```

- FEP1_TRUNK1 as the generic name for the trunk device PT38A0:

```
% setenv FEP1_TRUNK1 PT38A0
```

- FEP1_LINK1 as the generic name for the link device PT38A001:

```
% setenv FEP1_LINK1 PT38A001
```

- STP_A as the generic name for destination 10:

```
% setenv STP_A 10
```


- ROUTE_A as the generic name for route 3:

```
% setenv ROUTE_A 3
```

- You can also use environment variables when creating FEP sub-entities, as follows:

```
S7MP_C>CREATE FEP FEP1 /OPC=SCP_OPC
```

```
S7MP_C>CREATE FEP FEP1 MTP2 TRUNK 1/DEVNAM=FEP1_TRUNK1/FEPNAM=FEP1
```

```
S7MP_C>CREATE FEP FEP1 MTP2 DATA_LINK 1/DEVNAM=FEP1_LINK1/FEPNAM=FEP1
```

```
S7MP_C>CREATE FEP FEP1 MTP3 DESTINATION STP_A ROUTE ROUTE_A /RANK=0
```

5.2 Starting S7MP

Start S7MP from the SS7 account.

S7MP needs a `platform_id` parameter to be launched. The value for `<platform_id>` is a user-defined value entered during the transport protocol configuration procedure (`ss7configure`).

To restore event and message reporting, an association identifier *must* be specified when S7MP applications are launched. If the `Association_id` is not specified or set to *zero (0)*, event and message reporting is not restored when using the reconfiguration facilities (see `SAVE` and `RECONF` commands under the `RECONFIGURE` command, and `SAVE_CONFIG`).

5.2.1 Starting S7MP without an association ID

Type `s7mp<platform_id>` and press the Return key. This starts an S7MP application and associates it to a DIR process.

The enabled events are lost if you exit S7MP with an `Exit` command, or a `Ctrl/C` command.

5.2.2 Starting S7MP for preserving context

To do this you must start S7MP with an association ID. This identifier uniquely defines each management application on a given DIRECTOR. A management application that restarts on a platform has to re-associate with the same association ID. If an `END_ASSOCIATE`, and a subsequent `ASSOCIATE` command are issued, the previously specified association ID is used.

Example

Type `s7mp<platform_id> 3` and press the Return key.

This starts an S7MP application and associates it to a DIR process with the identifier 3.

If you type a **0** (zero) as the association ID, the result is the same as if you had not entered an association ID, i.e. you have all the features of the full *AutoReconfiguration* facility, but event reporting is not restored.

If you specify the same association ID as before, all previously enabled event message reporting is still registered.

The allowed range of values for an association ID is:

0.. 65535

Bee-CUSP Convergent Signaling Platform operating guide

For event message reporting to be restored, the value of the association ID must be in the range:

1.. 65535

The S7MP prompt

When you have issued the start command, the S7MP prompt is displayed.

```
S7MP_A>
```

The last character in the S7MP prompt indicates the platform ID value. S7MP can only see the processes on this one platform. On a server where two platforms coexist, you need to run S7MP twice with different platform IDs.

The management application is automatically associated with Bee-CUSP CSP when S7MP is started. Refer to **S7MP command descriptions** below for further details of the ASSOCIATE command.

5.3 Status indication display option

S7MP users have the option to display status indication reports in long or shortened versions. This choice is made by setting the environment variable S7MP_STATUS_SHORT to ON or OFF (Default=OFF).

When S7MP_STATUS_SHORT is set to ON, status indication messages are displayed on one line instead of the usual five lines:

Example: PSI display with S7MP_STATUS_SHORT = ON

```
galion> s7mpa

%%PSI REACHABLE DIR GALION 29-Aug-2002 12:10:08.00

%%PSI CONNECTED FEP GALION 29-Aug-2002 12:10:08.00
```

Example: PSI display with S7MP_STATUS_SHORT = OFF

```
galion> s7mpa

%%%%%%%% PROCESS STATUS INDICATION %%%%%%%%%

PROCESS REACHABLE:

Process type: DIR

Machine name: GALION

Timestamp: 29-Aug-2002 12:10:08.00

%%%%%%%% PROCESS STATUS INDICATION %%%%%%%%%

PROCESS CONNECTED:

Process type: FEP

Machine name: GALION

Timestamp: 29-Aug-2002 12:10:08.00 14
```

5.4 Restrictions

The *AutoReconfiguration* facility does not restore:

- `SYSFAIL` event reporting the values of the TRUNK entity attributes:
 - `SS7_K_TRNK_CGRBONSET`
 - `SS7_K_TRNK_CGRBABAT`

If these values have been modified and are not the same as the default values.

You must manually enable `SYSFAIL` event reporting using the `ENAEVT` command, and manually restore the values you want to the TRUNK entity attributes, using the `MODIFY` command. See **S7MP command descriptions** below, `RECONFIGURE`, `SAVE_CONFIG`, `MODIFY` and `ENAEVT`.

5.5 Command Syntax

An S7MP command is usually composed of three parts:

1. The command verb
2. A parameter (usually, but not always, required)
3. One or more qualifiers, which may be mandatory or optional.

For details see the *S7MP qualifiers* column in the tables shown in the *Bee-CUSP CSP entity reference guide* . Examples are shown in the table below.

Table 6: S7MP command syntax

Command verb	Parameter	Qualifier
SHOW	<i>entity_name</i>	<i>/ALL</i>
DELETE	<i>entity_name</i>	
GTT_RULE_CREATE	<i>gtt_rule_id</i>	<i>gtt_qualifiers</i>
TRACE	<i>file_spec</i>	

Command verb

See **S7MP command descriptions** below.

Parameter

The parameter can be any one of the following, depending on the command:

- *entity_name*
- *file_spec*
- *timestamp*
- *mode*
- *period_in_seconds*

You require only one parameter at a time for each command.

Entity_Name parameter

The *entity_name* parameter is an ASCII character string that specifies the name of an entity. It is composed of the different elements of the path taken (from the root) in the hierarchy tree, to arrive at the entity class.

Bee-CUSP Convergent Signaling Platform operating guide

For example, to describe link 1 belonging to linkset 2, managed by the MTP3 layer localized on the FE process subsystem, use the following path:

```
FEP_SUBSYSTEM NAME MTP3 LINKSET 2 LINK 1
```



Note

For FE process subsystem entities and sub-entities, the request is distributed if the FE process subsystem instance is '*'.

For the ANSI protocol, a destination instance can be entered either as an integer, or using the syntax:

for member:

```
<network_id>="<cluster_id>="<member_id>
```

for cluster:

```
<network_id>="<cluster_id>="
```

for network:

```
<network_id>=" ="
```

Network destinations are always referred to as <network_id>=" ="

Qualifiers

S7MP command qualifiers correspond to entity attributes. For full details of entity attributes, see the *Bee-CUSP CSP entity reference guide*.

5.5.1 Entering commands

S7MP commands are entered as verbs, parameters and qualifiers, separated by a space or tab character.

Continuation lines

Where there is not enough space to enter a long command on one line, use the standard convention of typing – (hyphen) at the end of the line. The continuation `_S7MP_C>` prompt is displayed and you can continue to type the rest of your command and press Return. For example:

```
S7MP_C>SHOW FEP_SUBSYSTEM * -
```

```
_S7MP_C>/ALL
```

Recalling commands

To recall a previously entered command, press the **up** arrow key at the `S7MP_C>` prompt. To modify a recalled command, use the **left** arrow or **right** arrow keys to position the cursor where you want to make the change. If the previous command filled more than one line, only the last line of the command is recalled.

Command prompting

S7MP prompts you for selected parameters if you do not provide them. If you omit required qualifiers, the command is not executed and a response is returned. When S7MP receives input from a command file, you are not prompted for missing items. All necessary verbs, parameters and qualifiers must be provided in the file itself.

5.5.2 Command results

The output result can be a response or notification.

5.5.2.1 Response

The result of a command is displayed on the terminal. The format of the reply is a `Response_code` and more information, given in text form, depending on the command. Examples are given in **S7MP command descriptions**.

S7MP formats the response information returned after the execution of a command. If the request is distributed, the consolidated response or all the responses are displayed, depending on the consolidation mode that has been defined. A consolidated response is not returned if the responses are not consistent. See `SET MODE /CONSOLIDATION` under **SET**.

Two kinds of response code are displayed, those set by Bee-CUSP CSP management and those indicating syntax errors.

Response_codes set by Bee-CUSP CSP management

These codes are always prefixed by `@SS7-` and contain a one-letter severity code. The one-letter severity codes are detailed below:

Table 7: One-letter severity codes

S	Success
I	Information
W	Warning
E	Error
F	Fatal

Examples of response_codes

```
@_SS7-S-NORMAL, normal successful completion
@_SS7-E-NOSUCHENT, no such entity
@_SS7-W-ENTITYEXISTS, entity already exists
@_SS7-I-AVAILALREADY, already available
@_SS7-F-NOSS7MGT, Bee-CUSP CSP management is not running
```

When the command has completed successfully, the success message is displayed only if Bee-CUSP CSP is operating in verbose mode.

Syntax errors

Syntax errors indicate that the format or qualifiers of the command are incorrect or insufficient. Whatever the cause, the command is not executed by Bee-CUSP CSP. Format: always prefixed by `@`

Example of syntax error

```
@NONAME-W-NOMSG, Message number is 229952
```

5.5.2.2 Notification

The Bee-CUSP CSP management interface returns all forms of output asynchronously. However, S7MP waits for a response output from Bee-CUSP CSP and displays it, after which you return to the S7MP prompt so that

interaction with S7MP is synchronous. If an event report occurs during the waiting time, it is handled after response completion.

When asynchronous event reports are received at the Bee-CUSP CSP management interface (once they have been enabled by the `ENAEVT` command), these are displayed on the terminal. If you do not wish these messages to be displayed, use the appropriate S7MP `DISEVT` command.

Synchronous responses have priority over asynchronous responses and this can cause situations where event reports are displayed immediately after a `DISEVT` command has been issued. This only occurs when the information was received while the S7MP program was processing the `DISEVT` command response. Disabled event reports received after the `DISEVT` command has been processed are not displayed.

5.6 Exiting S7MP

To leave S7MP, type `EXIT` and press Return.

5.7 S7MP Help system

S7MP provides a `HELP` system with information on each S7MP command, parameter and qualifier. See **S7MP command descriptions** for examples.

Chapter 6. S7MP command descriptions

This chapter describes the S7MP commands that can be executed on Bee-CUSP CSP. You must be operating in verbose mode for a message to be displayed when a command completes successfully.

The character * has two possible meanings in S7MP:

1. Used for global view, where '*' is specified as instance_id for the FEP entity. In this case only one entity is impacted by the command, and one result is provided to the user. For instance:

```
SHOW FEP * MTP3 DESTINATION 10/STATUS

FEP target list           : TRAIN2
Number of FEP target      : 1
Attribute Group           : Status
Destination State         : SS7_K_DEST_INACCESSIBLE
```

2. Used as a wildcard, where '*' is specified as instance_id for an entity other than the FEP entity. In this case, the user will receive as many results as there are entities. For instance:

```
SHOW FEP * MTP3 DESTINATION * /STATUS

FEP target list           : TRAIN2
Number of FEP target      : 1
Entity Name               : FEP_SUBSYSTEM * MTP3 DESTINATION 2
Attribute Group           : Status
Destination State         : SS7_K_DEST_ACCESSIBLE
FEP target list           : TRAIN2
Number of FEP target      : 1

Entity Name               : FEP_SUBSYSTEM * MTP3 DESTINATION 10
Attribute Group           : Status
Destination State         : SS7_K_DEST_INACCESSIBLE
.....
@SS7-I-NOOTHERINSTANCE, No other instance
```

6.1 ACTIVATE

Format

```
ACTIVATE linkset_entity_name
```

```
ACTIVATE link_entity_name
```

Parameters

linkset_entity_name, link_entity_name

The name of a LINKSET or LINK entity.

Qualifiers

None.

Description

Bee-CUSP Convergent Signaling Platform operating guide

Activates an MTP3 LINKSET or LINK.

When activating a linkset, all the links in the linkset will be activated.

You must activate the linkset *before* activating any of the links in this linkset.

This command supports the use of wildcards.

Example

```
S7MP_C> ACTIVATE FEP_SUBSYSTEM PARMA MTP3 LINKSET 1 LINK 1
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
```

```
Entity Name: FEP_SUBSYSTEM PARMA MTP3 LINKSET 1 LINK 1
```

```
Timestamp: 8-MAR-2003 15:08:31.10
```

Example

```
S7MP_C> ACTIVATE FEP_SUBSYSTEM PARMA MTP3 LINKSET 1
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
```

```
Entity Name: FEP_SUBSYSTEM PARMA MTP3 LINKSET 1
```

```
Timestamp: 8-MAR-2003 15:08:31.10
```

Example.

```
S7MP_C> ACTIVATE FEP_SUBSYSTEM PARMA MTP3 LINKSET 1
```

Produces the following result if the LINKSET is disabled:

```
@SS7-E-NOTDONE, action not performed
```

```
Entity Name: FEP_SUBSYSTEM PARMA MTP3 LINKSET 1
```

```
Timestamp: 8-MAR-2003 15:51:48.87
```

```
Action Status: SS7_K_STATE_INVALID
```

```
Reason: SS7_K_LKST_DISABLED
```

6.2 ADD

Format

ADD *entity_name* /*characteristic_attribute_values*

Parameters

entity_name

The name of an SCCP or LINKSET entity.

Qualifiers

CLINKSETLIST=

The member list of a combined LINKSET entity.

SSNLIST=

The SSN list for the SCCP entity.



Note

- The qualifiers shown can only be used with the ADD and REMOVE commands.
- Entities must be created before the ADD command can be used.
- The ADD command accepts more than one value per qualifier.

Description

Adds settable characteristic attribute values to a FEP or combined LINKSET or DESTINATION entity.

Refer to the *Bee-CUSP CSP entity reference guide* for details.

This command supports the use of wildcards.

Example

```
S7MP_C> ADD FEP_SUBSYSTEM BGATES SCCP /SSNLIST=(7,8,9)
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
Entity Name : FEP_SUBSYSTEM BGATES SCCP
Timestamp : 4-JUL-1994 13:09:19.33
Concerned SSN List (SSNLIST) :
SS7_K_SUCCESS: 7
SS7_K_SUCCESS: 8
SS7_K_SUCCESS: 9
```

6.3 ASSOCIATE

Format

ASSOCIATE /qualifiers

Parameters

None.

Qualifiers

PROC_STATUS_INDIC={ON, ALL, or OFF}.

Sets the value to ON, ALL or OFF.

Description

Bee-CUSP Convergent Signaling Platform operating guide

Associates an S7MP application with Bee-CUSP CSP management and declares it as a master copy.

The `ASSOCIATE` command is included in the S7MP code; therefore S7MP is automatically associated when it is started. However, if an `END_ASSOCIATE` command has been issued previously, the `ASSOCIATE` command must be issued to re-establish the virtual connection to Bee-CUSP CSP.

The `MASTER` role and the `NOCHECK` option are included in the command, so the S7MP application is declared as a master copy when the `ASSOCIATE` command is performed. If the `ASSOCIATE` command is run without specifying the `PROC_STATUS_INDIC` qualifier, as is the case at S7MP startup, the value taken is the value of the `PROC_STATUS_INDIC` environment variable, if defined. If it is not defined, the `PROC_STATUS_INDIC` option `ON` is taken.

If you wish to change the `PROC_STATUS_INDIC` option, you must first type the `END_ASSOCIATE` command and then type either `ASSOCIATE/PROC_STATUS_INDIC= ON, OFF or ALL` or `ASSOCIATE`. If the `PROC_STATUS_INDIC` environment variable is defined, it takes this value. If it is not defined, the value `ON` is taken.

The qualifiers `ON, OFF or ALL` determine what status indications are returned.

If set to `ON` or `ALL`, process status indications and last process status indications are returned.

If set to `OFF`, no status indications are returned.

Note



Several management applications can run concurrently, but you must ensure consistent behavior. This is not ensured by Bee-CUSP CSP.

This command does **not** support the use of wildcards.

Example

```
S7MP_C> ASSOCIATE
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion  
Timestamp: 4-MAR-2003 12:03:17.07
```

Example

```
S7MP_C> ASSOCIATE
```

Produces the following result if this management application has already been associated:

```
@SS7-W-ASSOALREADY associated already  
Timestamp: 4-MAR-2003 12:03:17.07
```

Example

The environment variable `S7MP_PROC_STATUS_INDIC` is not defined.

\$s7mpc

S7MP is started with the process status indication set to `ON`, producing output similar to the following:

```
Option: NO_CHECK
- PROCESS_STATUS_INDIC
role: MASTER
Association_id: 6
Timestamp: 12-MAR-2003 11:09:52.40
```

S7MP_C

```
%%%%%%%% PROCESS STATUS INDICATION %%%%%%%%%
PROCESS REACHABLE:
Process type: DIR
Machine name: COLETT
Timestamp: 12-MAR-2003 11:09:52.40
```

```
%%%%%%%% PROCESS STATUS INDICATION %%%%%%%%%
PROCESS REACHABLE:
Process type: FEP_SUBSYSTEM
Machine name: COLETT
Timestamp: 12-MAR-2003 11:09:52.40
```

To change the value of the qualifier process status indication, type the command:

```
S7MP_C> END_ASSOCIATE
```

Produces the following result:

```
@SS7-S-NORMAL, Normal successful completion
Association_id: 6
```

This ends the association 6.

```
S7MP_C> ASSOCIATE/PROC_STATUS_INDIC=OFF
```

Sets the new value for the qualifier process status indication to OFF and produces the following output:

```
@SS7-S-NORMAL, Normal successful completion
Option: NO_CHECK
- NO_PROCESS_STATUS_INDIC
role: MASTER
Association_id: 7
Timestamp: 12-MAR-2003 11:10:10.59
```

Process status indication information is **not** displayed.

If you want process status indications to be displayed again, you must follow the same procedure as before.

```
S7MP_C> END_ASSOCIATE
```

Produces the following result:

```
@SS7-S-NORMAL, Normal successful completion
Association_id: 7
Ends association 7.
```

```
S7MP_C> ASSOCIATE/PROC_STATUS_INDIC=ON
```

Produces the following result:

```
@SS7-S-NORMAL, Normal successful completion
Option: NO_CHECK
- PROCESS_STATUS_INDIC
role: MASTER
Association_id: 8
Timestamp: 12-MAR-2003 11:10:21.46
```

Process status indications will now be displayed. For example:

```
S7MP_C>

%%%%% PROCESS STATUS INDICATION %%%%%
PROCESS REACHABLE:
Process type: FEP_SUBSYSTEM
Machine name: COLETT
Timestamp: 12-MAR-2003 11:10:21.46
```

6.4 CLOSE_DIALOGS

Format

CLOSE_DIALOGS *tcap_bep_entity_name*

Parameters

tcap_bep_entity_name

The name of a TCAP_BEP entity.

Qualifiers

None.

Description

Prevents a TCAP_BEP entity from receiving any further BEGIN_DIALOG messages, but the current dialogs complete. This puts an application copy out of service.

This command supports the use of wildcards.

Example

```
S7MP_C> CLOSE_DIALOGS BEP_SUBSYSTEM KALIF TCAP_BEP 56
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
Entity Name: BEP_SUBSYSTEM KALIF TCAP_BEP 56
Timestamp : 21-MAR-2001 11:19:36.
```

Example

```
S7MP_C> CLOSE_DIALOGS BEP_SUBSYSTEM KALIF TCAP_BEP 57
```

Produces the following result if the TCAP_BEP entity instance does not exist:

```
@SS7-E-NOSUCHINSTANCE, no such instance
Class : TCAP_BEP
Instance_id : 57
Timestamp : 21-MAR-2001 11:19:36.90
```

Example

```
S7MP_C> CLOSE_DIALOGS BEP_SUBSYSTEM KALIF TCAP_BEP 156
```

Produces the following result if the TCAP_BEP entity is in a disabled state:

```
@SS7-E-NOTDONE, action not performed
Entity Name: BEP_SUBSYSTEM KALIF TCAP_BEP 156
Timestamp : 21-MAR-2001 11:26:26.81
Action Status : SS7_K_STATE_INVALID
Reason : SS7_K_TCAB_DISABLED
```

6.5 CONNECT

Format

CONNECT *entity_name*

Parameters

entity_name

The name of a FEP_SUBSYSTEM or a BEP_SUBSYSTEM entity.

Qualifiers

None.

Description

Connects FEP_SUBSYSTEM and BEP_SUBSYSTEM entities to the rest of the configuration, allowing FEP-to-BEP and peer-to-peer communication.

This command does **not** support the use of wildcards.

Examples

```
S7MP_C> CONNECT BEP_SUBSYSTEM FLORA
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
Entity Name: BEP_SUBSYSTEM FLORA
Timestamp : 14-MAR-2001 17:29:50.07
```

```
S7MP_C> CONNECT FEP_SUBSYSTEM PARMA
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
Entity Name: FEP_SUBSYSTEM PARMA
Timestamp: 12-MAR-2003 17:29:50.07
```

Example

```
S7MP_C> CONNECT FEP_SUBSYSTEM PARMA
```

Produces the following result if the child entities are disabled:

```
@SS7-E-NOTDONE, action not performed
Entity Name: FEP_SUBSYSTEM PARMA
Timestamp: 12-MAR-2003 17:29:40.91
Action Status: SS7_K_CONTEXT_INVALID
Reason: SS7_K_CHILDREN_DISABLED
```

6.6 COPY_ACTIVE

Format

`COPY_ACTIVE` *entity_name*

Parameters

entity_name

The name of the SCCP or NIF entity.

Qualifiers

None.

Description

Copies the contents of the active GTT rule file and table into the standby GTT rule file and table.
Copies the content of the active RK rule file and table into the standby RK rule file and table.
This command does **not** support the use of wildcards.

Sample command for GTT rules

```
S7MP_C> COPY_ACTIVE FEP_SUBSYSTEM * SCCP
```

Sample command for RK rules

```
S7MP_C> COPY_ACTIVE FEP_SUBSYSTEM * NIF
```

When operating in verbose mode, if no system message is issued after you have entered the command, the command has been executed successfully.

6.7 CREATE

Format

`CREATE` *entity_name* */list_of_characteristic_attributes_with_values*

Parameters

entity_name

The name of an entity

Qualifiers

/list_of_characteristic_attributes_with_values

List of items formatted in the following way: */qualifier=value*.

This command accepts as qualifiers any of the characteristic attributes of the entity to be created. When these attributes are “settable at creation” and “mandatory”, they must be specified as the qualifiers.

Refer to the *Bee-CUSP CSP entity reference guide* for details of the characteristic attributes of all entities.

Description

Creates an entity instance.

**Note**

When creating TCAP_BEP entities, the instance_id must be a decimal number (not hexadecimal).

This command does **not** support the use of wildcards.

Example

```
S7MP_C> CREATE FEP_SUBSYSTEM PARMA /OPC=12
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
Entity Name: FEP_SUBSYSTEM PARMA
Timestamp: 8-MAR-2003 15:03:56.78
Create_Att_Status: SS7_K_NO_STATUS
Originating Point Code: SS7_K_SUCCESS 12
```

Example

```
S7MP_C> CREATE FEP_SUBSYSTEM XGATES /OPC=16384
```

Produces an error result for ITU-T because the OPC cannot be > 16383:

```
@SS7-E-CREATERROR, create error
Entity Name: FEP_SUBSYSTEM XGATES
Timestamp: 1-MAR-2003 14:47:34.15
Create_Att_Status: SS7_K_NO_STATUS
Originating Point Code (OPC): SS7_K_INVALID_ATT_VAL 16384
```

Example

```
S7MP_C> CREATE FEP_SUBSYSTEM XGATES MTP3 LINKSET 1 -
_S7MP_C> LINK 1 /PLN=1/RANK=0/FEPNAM=XGATES
```

Produces the following result:

Bee-CUSP Convergent Signaling Platform operating guide

```
@SS7-S-NORMAL, normal successful completion
Entity Name: FEP_SUBSYSTEM XGATES MTP3 LINKSET 1 LINK 1
Timestamp: 1-MAR-2003 15:00:38.21
Create_Att_Status: SS7_K_NO_STATUS
Physical Link Number (PLN): SS7_K_SUCCESS 1
FEP Name (FEPNAME): SS7_K_SUCCESS XGATES
Link Rank (RANK): SS7_K_SUCCESS 0
```

Example

```
S7MP_C> CREATE FEP_SUBSYSTEM XGATES MTP3 LINKSET 1 LINK 1 /PLN=1
```

Produces the following result because the FE process name and rank qualifiers have not been specified:

```
@SS7-E-CREATERROR, create error
Entity Name: FEP_SUBSYSTEM XGATES MTP3 LINKSET 1 LINK 1
Timestamp: 1-MAR-2003 14:59:29.09
Create_Att_Status: SS7_K_NO_STATUS
Physical Link Number (PLN): SS7_K_SUCCESS 1
FEP Name (FEPNAME): SS7_K_REQUIRED_ATT_OMITTED
Link Rank (RANK): SS7_K_REQUIRED_ATT_OMITTED
```

Example

```
S7MP_C> CREATE BEP_SUBSYSTEM XGATES TCAP_BEP 83 /SSN=6
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
Entity Name : BEP_SUBSYSTEM XGATES TCAP_BEP 83
Timestamp : 1-JUL-2001 15:05:28.17
Create_Att_Status : SS7_K_NO_STATUS
Related Appli SSN (SSN) : SS7_K_SUCCESS 6
```

Note



You can create up to 9999 application level IDs for TCAP_BEP, SCCP_BEP and MTP_BEP entities:

The integer value of the Bee-CUSP CSP constants:

- SS7_K_TCAB_NAME
- SS7_K_SCCP_NAME
- SS7_K_MTPB_NAME

must be the same as the application level ID (in the range 0 to 9999)

6.8 DEACTIVATE

Format

DEACTIVATE *linkset_entity_name*

DEACTIVATE *link_entity_name*

Parameters

linkset_entity_name, link_entity_name

The name of a LINKSET or LINK entity.

Qualifiers

None.

Description

Deactivates an MTP3 LINKSET or LINK.

This command supports the use of wildcards.

Example

S7MP_C> DEACTIVATE FEP_SUBSYSTEM PARMA MTP3 LINKSET 1

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
Entity Name: FEP_SUBSYSTEM PARMA MTP3 LINKSET 1
Timestamp: 12-MAR-2003 17:39:45.49
```

Example

S7MP_C> DEACTIVATE FEP_SUBSYSTEM PARMA MTP3 LINKSET 1 LINK 1

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
Entity Name: FEP_SUBSYSTEM PARMA MTP3 LINKSET 1 LINK 1
Timestamp: 12-MAR-2003 17:39:45.92
```

Example

S7MP_C> DEACTIVATE FEP_SUBSYSTEM PARMA MTP3 LINKSET 1 LINK 1

Produces an error result if the parent entity of the LINK does not exist:

```
@SS7-E-NOSUCHPARENT, no such parent
Class: LINKSET
Instance_id: 1
Timestamp: 12-MAR-2003 17:39:45.92
```

6.9 DELETE

Format

DELETE *entity_name*

Parameters

entity_name

The name of an entity.

Qualifiers

None.

Description

Deletes an entity instance.

This command supports the use of wildcards.



Note

Certain entities cannot be deleted: refer to the *Bee-CUSP CSP entity reference guide* for further details.

Example

```
S7MP_C> DELETE FEP_SUBSYSTEM PARMA MTP3 DESTINATION 10 ROUTE 1
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
Entity Name: FEP_SUBSYSTEM PARMA MTP3 DESTINATION 10 ROUTE 1
Timestamp: 8-MAR-2003 15:05:06.60
```

Example

```
S7MP_C> DELETE FEP_SUBSYSTEM PARMA MTP3 DESTINATION 10
```

Produces an error result if the child entities of the DESTINATION have not previously been deleted:

```
@SS7-E-DELETERROR, delete error
Entity Name: FEP_SUBSYSTEM PARMA DESTINATION 10
Timestamp: 8-MAR-2003 15:51:48.87
Reason: SS7_K_HAS_CHILDREN
```

6.10 DISABLE

Format

DISABLE *entity_name*

Parameters

entity_name

The name of an entity.

Qualifiers

None.

Description

Disables an entity.

**Note**

A `DISABLE` action takes one second to complete when performed on a `ROUTE` entity. You should wait for the `SS7_K_ROUT_EVDISABLED` event to be reported before trying to disable the destination, and before using a `SHOW` directive to see the `ROUTE` status.

This command supports the use of wildcards.

Example

```
S7MP_C> DISABLE FEP_SUBSYSTEM PARMA MTP2 DATA_LINK 1
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
Entity Name: FEP_SUBSYSTEM PARMA MTP2 DATA_LINK 1
Timestamp: 12-MAR-2003 17:39:46.35
```

Example

```
S7MP_C> DISABLE FEP_SUBSYSTEM AMEMET SAAL HSSL 1
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
Entity Name: FEP_SUBSYSTEM AMEMET SAAL HSSL 1
Timestamp: 07-Feb-2002 15:32:03.00
```

Example

```
S7MP_C> DISABLE FEP_SUBSYSTEM PARMA MTP3 LINKSET 1
```

Produces the following result if the child entities of the `LINKSET` have not been previously disabled:

```
@SS7-E-NOTDONE, action not performed
Entity Name: FEP_SUBSYSTEM PARMA MTP3 LINKSET 1
Timestamp: 8-MAR-2003 15:51:48.87
Action Status: SS7_K_CONTEXT_INVALID
Reason: SS7_K_CHILDREN_ENABLED
```

6.11 DISCONNECT

Format

```
DISCONNECT subsystem_entity_name
```

Parameters**entity_name**

The name of a `FEP_SUBSYSTEM` entity and `BEP_SUBSYSTEM` entity.

Qualifiers

None.

Description

Disconnects FEP_SUBSYSTEM and BEP_SUBSYSTEM entities from the rest of the configuration, preventing both FEP-to-BEP and peer-to-peer communication. It also stops the FE process.

This command does **not** support the use of wildcards.

Example

```
S7MP_C> DISCONNECT FEP_SUBSYSTEM LEILA
```

Produces the following result:

```
S7MP_Y>
%%%%%%%% PROCESS STATUS INDICATION %%%%%%%%%
PROCESS UNREACHABLE:
Process type: FEP
Machine name: LEILA
Timestamp: 07-MAR-2003 16:18:01.00
```

Example

```
S7MP_C> DISCONNECT FEP_SUBSYSTEM GENOVA
```

Produces the following result if there is no response within the timeout:

```
@SS7-E-NORESPONSE, entity response timeout
Entity Name: FEP_SUBSYSTEM GENOVA
Timestamp: 12-MAR-2003 17:29:51.36
```

6.12 DISEVT

Format

DISEVT *entity_name* /*qualifiers*

Parameters

entity_name

The name of an entity.

Qualifiers

event_attributes

EVENTS

ALL

This command accepts as qualifiers any of the event attributes of the specified entity. Either the /events or /ALL qualifier can be used to disable all the events of the specified entity, but you cannot use other qualifiers at the same time.

Description

Disables some or all event reporting of the specified entity. Refer to the *Bee-CUSP CSP entity reference guide* for details of entity events. This command supports the use of wildcards.

Example

```
S7MP_E> DISEVT FEP_SUBSYSTEM ETALON SCCP /ALL
```

Produces the following result:

```
@SS7-S-NORMAL, Normal successful completion
Entity Name: FEP_SUBSYSTEM ETALON SCCP
Timestamp: 03-Jul-2003 14:58:52.00
SCCP Soft Failure (LEVELFAIL) : SS7_K_SUCCESS SS7_K_EVT_DISABLED
SCCP Soft Error (SWERROR) : SS7_K_SUCCESS SS7_K_EVT_DISABLED
Unknown Reason RFail (UNRSRF) : SS7_K_SUCCESS SS7_K_EVT_DISABLED
Unequip User RFail (UNQURF) : SS7_K_SUCCESS SS7_K_EVT_DISABLED
Prohibited SSN RFail (PSSNRF) : SS7_K_SUCCESS SS7_K_EVT_DISABLED
Congested SSN RFail (CSSNRF) : SS7_K_SUCCESS SS7_K_EVT_DISABLED
Prohibited SPC RFail (PSPCRF) : SS7_K_SUCCESS SS7_K_EVT_DISABLED
Congested SPC RFail (CSPCRF) : SS7_K_SUCCESS SS7_K_EVT_DISABLED
Syntax Error (SYNTAXER) : SS7_K_SUCCESS SS7_K_EVT_DISABLED
SCCP Mgt request dicarded (MGTRQDISC) : SS7_K_SUCCESS SS7_K_EVT_DISABLED
SCCP ready (READY) : SS7_K_SUCCESS SS7_K_EVT_DISABLED
Error when translating GT (GTERROR) : SS7_K_SUCCESS SS7_K_EVT_DISABLED
Segmentation error (SEGERR) : SS7_K_SUCCESS SS7_K_EVT_DISABLED
Reassembly error (REAERR) : SS7_K_SUCCESS SS7_K_EVT_DISABLED
Remote Delivery Problem (REMDELPB) : SS7_K_SUCCESS SS7_K_EVT_DISABLED
```

Example

```
S7MP_C> DISEVT FEP_SUBSYSTEM ETALON SCCP /READY
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
Entity Name: FEP_SUBSYSTEM ETALON SCCP
Timestamp: 1-MAR-2003 15:08:50.47
SCCP ready (READY): SS7_K_SUCCESS SS7_K_EVT_DISABLED
```

Example

```
S7MP_C> DISEVT FEP_SUBSYSTEM ETALON MTP3 LINKSET 1 /EVENTS
```

Produces the following result if LINKSET 1 does not exist:

```
@SS7-E-NOSUCHINSTANCE, no such instance
Class: LINKSET
Instance_id: 1
Timestamp: 1-MAR-2003 15:09:31.20
```

6.13 DISMSG

Format

DISMSG *sccp_entity_name*

Parameters

sccp_entity_name

The name of an SCCP entity.

Qualifiers

None.

Description

Disables reporting of SCCP messages. Refer to the *Entity Reference Guide* for further details.

This command does not support the use of wildcards.

Example

```
S7MP_C> DISMSG FEP_SUBSYSTEM PARMA SCCP
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion  
Entity Name: FEP_SUBSYSTEM PARMA SCCP  
Timestamp : 5-JUNE-2001 14:50:26.86
```

Example

```
S7MP_C> DISMSG FEP_SUBSYSTEM PILOTE SCCP
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion  
Entity Name: FEP_SUBSYSTEM PILOTE SCCP  
Timestamp : 5-JUNE-2001 14:50:26.86
```

Example

```
S7MP_C> DISMSG FEP_SUBSYSTEM PILOTE SCCP
```

Produces the following result if the subsystem node is unreachable:

```
@SS7-E-NODEUNREACH, node unreachable  
Class : FEP_SUBSYSTEM  
Instance_id : PILOTE  
Timestamp : 5-JUNE-2001 14:50:26.86
```

6.14 DO

Format

DO *file_spec*

Parameters

file_spec

The name of a file.

Qualifiers

None.

Description

Executes a command file.

**Note**

The file itself cannot contain other DO commands. The file extension name can be anything you want, but if no extension is used the value *.S7MP* (in upper case) is used by default.

Example

```
S7MP_C> DO CONFIGURATION
```

If the file CONFIGURATION.S7MP contains the following commands:

```
! create commands
```

```
CREATE FEP_SUBSYSTEM XGATES MTP3 LINKSET 1 LINK 2 /PLN=1/
```

```
FEPNAM=XGATES/RANK=2CREATE FEP_SUBSYSTEM XGATES MTP3 LINKSET 1 LINK 3 /PLN=2/
```

```
FEPNAM=XGATES/RANK=3
```

```
! show commands
```

```
SHOW FEP_SUBSYSTEM XGATES MTP3 LINKSET 1 LINK 2 /CHARACTERISTICS
```

```
SHOW FEP_SUBSYSTEM XGATES MTP3 LINKSET 1 LINK 3 /CHARACTERISTICS
```

The output is as follows:

```
CREATE FEP_SUBSYSTEM XGATES MTP3 LINKSET 1 LINK 2 /PLN=1 /FEPNAM=XGATES/RANK=2
```

```
@SS7-S-NORMAL, normal successful completion
```

```
Entity Name: FEP_SUBSYSTEM XGATES MTP3 LINKSET 1 LINK 2
```

```
Create_Att_Status: SS7_K_NO_STATUS
```

```
Timestamp: 1-MAR-2003 15:14:42.00
```

```
Physical Link Number (PLN): SS7_K_SUCCESS 1
```

```
FEP Name (FEPNAME): SS7_K_SUCCESS XGATES
```

```
Link Rank (RANK): SS7_K_SUCCESS 2
```

```
CREATE FEP_SUBSYSTEM XGATES MTP3 LINKSET 1 LINK 3 /PLN=2 /FEPNAM=XGATES/RANK=3
```

```
@SS7-S-NORMAL, normal successful completion
```

```
Entity Name: FEP_SUBSYSTEM XGATES MTP3 LINKSET 1 LINK 3
```

```
Timestamp: 1-MAR-2003 15:14:42.37
```

Bee-CUSP Convergent Signaling Platform operating guide

```
Create_Att_Status: SS7_K_NO_STATUS
Physical Link Number (PLN): SS7_K_SUCCESS 2
FEP Name (FEPNAME): SS7_K_SUCCESS XGATES
Link Rank (RANK): SS7_K_SUCCESS 3

SHOW FEP_SUBSYSTEM XGATES MTP3 LINKSET 1 LINK 2 /CHARACTERISTICS

@SS7-S-NORMAL, normal successful completion

Entity Name: FEP_SUBSYSTEM XGATES MTP3 LINKSET 1 LINK 2
Timestamp: 1-MAR-2003 15:14:42.98
Attribute Group: Characteristics
Physical Link Number (PLN): 1
FEP Name (FEPNAME): XGATES
Link Rank (RANK): 2TB congestion level (TBCONGLEVEL): 0
Periodic SLT procedure used (SLTUSE): SLT_IS_ONLY

SHOW FEP_SUBSYSTEM XGATES MTP3 LINKSET 1 LINK 3 /CHARACTERISTICS

@SS7-S-NORMAL, normal successful completion

Entity Name: FEP_SUBSYSTEM XGATES MTP3 LINKSET 1 LINK 3
Timestamp: 1-MAR-2003 15:14:43.34
Attribute Group: Characteristics
Physical Link Number (PLN): 2
FEP Name (FEPNAME): XGATES
Link Rank (RANK): 3TB congestion level (TBCONGLEVEL): 0
Periodic SLT procedure used (SLTUSE): SLT_IS_ONLY
```

6.15 DUMP

Format

DUMP *entity_name* [/qualifier]

DUMP FEP * NIF

DUMP FEP * MTP3

DUMP FEP * SCCP

Parameters

Entity_name

Description

The output directory in all cases is: `usr/var/ss7/platform_id`

- DUMP FEP * NIF dumps the active RK and screening (SR) rules files.
The file names are:
 - FEP_< PLATFORM_ID>_<MACHINE_NAME>_RK_RULES.DMP
 - FEP_< PLATFORM_ID>_<MACHINE_NAME>_SR_RULES.DMP

- `DUMP FEP * MTP3` dumps the MTP3 database.
The file name is: `FEP_< PLATFORM_ID>_<MACHINE_NAME>_MTP3.DMP`
- `DUMP FEP * SCCP` dumps the SCCP database.
The file name is: `FEP_< PLATFORM_ID>_<MACHINE_NAME>_SCCP.DMP`

Examples

The following examples show the contents of the .DMP files (when displayed with the `more` command):

- `FEP_C_PRO24_RK_RULES.DMP`
- `FEP_C_PRO24_SR_RULES.DMP`
- `FEP_C_PRO24_MTP3.DMP`
- `FEP_C_PRO24_SCCP.DMP`

where C is PLATFORM_ID and PRO24 is MACHINE_NAME.

RK rules (FEP_C_PRO24_RK_RULES.DMP)

```
ROUTING_SECTION
# 4 RK rules found
# RK rule 1
cgpa=(11/pc,6/ssn);pc=1;ssn=6;as=2;
# RK rule 2
cgpa=(15/pc,6/ssn);pc=1;ssn=6;as=2;
# RK rule 3
igt="1010";as=1;
# RK rule 4
pc=1;ssn=6;as=10;
FOR SR RULES
```

SR rules (FEP_C_PRO24_SR_RULES.DMP)

```
SCREENING_SECTION
# 300 SR rules found
# SR rule 1
cdpa=("deadbeef"/igt);opc!=4321;screen=1;
# SR rule 2
cdpa=("deadbeef.*"/igt);opc!=4321;screen=2;
# SR rule 3
cdpa=("cafedeca.*"/igt);opc!=678;screen=3;
# SR rule 4
opc=555;screen=4;
# SR rule 5
igt="0123456789";so="628122834920";sr="628129537207";si=23;screen=5;
# SR rule 6
pc=1234;ssn=5;screen=6;
# SR rule 7
so="12";sr="22";screen=7;
# SR rule 8
cdpa=("8888.*"/igt);opc!=88;screen=8;
# SR rule 9
cdpa=("9999.*"/igt);opc!=99;screen=9;
# SR rule 10
```

Bee-CUSP Convergent Signaling Platform operating guide

```
cdpa= ("1010"/igt);opc!=999;screen=10;
# SR rule 11
cdpa= ("1111"/igt);opc!=999;screen=11;
# SR rule 12
cdpa= ("1212"/igt);opc!=999;screen=12;
# SR rule 13
cdpa= ("1313"/igt);opc!=999;screen=13;
# SR rule 14
cdpa= ("1414"/igt);opc!=999;screen=14;
....
```

MTP3 database (FEP_C_PRO24_MTP3.DMP)

```
DECss7_PLATFORM_ID : C
DECss7_PROTOCOL : ITU-T
DECss7_SUBSYSTEM : FEP
SUBSYSTEM_NODE : PRO242
DUMP_DATE_TIME : 05-08-2005:13:06:57
```

```
TOTAL_NUMBER_OF_LINKSETS : 8
TOTAL_NUMBER_OF_DESTINATIONS : 9
```

===== DETAILS OF THE LINKSETS =====

----- LINKSET 1 -----

```
LINKSET_STATE : AVAILABLE
NUMBER_OF_LINKS : 16
```

LINKS_LIST :

LINK_ID	STATE	SUB_STATE	ON_THIS_FEP	RANK
-----	-----	-----	-----	----
LINK 0	AVAILABLE	UNBUFFERED	TRUE	0
LINK 1	AVAILABLE	UNBUFFERED	TRUE	1
LINK 2	AVAILABLE	UNBUFFERED	TRUE	2
LINK 3	AVAILABLE	UNBUFFERED	TRUE	3
LINK 4	AVAILABLE	UNBUFFERED	TRUE	4
LINK 5	AVAILABLE	UNBUFFERED	TRUE	5
LINK 6	AVAILABLE	UNBUFFERED	TRUE	6
LINK 7	AVAILABLE	UNBUFFERED	TRUE	7
LINK 8	AVAILABLE	UNBUFFERED	TRUE	8
LINK 9	AVAILABLE	UNBUFFERED	TRUE	9
LINK 10	AVAILABLE	UNBUFFERED	TRUE	10
LINK 11	AVAILABLE	UNBUFFERED	TRUE	11
LINK 12	AVAILABLE	UNBUFFERED	TRUE	12
LINK 13	AVAILABLE	UNBUFFERED	TRUE	13
LINK 14	AVAILABLE	UNBUFFERED	TRUE	14
LINK 15	AVAILABLE	UNBUFFERED	TRUE	15

ROUTING_TABLE :

SLS	CURRENT_RTG	REFERENCE_RTG	RANK	LINK_RTG
---	-----	-----	----	-----
0	LINK 0	LINK 0	0	LINK 0

1	LINK 1	LINK 1	1	LINK 1
2	LINK 2	LINK 2	2	LINK 2
3	LINK 3	LINK 3	3	LINK 3
4	LINK 4	LINK 4	4	LINK 4
5	LINK 5	LINK 5	5	LINK 5
6	LINK 6	LINK 6	6	LINK 6
7	LINK 7	LINK 7	7	LINK 7
8	LINK 8	LINK 8	8	LINK 8
9	LINK 9	LINK 9	9	LINK 9
10	LINK 10	LINK 10	10	LINK 10
11	LINK 11	LINK 11	11	LINK 11
12	LINK 12	LINK 12	12	LINK 12
13	LINK 13	LINK 13	13	LINK 13
14	LINK 14	LINK 14	14	LINK 14
15	LINK 15	LINK 15	15	LINK 15

DESTINATION_LIST (Destinations reachable by this linkset) :
10

```
----- LINKSET 2 -----
LINKSET_STATE : AVAILABLE
NUMBER_OF_LINKS : 8
.....
```

SCCP database (FEP_C_PRO24_SCCP.DMP)

```
DUMP REMOTE SCCP DATABASE: ==>>
PC: 12                                PROHIBITED
```

6.16 ENABLE

Format

ENABLE *entity_name*

Parameters

entity_name

The name of an entity.

Qualifiers

None.

Description

Enables an entity.

This command supports the use of wildcards.

Example

```
S7MP_C> ENABLE FEP_SUBSYSTEM PARMA
```

Produces the following result:

Bee-CUSP Convergent Signaling Platform operating guide

```
@SS7-S-NORMAL, normal successful completion
Entity Name: FEP_SUBSYSTEM PARMA
Timestamp: 8-MAR-2003 15:10:57.39
```

Example

```
S7MP_C> ENABLE FEP_SUBSYSTEM PARMA SCCP
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
Entity Name: FEP_SUBSYSTEM PARMA SCCP
Timestamp: 5-MAR-2003 14:40:49.34
```

Example

```
S7MP_C> ENABLE FEP_SUBSYSTEM PARMA SCCP
```

Produces the following result if the entity has already been enabled:

```
@SS7-I-AVAILALREADY, available already
Entity Name: FEP_SUBSYSTEM PARMA SCCP
Timestamp: 5-MAR-2003 14:40:49.34
```

Example

```
S7MP_C> ENABLE FEP_SUBSYSTEM PARMA MTP3 LINKSET 1
```

Produces the following result if the parent entity is disabled:

```
@SS7-E-NOTDONE, action not performed
Entity Name: FEP_SUBSYSTEM PARMA MTP3 LINKSET 1
Timestamp: 8-MAR-2003 15:51:50.59
Action Status: SS7_K_CONTEXT_INVALID
Reason: SS7_K_PARENT_DISABLED
```

6.17 ENAEVT

Format

ENAEVT *entity_name* /*qualifiers*

Parameters

entity_name

The name of an entity.

Qualifiers

event_attributes

EVENTS

ALL

This command accepts the event attributes of the specified entity as qualifiers. Either the /EVENTS or the /ALL qualifier can be used to enable all the events of the specified entity, but you cannot use another qualifier at the same time.

Description

Enables some or all entity events to be reported to the S7MP that has issued the command, but does not cause the occurrence of events. Refer to the *Bee-CUSP CSP entity reference guide* for details of all entity events. This command supports the use of wildcards.

Example

```
S7MP_C> ENAEVT FEP_SUBSYSTEM ETALON MTP3 LINKSET 1 /EVENTS
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
Entity Name           : FEP_SUBSYSTEM ETALON MTP3 LINKSET 1
Timestamp             : 03-Jul-2003 15:17:39.48
Start LinkSet Unavail (STAUNAVAIL) : SS7_K_SUCCESS
SS7_K_EVT_ENABLED
Stop LinkSet Unavail (STOUNAVAIL)  : SS7_K_SUCCESS
SS7_K_EVT_ENABLED
LinkSet modif completed (RULEMODIF) : SS7_K_SUCCESS
SS7_K_EVT_ENABLED
```

Example

```
S7MP_C> ENAEVT FEP_SUBSYSTEM XGATES MTP3 LINKSET 1 LINK /ACHBACK/SLTSUCC
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
Entity Name           : FEP_SUBSYSTEM XGATES MTP3 LINKSET 1 LINK 1
Timestamp             : 1-MAR-2003 15:18:40.99
Local Auto. Chg.back (ACHBACK)     : SS7_K_SUCCESS SS7_K_EVT_ENABLED
Signalling Link Test Success (SLTSUCC) : SS7_K_SUCCESS SS7_K_EVT_ENABLED3
```

Example

```
S7MP_C> ENAEVT FEP_SUBSYSTEM XGATES MTP3 DESTINATION 23 /ALL
```

Produces the following result if DESTINATION 23 has not been defined:

```
@SS7-E-NOSUCHINSTANCE, no such instance
Class           : DESTINATION
Instance_id     : 23
Timestamp       : 1-MAR-2003 15:19:55.46
```

Example

The following examples show event reports that could be displayed on the terminal if the related event reporting is enabled:

```
%%%%% E V E N T %%%%%%
Entity Name           : FEP_SUBSYSTEM PARMA MTP3 LINKSET 1 LINK 1
Event Id             : 65
Event Name           : Automatic ChangeOver
```

```
Timestamp                : 1-MAR-2003 18:16:50.48

%%%% E V E N T %%%%%%
Entity Name               : FEP_SUBSYSTEM PARMA MTP2 DATA_LINK 1
Event Id                  : 25
Event Name                : Link Failure
Timestamp                 : 1-MAR-2003 18:13:20.50
```

6.18 ENAMSG

Format

ENAMSG *sccp_entity_name*

Parameters

sccp_entity_name

The name of an SCCP entity.

Qualifiers

None.

Description

Enables SCCP messages to be reported to the S7MP that has issued the command. Refer to the *Entity reference guide* for further details.

This command does not support the use of wildcards.

Example

```
S7MP_C>ENAMSG FEP_SUBSYSTEM RIMINI SCCP
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
Entity Name: FEP_SUBSYSTEM RIMINI SCCP
Timestamp : 5-MAR-2001 14:50:25.14
```

The following example shows an SCCP message report that could be displayed on the terminal if SCCP message reporting has been enabled:

```
%%%% S C C P M E S S A G E %%%%%%
Entity Name: FEP_SUBSYSTEM RIMINI SCCP APPLICATION 7
N_PC_STATE
AFFECTED_PC: 13
PC Status: PC Inaccessible
Timestamp : 15-MAR-2001 10:31:19.45
```

Example

```
S7MP_C>ENAMSG FEP_SUBSYSTEM PARMAA SCCP
```

Produces the following result if the parent entity of SCCP does not exist:

```
@SS7-E-NOSUCHPARENT, no such parent  
Class : FEP_SUBSYSTEM  
Instance_id : PARMAA  
Timestamp : 5-MAR-2001 14:50:25.14
```

6.19 END_ASSOCIATE

Format

END_ASSOCIATE

Parameters

None.

Qualifiers

None.

Description

Disassociates an S7MP application from Bee-CUSP CSP management.



Caution

When you issue the END_ASSOCIATE command, any previously enabled event and SCCP message reporting is lost.

This command does **not** support the use of wildcards.

Example

```
S7MP_C> END_ASSOCIATE
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion  
Timestamp: 8-MAR-2003 15:05:06.60
```

Example

```
S7MP_C> END_ASSOCIATE
```

Produces the following result if the association has already been closed:

```
@SS7-E-ASSOCLOSED, association closed  
Timestamp: 8-MAR-2003 15:05:06.60
```

Example

```
S7MP_C> END_ASSOCIATE
```

Produces the following result if the association does not exist:

```
@SS7-E-NOSUCHASSO, no such association
Association: 0
Timestamp : 31-DEC-1993 14:49:42.06
```

6.20 EXIT

Format

EXIT

Parameters

None.

Qualifiers

None.

Description

The **EXIT** command causes you to exit the management program and return to the current shell. You can also type **Ctrl/C** at the **S7MP_C>** prompt to quit **S7MP**.

Example

```
S7MP_C>exit
```

Exits **S7MP**.

6.21 GTT_RULE_CREATE

Format

GTT_RULE_CREATE *gtt_rule_id* /*gtt_qualifiers*

Parameters

gtt_rule_id

The identification of the rule to be created.

Qualifiers

P_GTAI

Address information field for the GTT rule pattern. GTAI qualifiers in rule definitions are entered as a hexadecimal sequence in network or natural order depending on the choice of GTT release. Refer to the relevant section in the *Installation & configuration guide* for syntax details.

P_GTI

Global title indicator field for the GTT rule pattern (NAI_ONLY, TT_ONLY, TT_NP_ES, TT_NP_ES_NAI).

P_MATCH

Pattern matching mechanism for the GTT rule pattern (EXACT / WILDCARD).

P_NAI

Nature of address information field for the GTT rule pattern.

P_NP

Numbering plan field for the GTT rule pattern.

P_RNB

Reserved national bit field for the GTT rule pattern.

P_TT

Translation type field for the GTT rule pattern.

T_DISTRIB

Distribution mechanism field for the GTT rule translation (LOADSHARING /MASTERBACKUP).

T_GTAI

Address information field for the GTT rule translation. See examples under **GTT_RULE_CREATE**. You can also refer to the *Bee-CUSP CSP installation & configuration guide* for syntax details.

T_GTI

Global title indicator field for the GTT rule translation (NO_GT, NAI_ONLY, TT_ONLY, TT_NP_ES, TT_NP_ES_NAI).

T_NAI

Nature of address information field for the GTT rule translation.

T_NP

Numbering Plan field for the GTT rule translation.

T_PPC

Primary point code for the GTT rule translation.

T_PSSN

Primary subsystem number for the GTT rule translation.

T_RI

Routing indicator field for the GTT rule translation.

T_RNB

Reserved national bit field for the GTT rule translation.

T_TT

Translation type field for the GTT rule translation.

T_SPC

Secondary point code for the GTT rule translation.

T_SSSN

Secondary subsystem number for the GTT rule translation.

Description

Creates a GTT rule in the standby file. The command has completed successfully if there is no system message. When the command has not been successful, there is a system message indicating the cause of the failure.

All values must be supplied in hexadecimal, except for T_PPC, T_PSSN, T_SPC, and T_SSSN qualifiers; see below.

Note



The default mode is natural order.

Special case: T_PPC, T_PSSN, T_SPC, and T_SSSN qualifiers

Two GTT options are available to users:

- If the `SS7_GTT_RELEASE` environment variable in the `ss7_parameters` file is set to `NATURAL_ORDER` (default), or if an unknown value or no value is specified, then the enhanced GTT version is used (natural order of GTAI qualifiers, odd and even number of bytes in GTAI pattern description), and the values of the PC and SSN qualifiers must be supplied in decimal.
- If the `SS7_GTT_RELEASE` environment variable in the `ss7_parameters` file is set to `INITIAL`, the values of the PC and SSN qualifiers must be supplied in hexadecimal. The GTAI pattern must be specified in network order and, for an exact match, with an even number of bytes in the GTAI pattern description.

Note



The default mode is natural order.

Example

```
S7MP_C> GTT_RULE_CREATE 1 /P_RNB=1 /P_GTI=TT_ONLY /P_TT=1 -
_S7MP_C> /P_GTAI=800/P_MATCH=WILDCARD /T_RNB=1 /T_PPC=2 -
_S7MP_C> /T_PSSN=6 /T_SPC=3/T_GTI=TT_NP_ES /T_TT=2 /T_NP=2 -
_S7MP_C> /T_GTAI=ABCD /T_DISTRIB=LOADSHARING
```

Example

```
S7MP_C> GTT_RULE_CREATE 1 /P_RNB=1 /P_GTI=TT_ONLY /P_TT=1 -
_S7MP_C> /P_GTAI=800 /P_MATCH=WILDCARD /T_RNB=1 /T_PPC=2 -
_S7MP_C> /T_PSSN=6 /T_SPC=3 /T_GTI=TT_NP_ES /T_TT=2 /T_NP=2 -
_S7MP_C> /T_GTAI=ABCDEF /T_DISTRIB=LOADSHARING
```

Produces the following result if rule 1 already exists:

```
@SS7-E-RULEEXISTS, Rule already exists
```

Example

```
S7MP_C> GTT_RULE_CREATE 2 /P_RNB=1 /P_GTI=TT_ONLY /P_TT=1 -
_S7MP_C>/P_MATCH=EXACT /T_RNB=1/T_PPC=2 -
_S7MP_C>/T_DISTRIB=LOADSHARING
```

Produces the following result as the pattern you entered was not correct:

@SS7-E-INVPATTERN, Invalid pattern

Example

```
S7MP_C> GTT_RULE_CREATE 2 /P_RNB=1 /P_GTI=TT_ONLY /P_TT=1 -
_S7MP_C> /P_MATCH=WILDCARD/T_DISTRIB=LOADSHARING
```

Produces the following result:

@SS7-E-INVACTION, Invalid action

Example

```
S7MP_C> GTT_RULE_CREATE 2 /P_RNB=1 /P_GTI=TT_ONLY /P_TT=2 /P_MATCH=WILDCARD
/P_GTAI=9 -
_S7MP_C> /T_GTAI=01234ABC$ /T_DISTRIB=LOADSHARING /T_PPC=5 /T_PSSN=7
```

where \$ represents the GTAI incoming value and 01234ABC is the header to be added to the GTAI of the incoming message. If the incoming GTAI is 99, this rule produces the following result: the outgoing GTAI is 01234ABC99.

Example

```
S7MP_C> GTT_RULE_CREATE 2 /P_RNB=1 /P_GTI=TT_ONLY /P_TT=3 /P_MATCH=WILDCARD
/P_GTAI=6
-_S7MP_C>/T_GTAI=$:2 /T_DISTRIB=LOADSHARING /T_PPC=5 /T_PSSN=7
```

where \$ represents the GTAI incoming value and :2 is used to remove the first 2 digits of the incoming GTAI value. If the incoming GTAI is 678, this rule produces the following result: the outgoing GTAI is 8.

6.22 GTT_RULE_DELETE

Format

GTT_RULE_DELETE *gtt_rule_id*

Parameters

gtt_rule_id

The identification of the rule to be deleted.

Qualifiers

None.

Description

Deletes a GTT rule from the standby file. The rule has been successfully deleted if there is no system message. If the specified rule does not exist in the standby file, an error occurs and a system message is displayed.

Example

```
S7MP_C> GTT_RULE_DELETE 1
```

Example

```
S7MP_C> GTT_RULE_DELETE 3
```

Produces the following result if rule 3 does not exist:

```
@SS7-E-NOSUCHINSTANCE, No such instance
```

6.23 GTT_RULE_MODIFY

Format

`GTT_RULE_MODIFY gtt_rule_id /gtt_qualifiers`

Parameters

gtt_rule_id

The identification of the rule to be modified.

Qualifiers

P_GTAI

Address information field for the GTT rule pattern.

P_GTI

Global title indicator field for the GTT rule pattern (NAI_ONLY, TT_ONLY, TT_NP_ES, TT_NP_ES_NAI).

P_MATCH

Pattern matching mechanism for the GTT rule pattern (EXACT / WILDCARD).

P_NAI

Nature of address information field for the GTT rule pattern.

P_NP

Numbering plan field for the GTT rule pattern.

P_RNB

Reserved national bit field for the GTT rule pattern.

P_TT

Translation type field for the GTT rule pattern.

T_DISTRIB

Distribution mechanism field for the GTT rule translation (LOADSHARING /MASTERBACKUP).

T_GTAI

Address information field for the GTT rule translation.

See examples in **GTT_RULE_CREATE**.

T_GTI

Global title indicator field for the GTT rule translation (NO_GT, NAI_ONLY, TT_ONLY, TT_NP_ES, TT_NP_ES_NAI).

T_NAI

Nature of address information field for the GTT rule translation.

T_NP

Numbering plan field for the GTT rule translation.

T_PPC

Primary point code for the GTT rule translation.

T_PSSN

Primary subsystem number for the GTT rule translation.

T_RI

Routing indicator field for the GTT rule translation.

T_RNB

Reserved national bit field for the GTT rule translation.

T_TT

Translation type field for the GTT rule translation.

T_SPC

Secondary point code for the GTT rule translation.

T_SSSN

Secondary subsystem number for the GTT rule translation.

Description

Modifies a GTT rule in the standby file.

All values must be supplied in decimal or hexadecimal (*natural* or *initial* order, respectively) except for T_PPC, T_PSSN, T_SPC, and T_SSSN qualifiers.

Example

```
S7MP_C> GTT_RULE_MODIFY 1 /P_GTAI=900
```

Example

```
S7MP_C > GTT_RULE_MODIFY 2 /P_GTAI=900
```

Produces the following result:

```
@SS7-E-NOSUCHINSTANCE, No such instance
```

Example

```
S7MP_C> GTT_RULE_MODIFY 1 /P_TT=62620
```

Produces the following result as the P_TT field value was incorrect:

```
@SS7-E-INVPATTERN, Invalid pattern
```

Example

```
S7MP_P> GTT_RULE_MODIFY 1 /T_NAI=1
```

Produces the following result:

```
@SS7-E-INVACTION, Invalid action
```

6.24 GTT_RULE_SHALL

Format

GTT_RULE_SHALL /*qualifiers*

Parameters

None.

Qualifiers

ACTIVE

The active file is searched for the GTT rules.

STANDBY

The standby file is searched for the GTT rules.

Description

Displays the GTT rule identifiers contained in the active or standby file, whether loaded or unloaded. If you do not specify a qualifier, the standby file is searched by default.

Example

```
S7MP_C> GTT_RULE_SHALL
```

Produces the following result:

```
Registered rules:
1
```

Example

```
S7MP_C> GTT_RULE_SHALL
```

Produces the following result when there are no rules registered in the standby set:

```
No rule registered
```

Example

```
S7MP_C> GTT_RULE_SHALL /STANDBY
```

Produces the following result when there are rules registered in the standby set:

```
Registered rules:
10
```

Example

```
S7MP_C> GTT_RULE_SHALL /ACTIVE
```

Produces the following result when there are rules registered in the standby set:

```
No rule registered
```

6.25 GTT_RULE_SHOW

Format

GTT_RULE_SHOW *gtt_rule_id* /*qualifiers*

Parameters

GTT_rule_id

The identification of the rule to be displayed.

Qualifiers

ACTIVE

The active file is searched for the GTT rule.

STANDBY

The standby file is searched for the GTT rule.

T_GTAI

Displays the T_GTAI qualifier with the new syntax.

See examples in **GTT_RULE_CREATE**.

Description

Displays the contents of a GTT rule in the active or standby file. If you do not specify a qualifier, the STANDBY file is searched.

Field values are in hexadecimal mode.

Example

```
S7MP_C> GTT_RULE_SHOW 1
```

Produces information for rule 1 in the standby file, similar to the following:

```
Rule:
      Rule Id                      (RULEID) : 1
Pattern:
      Match                        (P_MATCH) : WILDCARD
      Reserved National Bit        (P_RNB) : 1
      Global Title Indicator        (P_GTI) : 2
      Translation Type              (P_TT) : 1
      Global Title Address Information (P_GTAI) : 900
Action:
      Distribution                  (T_DISTRIB) : LOADSHARING
      Reserved National Bit        (T_RNB) : 1
      Global Title Indicator        (T_GTI) : 3
      Translation Type              (T_TT) : 2
      Numbering Plan                (T_NP) : 2
      Primary PC                    (T_PPC) : 2
      Secondary PC                  (T_SPC) : 3
      Primary SSN                   (T_PSSN) : 6
      Global Title Address Information (T_GTAI) : ABCD
Status:
      State                        (STATE) : NOT_LOADED
```

Example

```
S7MP_P> GTT_RULE_SHOW 2
```

Produces the following result if rule 2 does not exist:

```
@SS7-E-NOSUCHINSTANCE, No such instance
```

Example

```
S7MP_P> GTT_RULE_SHOW 1 /active
```

Produces the following result if rule 1 does not exist in the active file:

```
@SS7-E-NOSUCHINSTANCE, No such instance
```

6.26 HELP

Format

HELP

Parameters

None.

Qualifiers

COMMANDS

Topics

Subtopics

Description

The **HELP** command gives you information about S7MP commands and parameters. When you call the **HELP** system without specifying a command or topic, it displays a list of commands and parameters for which help is available.

Type the name of the command or topic you want after the *topic ?* prompt, and then press the return key. The help text is displayed and if there is more information

```
Additional information available:
```

is displayed with a list of the subtopics for which there is more information.

To return to a higher level of information at the **Subtopic** level, press the return key. The **Topic?** prompt is displayed. Type the **?** character and press Enter. The **Information available:** list is displayed again with the **Topic?** prompt.

You can now choose another command or topic. You can abbreviate command words when you issue a command. If you know the name of the topic on which you want help, you can type this directly after you type **HELP**. The list of **HELP** topics is not displayed and the information you want is displayed directly.

To leave help, press Return repeatedly until the S7MP_C> prompt is displayed.

6.27 HISTORY

Format

HISTORY

Parameters

None.

Qualifiers

None.

Description

The HISTORY command displays the command history list with line numbers and waits for a line entry to recall the command. The history is stored in a file `/usr/var/ss7/.s7mp_history`

Example

```
S7MP_C> HISTORY
```

Prompt for the id of the command to redo:

```
1: show fep * /stat1
2: show fep * /stat2
3: show fep * /stat4
4: show fep * /stat5
5: show fep * /stat6
6: show fep * /stat7
7: show fep * /stat3
8: show fep * /stat10
9: creat bep *
10: enab bep *
11: conn bep *
12: sav bep *
13: show fep * /stat8
14: do app_loop_krest_a.s7mp
15: show f * mtp2 t */all
16: show f * mtp3 l */all
17: history
id ?
```

6.28 INHIBIT (ANSI and ITU-T only)

Format

INHIBIT *link_entity_name*

Parameters

link_entity_name

The name of a LINK entity

Qualifiers

None.

Description

Inhibits a link.

This command supports the use of wildcards.

Example

```
S7MP_C> INHIBIT FEP_SUBSYSTEM PILOTE MTP3 LINKSET 1 LINK 1
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
Entity Name: FEP_SUBSYSTEM PILOTE MTP3 LINKSET 1 LINK 1
Timestamp: 5-MAR-2003 14:50:25.14
```

Example

```
S7MP_C> INHIBIT FEP_SUBSYSTEM PILOTE MTP3 LINKSET 2 LINK 1
```

Produces the following result if the LINK entity is in a disabled state:

```
@SS7-E-NOTDONE, action not performed
Entity Name: FEP_SUBSYSTEM PILOTE MTP3 LINKSET 2 LINK 1
Timestamp: 8-MAR-2003 15:51:48.87
Action Status: SS7_K_STATE_INVALID
Reason: SS7_K_LINK_DISABLED
```

Example

```
S7MP_C> INHIBIT FEP_SUBSYSTEM PILOTE MTP3 LINKSET 1 LINK 1
```

Produces the following result if the parent entity of the LINK does not exist:

```
@SS7-E-NOSUCHPARENT, no such parent
Class: LINKSET
Instance_id: 1
Timestamp: 5-MAR-2003 14:50:25.14
```

6.29 INIT_LINK

Format

```
INIT_LINK [data_link_entity_name][hssl_entity_name]
```

Parameters

data_link_entity_name

hssl_entity_name

The name of a DATA_LINK or HSSL entity.

Qualifiers

None.

Description

Initializes a DATA_LINK or HSSL entity.

This command supports the use of wildcards.

Example

```
S7MP_C> INIT_LINK FEP_SUBSYSTEM PILOTE MTP2 DATA_LINK 1
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
Entity Name: FEP_SUBSYSTEM PILOTE MTP2 DATA_LINK 1
Timestamp: 5-MAR-2003 14:50:25.14
```

Example

```
S7MP_C> INIT_LINK FEP_SUBSYSTEM PILOTE MTP2 DATA_LINK 1
```

Produces the following result if the DATA_LINK is already in service:

```
@SS7-E-NOTDONE, action not performed
Entity Name: FEP_SUBSYSTEM PILOTE MTP2 DATA_LINK 1
Timestamp: 8-MAR-2003 15:51:50.59
Action Status: SS7_K_STATE_INVALID
Reason: SS7_K_DTLK_IN_SERVICE
```

Example.

```
S7MP_C> INIT_LINK FEP AMAMET SAAL HSSL 1
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
Entity Name: FEP_SUBSYSTEM AMEMET SAAL HSSL 1
Timestamp: 07 Feb-2002 15:33:21.00
```

6.30 LINKTEST

Format

LINKTEST *link_entity_name*

Parameters

link_entity_name

The name of a LINK entity.

Qualifiers

None.

Description

Bee-CUSP Convergent Signaling Platform operating guide

Tests a LINK.

This command supports the use of wildcards.

Example

```
S7MP_C> LINKTEST FEP_SUBSYSTEM PARMA MTP3 LINKSET 1 LINK 1
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
Entity Name: FEP_SUBSYSTEM PARMA MTP3 LINKSET 1 LINK 1
Timestamp: 5-MAR-2003 14:50:25.14
```

Example

```
S7MP_C> LINKTEST FEP_SUBSYSTEM PARMA MTP3 LINKSET 1 LINK 1
```

Produces output similar to the following if the LINK is in a failed state:

```
@SS7-E-NOTDONE, action not performed
Entity Name: FEP_SUBSYSTEM PARMA MTP3 LINKSET 1 LINK 1
Timestamp: 8-MAR-2003 15:51:50.59
Action Status: SS7_K_STATE_INVALID
Reason: SS7_K_LINK_FAILED
```

6.31 LOAD

Format

LOAD

Parameters

None.

Qualifiers

None.

Description

Dynamically loads the filter file which contains the filtering rules for the incoming user traffic interceptor on all FE processes. When using this command you do not need to stop and restart any of the FE processes.

You can only use this command to apply the filter file on **all** the FE processes (i.e. `FEP *`), you cannot use it with a specific FE process name.

Example

```
S7MP_C> LOAD FEP * MTP3
```

**Note**

You need to check the filtering log in `/usr/var/ss7/platform_<platform_id>/log/fep_<platform_id>.log` to ensure that the filter file was applied successfully. For more information, see the section on the **incoming user traffic interceptor**.

6.32 LOAD_STAND_BY

Format

LOAD_STAND_BY *entity_name*

Parameters**entity_name**

The name of the SCCP or NIF entity.

Qualifiers

None.

Description

Loads the GTT and RK standby tables with the contents of the GTT and RK standby file. No system message is displayed when the command has completed successfully.

Sample command for GTT rules

```
S7MP_C> LOAD_STAND_BY FEP_SUBSYSTEM * SCCP
```

Sample command for RK rules

```
S7MP_C> LOAD_STAND_BY FEP_SUBSYSTEM * NIF
```

6.33 MODIFY

Format

MODIFY *entity_name* /*list_of_characteristic_attributes_with_values*

Parameters**entity_name**

The name of an entity.

Qualifiers**list_of_characteristic_attributes_with_values**

A list of items whose format is: /qualifier=value

This command accepts as qualifiers the settable characteristic attributes of the entity to be modified. When attributes are "settable at creation" and "mandatory", they are not accepted as qualifiers for the `MODIFY` command.

Description

Modifies one or more characteristic attributes of an entity. Refer to the *Bee-CUSP CSP entity reference guide* for details of the characteristic attributes of all entities. This command supports the use of wildcards.

Example

```
S7MP_C> MODIFY FEP_SUBSYSTEM XGATES MTP3 /T21=10000
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
Entity Name: FEP_SUBSYSTEM XGATES MTP3
Timestamp: 1-MAR-2003 15:23:32.05
T21 (T21): SS7_K_SUCCESS 10000
```

Example

```
S7MP_C> MODIFY FEP_SUBSYSTEM XGATES MTP3 /T1=49/T2=69/T3=120
```

Produces the following result because the values for T1 and T2 are out of the permitted range:

```
@SS7-E-MODIFYERROR, modify error
Entity Name: FEP_SUBSYSTEM XGATES MTP3
Timestamp: 1-MAR-2003 15:24:48.32
T1 (T1): SS7_K_INVALID_ATT_VAL 49
T2 (T2): SS7_K_INVALID_ATT_VAL 69
T3 (T3): SS7_K_SUCCESS 120
```

Example

```
S7MP_C> MODIFY FEP_SUBSYSTEM XGATES MTP3 /NETINDIC=3
```

Produces the following result if MTP3 is enabled because NETINDIC is a static characteristic:

```
@SS7-E-MODIFYERROR, modify error
Entity Name: FEP_SUBSYSTEM XGATES MTP3
Timestamp: 1-MAR-2003 15:22:46.88
Network Indicator (NETINDIC): SS7_K_STATIC_CHAR 3
```

6.34 RECONFIGURE

Format

RECONFIGURE

RECONFIGURE *entity_name*

Parameters

FEP *entity_name*

Reconfigures the specified FE process, if it is not connected.

FEP *

Reconfigures all the non-connected FE processes.

BEP entity_name

Reconfigures the specified BEP and its local applications as much as it needs to be reconfigured.

No parameter

Reconfigures all the FE processes on the platform that need to be reconfigured (global reconfiguration.)

The result is equivalent to:

RECONFIGURE FEP*

and

RECONFIGURE BEP1

RECONFIGURE BEP2

RECONFIGURE BEP3 ...

RECONFIGURE BEPn

Qualifiers

None.

Description

To perform this action you must have previously carried out a `SAVE_CONFIG` action, except where you want to reconfigure one or several FE processes and there is still a connected FE process on the platform.

For FE or BE process reconfiguration and global reconfiguration, you must issue the `RECONFIGURE` command with a management application that is on the same `DIRECTOR` as the one on which you issued the `SAVE_CONFIG` command.

Restrictions

There are two situations where the previous state will not be restored by a `RECONFIGURE` command:

- The `SYSFAIL` event on the FEP entities.
- The values of the following TRUNK entity characteristics have been modified and are different from the default values:

`SS7_K_TRNK_CGRBONSET`

`SS7_K_TRNK_CGRBABAT`

To restore the event on the FEP entities you must use the `ENAEVT` command.

To restore the values of `SS7_K_TRNK_CGRBONSET` and `SS7_K_TRNK_CGRBABAT` you must use the `MODIFY` command after you have issued the `RECONFIGURE` command.

This command does **not** support the use of wildcards.

Example

```
S7MP_C> RECONFIGURE FEP *
```

Reconfigures all the non-connected FE processes of the platform.

Example

```
S7MP_C> RECONFIGURE FEP PARMA
```

Bee-CUSP Convergent Signaling Platform operating guide

Reconfigures the FE process PARMA if it is not connected. If the FE process PARMA is connected, will produce the result:

```
%SS7-E-NOTDONE, Action not performed
Entity Name: FEP_SUBSYSTEM PARMA
Timestamp: 11-FEB-2000 14:58:09.05
Action Status: SS7_K_CONTEXT_INVALID
Reason: SS7_K_LEVEL_ALREADY_CONFIGURED
```

Example

```
S7MP_C> reconf fep CHIMAY
```

Will produce the following result if the FEP_SUBSYSTEM already exists:

```
@SS7-W-ENTITYEXISTS, Entity already exists
Entity Name: FEP_SUBSYSTEM CHIMAY
Timestamp: 28-Jan-2000 10:50:23.00
```

Example

```
S7MP_C> RECONFIGURE
```

Will reconfigure as much of the platform as can be reconfigured from the files previously saved.

Example

```
S7MP_C>RECONFIGURE BEP PARMA
```

Reconfigures BEP PARMA and the local applications defined on BEP PARMA which need to be reconfigured. If the BEP is already connected and no local applications were running, this command will return an error.

```
S7MP_A> reconf bep parma
%SS7-E-NOTDONE, Action not performed
Entity Name : BEP_SUBSYSTEM PARMA
Timestamp : 11-FEB-2000 14:58:09.05
Action Status : SS7_K_CONTEXT_INVALID
Reason : SS7_K_LEVEL_ALREADY_CONFIGURED
S7MP_A>
```

```
S7MP_C>RECONFIGURE BEP bep_name
```

The BEP may already be connected. If there are some applications to reconfigure, there will be no error and the command will reconfigure the local applications as much as possible, depending on what needs to be reconfigured and on what was saved with the SAVE_CONFIG command.

6.35 REMOVE

Format

REMOVE *entity_name /characteristic_attribute_values*

Parameters

entity_name

The name of a FEP (ALIAS_PC_LIST), an APPLICATION, SCCP, a combined LINKSET entity or DESTINATION (SUA) entity.

Qualifiers

SSNLIST=

The local concerned SSN list of an APPLICATION entity
The concerned SSN list of an SCCP entity

PCLIST=

The concerned PC list of an APPLICATION entity

CLINKSETLIST=

The member list of a combined LINKSET entity.

CPCLIST=

The list of PCs (no more than 100 PCs) to inform when the state of a DESTINATION entity changes.

ALIAS_PC_LIST=

List of alias point codes for an application entity.



Note

- The qualifiers shown can only be used with the ADD and REMOVE commands.
- Entities must be created before the REMOVE command can be used.
- The REMOVE command accepts more than one value per qualifier.

Description

Removes settable characteristic attribute values from a FEP, a combined LINKSET, a DESTINATION, an APPLICATION or SCCP entity. The format is /qualifier=(list_of_values).

You cannot remove more than 100 PCs at a time. If you want to remove 400 PCs for example, you must execute the REMOVE command four times. Refer to the *Bee-CUSP CSP entity reference guide* for details.

This command supports the use of wildcards.

Example

```
S7MP_C> REMOVE FEP_SUBSYSTEM PARMA SCCP APPLICATION 6 / PCLIST=(12,16383)
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
Entity Name: FEP_SUBSYSTEM PARMA SCCP APPLICATION 6
Timestamp : 5-MAR-1994 14:38:15.51
Concerned PC List :
SS7_K_SUCCESS:12 SS7_K_SUCCESS:16383
```

Example

```
S7MP_C> REMOVE FEP_SUBSYSTEM PARMA SCCP /SSNLIST=4
```

Bee-CUSP Convergent Signaling Platform operating guide

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
Entity Name: FEP_SUBSYSTEM PARMA SCCP
Timestamp : 5-MAR-1994 14:38:14.22
Local Concerned SSN :
SS7_K_SUCCESS: 4
```

Example

```
S7MP_C> REMOVE FEP_SUBSYSTEM BGATES SCCP /SSNLIST=(7,8,9)
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
Entity Name: FEP_SUBSYSTEM BGATES SCCP
Timestamp: 4-JUL-1994 13:09:19.33
Concerned SSN List (SSNLIST):
SS7_K_SUCCESS: 7
SS7_K_SUCCESS: 8
SS7_K_SUCCESS: 9
```

Example

```
S7MP_C> REMOVE FEP_SUBSYSTEM PARMA SCCP APPLICATION 6 / SSNLIST=(6,7,10,8)
```

Produces the following result if SSN 10 has already been removed from the SSNLIST:

```
@SS7-S-NORMAL, normal successful completion
Entity Name: FEP_SUBSYSTEM PARMA SCCP APPLICATION 6
Timestamp : 5-MAR-1994 14:38:16.38
Concerned SSN List (SSNLIST):
SS7_K_SUCCESS: 6
SS7_K_SUCCESS: 7
SS7_K_ALREADY_ABSENT: 10
SS7_K_SUCCESS: 8
```

Example

```
S7MP_C> REMOVE FEP_SUBSYSTEM PARMA SCCP APPLICATION 6 / SSNLIST=(1,2)
```

Produces the following result because the SSN value must be an integer between 2 and 254:

```
@SS7-E-ADDREMEROR, add or remove error
Entity Name: FEP_SUBSYSTEM PARMA SCCP APPLICATION 6
Timestamp : 5-MAR-1994 14:38:14.22
Concerned SSN List (SSNLIST):
SS7_K_OUT_OF_RANGE: 1
SS7_K_OUT_OF_RANGE: 2
```

6.36 REPEAT

Format

REPEAT

Parameters

None.

Qualifiers

None.

Description

The `REPEAT` command allow the repetition of a command by prompting for an occurrence, a period (in second) and the id of the command.

Example

In the following example, the command `SHOW f * mtp2 t */all`, id 15, will be repeated 10 times, with a delay of 2 seconds

```
S7MP_C> REPEAT
```

Prompt for the occurrence, the delay and the id of the command:

```
id ?occurrence [0=infinite]? 2
delay [default=1]? 10
1: show fep * /stat1
2: show fep * /stat2
3: show fep * /stat4
4: show fep * /stat5
5: show fep * /stat6
6: show fep * /stat7
7: show fep * /stat3
8: show fep * /stat10
9: creat bep *
10: enab bep *
11: conn bep *
12: sav bep *
13: show fep * /stat8
14: do app_loop_krest_a.s7mp
15: show f * mtp2 t */all
16: show f * mtp3 l */all
17: history
18: repeat
id ? 15
```

6.37 RK_RULE_SHALL

Format

`RK_RULE_SHALL /RuleType /StatusType`

Qualifiers**RuleType**

The type of rule you want to display, that is, `ROUTING` or `SCREENING` string. This qualifier is mandatory.

StatusType

The set of rules the rule belongs to, ACTIVE or STANDBY. If not specified, STANDBY is the default.

Description

This command displays all the NIF RK rule identifiers for SCREENING or ROUTING, contained either in the *active* or *standby* table (default mode).



Caution

The rule identifier is a unique identifier within its particular set of rules, i.e. either screening rules or routing rules. However, the same identifier can be used to identify a given rule in both rule sets, so a given screening rule can have the same identifier as a routing rule.

Example

```
S7MP_C> RK_RULE_SHALL /SCREENING /ACTIVE
```

Displays all NIF *screening* rule identifiers contained in the *active* table:

```
S7MP_C> RK_RULE_SHALL /SCREENING /ACTIVE
Screening Active registered rules are :
1
```

Example

```
S7MP_C> RK_RULE_SHALL /ROUTING
```

Displays all the NIF *routing* rule identifiers contained in the *standby* table:

```
S7MP_C> RK_RULE_SHALL /ROUTING
Routing Standby registered rules are :
1
2
3
```

Note



If no rule exists, the following message is displayed:

```
@NONAME-W-NOMSG, Message number is 20976562
```

6.38 RK_RULE_SHOW

Format

```
RK_RULE_SHOW RuleIdentifier /RuleType /StatusType
```

Parameters

RuleIdentifier

The identifier of the rule (RULEID) you want to show.

The rule identifier for a given rule is deduced from its position within the section in the rule provisioning file (ROUTING_SECTION or SCREENING_SECTION, identified by the qualifier). To retrieve the rule identifiers, you can use the RK_RULE_SHALL command (see RK_RULE_SHALL).

Qualifiers**RuleType**

The type of rule you want to display, that is, ROUTING or SCREENING string. This qualifier is mandatory.

StatusType

The set of rules the rule belongs to, ACTIVE or STANDBY. If not specified, STANDBY is the default.

Description

This command is used to display the contents of a NIF ROUTING or SCREENING rule, contained either in the *active* or in the *standby* table.

**Note**

The rule identifier is a unique identifier within its particular set of rules, i.e. either screening rules or routing rules. However, the same identifier can be used to identify a given rule in both rule sets, so a given screening rule can have the same identifier as a routing rule.

Example

```
S7MP_C> RK_RULE_SHOW 1 /SCREENING /STANDBY
```

Displays the following information for RuleIdentifier 1 in the standby table:

```
Rule :
  Rule Id                (RULEID) : 1
  Type Value              (TYPE) : Screening
  Set Value               (SET) : Standby
  Rule Text               (RULETXT) : opc=2;screen=1;
General Parameters :
  Forward msg to Value    (FORWARD MSG TO) : 1
MTP3 Pattern :
  Opc Value                                     (OPC) : 2
SCCP Calling Address Pattern :
  Calling Point Code Value                      (CG_PC) :
  Calling Subsystem Number Value                (CG_SSN) :
  Calling GTAI Value                            (CG_IGT) :
SCCP Called Address Pattern :
  Called Point Code Value                      (CD_PC) :
  Called Subsystem Number Value                (CD_SSN) :
  Input GT, Called GTAI Value                  (CD_IGT) :
TCAP Pattern :
  TCAP ITU-T Invoke Op Code Value              (IOP) :
  TCAP ANSI Invoke Op Code Value               (AOP) :
  TCAP ANSI Family Code Value                  (FAM) :
  TCAP ITU-T App Context Name Value            (ACN) :
Map/IS41 Pattern :
  GSM SMS Operation Value                      (SOP) :
  MAP Version Value                           (MV) :
  Service Center Originator Value              (SCO) :
  Service Center Recipient Value               (SCR) :
  IMSI Recipient Address Value                 (IMSIR) :
  SMS Recipient Value                          (SR) :
```

Bee-CUSP Convergent Signaling Platform operating guide

```
SMS Originator Value          (SO) :
SMS Protocol Identifier Value  (SI) :
Status :
State                          (STATE) : LOADED
```

Example

```
S7MP_C> rk_rule_show 1 /ACTIVE /ROUTING
```

Displays the following information for RuleIdentifier **1** in the **active** table:

```
Rule :
Rule Id                      (RULEID) : 1
Type Value                   (TYPE) : Routing
Set Value                    (SET) : Active
Rule Text                    (RULETXT) : cgpa="214";pc=9;ssn=9;dl3=12;
General Parameters :
Forward msg to Value         (FORWARD MSG TO) : dl3 12
SCCP Calling Address Pattern :
Calling Point Code Value     (CG_PC) :
Calling Subsystem Number Value (CG_SSN) :
Calling GTAI Value           (CG_IGT) : 214
SCCP Called Address Pattern :
Called Point Code Value      (CD_PC) : 9
Called Subsystem Number Value (CD_SSN) : 9
Input GT, Called GTAI Value  (CD_IGT) :
TCAP Pattern :
TCAP ITU-T Invoke Op Code Value (IOP) :
TCAP ANSI Invoke Op Code Value (AOP) :
TCAP ANSI Family Code Value (FAM) :
TCAP ITU-T App Context Name Value (ACN) :
Map/IS41 Pattern :
GSM SMS Operation Value      (SOP) :
MAP Version Value            (MV) :
Service Center Originator Value (SCO) :
Service Center Recipient Value (SCR) :
IMSI Recipient Address Value (IMSIR) :
SMS Recipient Value          (SR) :
SMS Originator Value          (SO) :
SMS Protocol Identifier Value  (SI) :
Substitution Pattern :
Output GTAI (Called Addr) Value (OGT) :
MAP Destination Address Value (SC) :
Status :
State                          (STATE) : LOADED
```

Example

```
S7MP_Y> rk_rule_show 10 /active/routing
```

Causes the following information to be displayed about the **active routing rule** number **10** that does not exist:

```
@SS7-E-NOSUCHINSTANCE, No such instance
```

6.39 SAVE_CONFIG

Format

SAVE_CONFIG *entity_name*

SAVE_CONFIG

Parameters

FE process **entity_name**

Saves the configuration of one of the connected FE processes.

FEP *

Saves the configuration of all the connected FE processes.

BEP entity_name

Saves the configuration of the specified BEP and its local applications.

SAVE_CONFIG

The SAVE_CONFIG command can also be issued without a parameter. Saves all running FE processes, BE processes and local user applications on the platform (global save).

The result is equivalent to:

```
SAVE_CONFIG FEP*
and
SAVE_CONFIG BEP1
SAVE_CONFIG BEP2
SAVE_CONFIG BEP3 . . . .
SAVE_CONFIG BEPn
```

Qualifiers

None.

Description

Saving FE process configurations

You can only save the FE process configuration if there is still a connected FE process on the platform. The FE process configuration is saved in a file and the file is sent to the Director, where the file is stored.

Saved FE process file

This file can be used later to configure one or several new FE processes, or to reconfigure some failed FE processes that are restarted. (See the RECONFIGURE command.)

Saving BEP configurations

You cannot save the local BEP configuration and its local application entities unless the BEP is connected.

Saved BEP file

Bee-CUSP Convergent Signaling Platform operating guide

This file can be used to configure failed BEP and application copies. See the **RECONFIGURE** command.



Caution

If these files are modified it is not guaranteed that Bee-CUSP CSP reconfiguration will operate correctly. It is the responsibility of the client to ensure correct operation of the procedure if these files are modified.

The **SAVE_CONFIG** command with no parameters (global save)

You can save the entire platform according to the connected state of the FE processes and BE processes. The resulting files can be used later to reconfigure the FE processes, BE processes and local applications depending on the information that was saved. See **RECONFIGURE**.

Restrictions

There are two situations where the previous state will not be restored by a subsequent `AutoReconfigure`:

- The `SYSFAIL` event on the FE process and BEP entities.
- The values of the TRUNK entity characteristics:

`SS7_K_TRNK_CGRBONSET`

`SS7_K_TRNK_CGRBABAT`

- If these values have been modified and are different from the default values. See **RECONFIGURE**.

This command does **not** support the use of wildcards.

Example

```
S7MP_C> SAVE_CONFIG FEP_SUBSYSTEM *
```

Saves the configuration of all connected FE processes. If no FE process is connected, an error is returned:

```
%SS7-E-NOTDONE, Action not performed
Entity Name: FEP_SUBSYSTEM *
Timestamp: 21-FEB-2000 16:11:06.00
Action Status: SS7_K_CONTEXT_INVALID
Reason: SS7_K_LEVEL_NOT_READY
```

Example

```
S7MP_C> SAVE_CONFIG FEP_SUBSYSTEM PARMA
```

Saves the configuration of the connected FE process specified in the command, in this case, PARMA.

Example

```
S7MP_C> SAVE_CONFIG
```

If you issue the `SAVE_CONFIG` command, without a parameter, as in this example, the configuration of entire platform is saved.

Example

```
S7MP_C> SAVE_CONFIG BEP PARMA
```

If the BEP is not connected will return an error:


```

S7MP_C> save_config bep *
%SS7-E-NOTDONE, Action not performed
Entity Name : BEP_SUBSYSTEM PARMA
Timestamp : 21-FEB-2000 16:11:06.00
Action Status : SS7_K_CONTEXT_INVALID
Reason : SS7_K_LEVEL_NOT_READY
S7MP_C>

```

6.40 SENDMSG

Format

SENDMSG *application_entity_name* /*qualifiers*

Parameters

application_entity_name

The name of an application entity.

Qualifiers

REQ_NCOORD

Sends an N_Coord_Request.

RSP_NCOORD

Sends an N_Coord_Response.

NSTATE_REQ

Sends an N_State_Request.

USER_STATUS {IN_SERVICE, OUT_OF_SERVICE}

Specifies the user status.

PC_AFFECTED = number

Specifies the PC to which a N_Coord_Request or a N_Coord_Response must be sent.

Description

Sends an SCCP management message. Refer to the *Entity reference guide* and the *Application developer's guide* for further details.



Note

1. Only the following qualifier combinations are allowed:
 - /REQ_NCOORD /PC_AFFECTED=XX
 - /RSP_NCOORD /PC_AFFECTED=XX
 - /NSTATE_REQ /USER_STATUS=IN_SERVICE, or OUT_OF_SERVICE

where XX = the PC to which the SOR or SOG is to be sent.

2. You cannot abbreviate the value `IN_SERVICE` for the qualifier `USER_STATUS`. If you enter the abbreviation `IN_SER`, it is ignored and the value `OUT_OF_SERVICE` is taken instead.
3. Even if the `SENDMSG` request is sent as a distributed request, only one FE processes the request. In this case, the management application receives only one response.

This command supports the use of wildcards.

Example

```
S7MP_C>SENDMSG FEP_SUBSYSTEM PARMA SCCP APPLICATION 6 -  
_S7MP_C>/NSTATE_REQ/USER_STATUS=IN_SERVICE
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion  
Entity Name: FEP_SUBSYSTEM PARMA SCCP APPLICATION 6  
Timestamp : 5-MAR-2001 14:50:24.52
```

Example

```
S7MP_C>SENDMSG FEP_SUBSYSTEM PARMA SCCP APPLICATION 6 -  
_S7MP_C>/NCOORD_REQ /PC_AFFECTED=6
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion  
Entity Name: FEP_SUBSYSTEM PARMA SCCP APPLICATION 6  
Timestamp : 5-MAR-2001 14:50:24.52
```

Example

```
S7MP_C>SENDMSG FEP_SUBSYSTEM PARMA SCCP APPLICATION 6 -  
_S7MP_C>/NCOORD_RSP /USER_STATUS=IN_SERVICE /PC_AFFECTED=6
```

Produces the following result because the `USER_STATUS` qualifier is invalid for an `NCOORD_RSP` primitive:

```
@SS7-E-INVSCCPPRIM, invalid SCCP primitive  
Entity Name: FEP_SUBSYSTEM PARMA SCCP APPLICATION 6  
Timestamp : 5-MAR-2001 14:50:24.52
```

6.41 SET

Format

SET *timestamp* /*qualifiers*

SET *mode* /*qualifiers*

Parameters

timestamp

mode

Qualifiers

SET TIMESTAMP qualifiers:

RESPONSE={ON, OFF}

EVENT={ON, OFF}

PROC_STATUS_INDIC={ON, OFF}

SET MODE qualifiers:

VERBOSE={ON, OFF}

CONSOLIDATION={ON, OFF}

SHOWWILD={ON, OFF}

Description

- SET TIMESTAMP

Sets the timestamp display associated with responses, events and process status indications to ON or OFF.

Note

This command does not have a response. Command responses received when this command is entered have no timestamp if you have set RESPONSE=OFF.

- SET MODE/VERBOSE

Sets the verbose mode. It is also possible to set the verbose mode using an environment variable.

Note

If the verbose mode is OFF, only the responses for which the response code is different from **NORMAL** are displayed, with the exception of the SHOW and SHRESET commands.

- SET MODE/CONSOLIDATION

The responses to a distributed request are consolidated in this mode.

- Consolidation mode ON:

The responses from different FE processes to a distributed request are consolidated in order to display one response answer.

If the responses are consistent only the consolidated response is displayed.

If the responses are inconsistent, the message:

%SS7-E-INCONSANSWER

is displayed, as well as the response of each FE process.

If the responses are consistent, but at least one FE process has not responded, the message:

%SS7-E-NORMALCONSOWITHNR

is displayed, as well as the response of each FE process.

- If consolidation mode is OFF:

The responses are not consolidated, each response is displayed.

- If the consolidation mode is BOTH:

The consolidated response is displayed only if the responses are consistent. Individual responses are also displayed.

- If the consolidation mode is NZERO:

The response displays only the value not equal to 0.

- If the consolidation mode is `ZERO`: The response displays all the value. This mode is the default display.
- **SET MODE/SHOWWILD**
When set to `SHOWWILD=ON`, allows the use of wild-carding with the `SHOW` and `SHRESET` commands. `SHOWWILD=ON` is the default setting and is set to `ON` when `S7MP` is started.

When `SHOWWILD` is set to `ON`, a `SHOW` command, or a `SHRESET` command targeting a non-wildcarded entity name is mapped to the `ss7_mgt_show` routine. When set to `SHOWWILD=OFF`, wild-carding is not allowed with the `SHOW` and the `SHRESET` commands.

When `SHOWWILD` is set to `OFF`, a `SHOW` command, or `SHRESET` command targeting a wildcarded entity name, returns an `SS7_WILDNOTSUPPORTED` response code. A `SHOW` or a `SHRESET` command targeting a non-wildcarded entity name is mapped to the `ss7_mgt_show` routine.

SET timestamp examples



Note

When `S7MP` is started the timestamp display is set to `ON`.

1.

```
S7MP_C> SET TIMESTAMP /RESPONSE=OFF
S7MP_C> SHOW FEP_SUBSYSTEM PARMA MTP3 LINKSET 1 LINK 1 /ALL
```

Produces a result without a timestamp

2. The environment variable `S7MP_PROC_STATUS_INDIC` is set to `ON`. Process status indications are displayed.

3. A timestamp is not required with process status indications, therefore the `SET TIMESTAMP` qualifier option must be set to `OFF`.

```
S7MP_C> SET TIMESTAMP/PROC_STATUS_INDIC=OFF
```

SET mode examples

Example

```
S7MP_C> SET MODE /CONSO=ON

S7MP_C> CREATE FEP_SUBSYSTEM * MTP3 LINKSET 4 /ADJPC=3 /RULEID=0
```

Produces this type of response:

```
@SS7-S-NORMALCONSO, normal successful consolidation
@SS7-S-NORMAL, normal successful completion
Entity Name:                FEP_SUBSYSTEM * MTP3 LINKSET 4
Timestamp:                  12-SEP-2002 14:17:56.20
Adjacent Point Code (ADJPC): SS7_K_SUCCESS 3
Rule Identifier (RULEID):    SS7_K_SUCCESS 0
```

Example

```
S7MP_C> SET MODE /CONSO=OFF
```

```
S7MP_C> CREATE FEP_SUBSYSTEM * MTP3 LINKSET 4 /ADJPC=3 /RULEID=0
```

Produces this type of response:

```
@SS7-S-NORMAL, normal successful completion
```

```
Entity Name:          FEP_SUBSYSTEM GIRELE MTP3 LINKSET 4
Timestamp:            12-SEP-1996 14:41:06.47
Adjacent Point Code (ADJPC): SS7_K_SUCCESS 3
Rule Identifier (RULEID):  SS7_K_SUCCESS 0
```

```
@SS7-S-NORMAL, normal successful completion
```

```
Entity Name:          FEP_SUBSYSTEM REQUIN MTP3 LINKSET 4
Timestamp:            12-SEP-1996 14:41:06.47
Adjacent Point Code (ADJPC): SS7_K_SUCCESS 3
Rule Identifier (RULEID):  SS7_K_SUCCESS 0
```

Example

```
S7MP_C> SET MODE /CONSO=BOTH
```

```
S7MP_C> SHOW FEP * /COUNTER /STATUS /IDENTIFIER
```

Produces this type of response:

```
@SS7-S-NORMALCONSO, normal successful consolidation
```

```
@SS7-S-NORMAL, normal successful completion
```

```
Entity Name:          FEP_SUBSYSTEM *
Timestamp:            12-SEP-2002 14:48:06.95
Attribute Group:      Identifiers
Name:                 SS7_K_NO_CONSO
Attribute Group:      Status
State:                SS7_K_SUBS_IN_SERVICE
Attribute Group:      Counters
FEP creation time (CREATIME): SS7_K_NO_CONSO
Nb of software errors (SWNBERROR): 0
Msg discarded by CS (MSG_DISC): 0
Nb of free buffers (NBFREEBUFF): 3982
```

```
@SS7-S-NORMAL, normal successful completion
```

```
Entity Name:          FEP_SUBSYSTEM GIRELE
Timestamp:            12-SEP-2002 14:48:06.95
Attribute Group:      Identifiers
Name:                 FEP_SUBSYSTEM GIRELE
Attribute Group:      Status
State:                SS7_K_SUBS_IN_SERVICE
Attribute Group:      Counters
FEP creation time (CREATIME): 12-SEP-2002 14:30:11.67
Nb of software errors (SWNBERROR): 0
Msg discarded by CS (MSG_DISC): 0
Nb of free buffers (NBFREEBUFF): 1982
```

@SS7-S-NORMAL, normal successful completion

Entity Name:	FEP_SUBSYSTEM REQUIN
Timestamp:	12-SEP-2002 14:48:06.95
Attribute Group:	Identifiers
Name:	FEP_SUBSYSTEM REQUIN
Attribute Group:	Status
State:	SS7_K_SUBS_IN_SERVICE
Attribute Group:	Counters
FEP creation time (CREATIME):	12-SEP-2002 14:30:11.68
Nb of software errors (SWNBERROR):	0
Msg discarded by CS (MSG_DISC):	0
Nb of free buffers (NBFREEBUFF):	2000

Example

```
S7MP_C> set mode /conso=zero
S7MP_C> show f * MTP2 t 1/count
```

Produces this type of response:

FEP target list :	KASROL
Number of FEP target :	1
Timestamp :	08-Jan-2010 17:15:38.27
Timestamp : 08-Jan-2010 17:15:38.27	
Attribute Group : Counters	
Trunk Creation Time (CREATIME) :	SS7_K_NO_CONSO
Connection timeout (CNTTIMEOUT) :	0
Connection loss (CNTLOSSES) :	0
Receive carrier losses (RCVCRLOSSES) :	1
Receive synchro losses (RCVSYNLOSSES) :	0
AIS transmitted (AISTRSM) :	0
AIS received (AISRCV) :	0
Alarms from remote end (ALRMREMEND) :	0
Code violation (CODVIOL) :	0
CRC errors (CRCERR) :	0
Error Free Seconds (EFS) :	0
Errored Seconds (ES) :	14695
Severely Errored Seconds (SES) :	0
Consecutive Sever. Errored Sec. (CSES) :	0
Critical alarms (CRITICALARM) :	1
Elastic store slip occurrences (SLIPS) :	0
Far End Block Errors (ElFEBE) :	0
Rdir Critical Alarms (RDIR_CRITICALARM) :	0
Trunk Reset Time (RESETIME) :	08-Jan-2010 13:10:44.30 14694 Seconds

6.42 SHNEXT

Format

SHNEXT *entity_name/qualifiers*

Parameters

entity_name

The name of an entity. The *entity_name* parameter is comprised of the different elements of the path (from the root) in the hierarchy tree required to reach the target entity class and instance.

Qualifiers**list_of_characteristic_attributes****list_of_event_attributes****ALL**

Displays all the attributes of the entity *next* to the entity specified by the *entity_name* parameter.

STATUS

Displays the attributes of the group STATUS for the entity *next* to the entity specified by the *entity_name* parameter.

CHARACTERISTICS

Displays the attributes of the group CHARACTERISTICS for the entity *next* to the entity specified by the *entity_name* parameter.

COUNTERS

Displays the attributes of the group COUNTERS for the entity *next* to the entity specified by the *entity_name* parameter.

EVENTS

Displays the attributes of the group EVENTS for the entity *next* to the entity specified by the *entity_name* parameter.

IDENTIFIERS

Displays the attributes of the group IDENTIFIERS for the entity *next* to the entity specified by the *entity_name* parameter. By using the appropriate qualifier, the command can be used to show a single group of attributes for the entity next to the entity specified by the *entity_name* parameter, or to show individual characteristics or events (see the *Bee-CUSP CSP entity reference guide* for more details of characteristics and events).

You can specify more than one qualifier at a time, but you cannot use another qualifier at the same time as the /ALL qualifier. The /ALL qualifier shows all the attribute groups for the entity next to the entity specified by the *entity_name* parameter.

Description

This command allows information to be displayed for the entity considered to be *next* to the entity specified by the *entity_name* parameter. An entity is considered to be the next entity to the entity specified by the *entity_name* parameter if it is next in ascending order in the list of entities belonging the same class as the entity specified by the *entity_name* parameter.

If an entity exists which fulfills the following conditions, then it is the entity next to the *entity_name* specified as the parameter.

- The entity must belong to the same class as the entity specified by the *entity_name* parameter, that is, the same class as the last child entity.
- The entity must have the same parent entity instance ID value as the entity specified by the *entity_name* parameter, or have the next highest parent entity instance ID value after the entity specified by the *entity_name* parameter, but priority is always given to the same parent entity instance ID value.

Bee-CUSP Convergent Signaling Platform operating guide

- If the entity has the same parent entity instance ID value, the value of the entity's instance ID is the next highest value after the entity identified by the parameter *entity_name*.
If the entity has the next highest parent entity instance ID value, the value of the entity's instance ID must be the lowest instance ID value of the parent entity's child entities.

The entity that is the last in the list (in ascending order) of all entities belonging to the entity specified by the *entity_name* parameter is the last entity.

- If the targeted entity is the *last* entity, or is next to the last entity, then there is no *next* entity.
- If there is no *next* entity, the response SS7_NOOTHERINSTANCE is returned.

You can use specific instance ID values, or you can use wildcards in the *entity_name* parameter to find the first instance of a class.

Where a target entity contains wildcarded instance IDs, the attribute values returned are those of the lowest instance ID of each wildcarded class, if this entity exists. This command is useful to find the very first instance of a given class.

Examples

In the following examples, we assume that the following entities have been created:

```
FEP_SUBSYSTEM * MTP3 LINKSET 0 LINK 1
FEP_SUBSYSTEM * MTP3 LINKSET 0 LINK 2
FEP_SUBSYSTEM * MTP3 LINKSET 1 LINK 1
FEP_SUBSYSTEM * MTP3 LINKSET 1 LINK 2
FEP_SUBSYSTEM * MTP3 LINKSET 2 LINK 5
FEP_SUBSYSTEM * MTP3 LINKSET 2 LINK 8
```

Example

```
S7MP_C> SHNEXT FEP_SUBSYSTEM * MTP3 LINKSET * LINK */CHAR
```

As LINK 1 is the first LINK of the first LINKSET (LINKSET 0), LINK 1 characteristics are displayed, for example:

FEP target list:	DKART
Number of FEP target:	1
Entity Name:	FEP_SUBSYSTEM * MTP3 LINKSET 0 LINK 1
Attribute Group:	Characteristics
Physical Link Number (PLN):	1
FEP Name (FEPNAME):	DKART
Link Rank (RANK):	0
TB congestion level (TBCONGLEVEL):	0
Periodic SLT procedure used (SLTUSE):	NO_PERSLT

Example

```
S7MP_C> SHNEXT FEP_SUBSYSTEM * MTP3 LINKSET 0 LINK 1/CHAR
```

As LINK 2 is the next LINK after LINK 1 in LINKSET 0 the following information is displayed:

FEP target list:	DKART
Number of FEP target:	1
Entity Name:	FEP_SUBSYSTEM * MTP3 LINKSET 0 LINK 2
Attribute Group:	Characteristics
Physical Link Number (PLN):	2


```

FEP Name (FEPNAME):          CHAR
Link Rank (RANK):             1
TB congestion level (TBCONGLEVEL): 0
Periodic SLT procedure used (SLTUSE): NO_PERSLT
S7MP_C>

```

Example

```
S7MP_C> SHNEXT FEP_SUBSYSTEM * MTP3 LINKSET 0 LINK 2/CHAR
```

LINKSET 0 comprises only two LINKS, therefore LINK 1 of LINKSET 1 is the next entity, and the following information is displayed:

```

FEP target list:              DKART
Number of FEP target:        1
Entity Name:                  FEP_SUBSYSTEM * MTP3 LINKSET 1 LINK 1
Attribute Group:              Characteristics
Physical Link Number (PLN):   3
FEP Name (FEPNAME):          DKART
Link Rank (RANK):             2
TB congestion level (TBCONGLEVEL): 0
Periodic SLT procedure used (SLTUSE): NO_PERSLT
S7MP_C>

```

Example

```
S7MP_C> SHNEXT FEP_SUBSYSTEM MTP3 LINKSET 1 LINK 1/CHAR
```

LINK 2 is the next LINK to LINK1 in LINKSET 1, therefore the following information is displayed:

```

FEP target list:              DKART
Number of FEP target:        1
Entity Name:                  FEP_SUBSYSTEM * MTP3 LINKSET 1 LINK 2
Attribute Group:              Characteristics
Physical Link Number (PLN):   4
FEP Name (FEPNAME):          CHAR
Link Rank (RANK):             3
TB congestion level (TBCONGLEVEL): 0
Periodic SLT procedure used (SLTUSE): NO_PERSLT
S7MP_C>

```

Example

```
S7MP_C> SHNEXT FEP_SUBSYSTEM * MTP3 LINKSET 1 LINK 2/CHAR
```

LINKSET 1 comprises only two LINKS, therefore LINK 5 of the next LINKSET (LINKSET 2) is the next entity. The following information is displayed:

```

FEP target list:              DKART
Number of FEP target:        1
Entity Name:                  FEP_SUBSYSTEM * MTP3 LINKSET 2 LINK 5
Attribute Group:              Characteristics
Physical Link Number (PLN):   5
FEP Name (FEPNAME):          DKART
Link Rank (RANK):             4
TB congestion level (TBCONGLEVEL): 0

```

```
Periodic SLT procedure used (SLTUSE): NO_PERSLT
S7MP_C>
```

Example

```
S7MP_C> SHNEXT FEP_SUBSYSTEM * MTP3 LINKSET 2 LINK 5/CHAR
```

LINKSET 2 comprises two LINKS (5 and 8), therefore LINK 8 is the next entity after LINK 5. The following information is displayed:

FEP target list:	DKART
Number of FEP target:	1
Entity Name:	FEP_SUBSYSTEM * MTP3 LINKSET 2 LINK 8
Attribute Group:	Characteristics
Physical Link Number (PLN):	6
FEP Name (FEPNAME):	CHAR
Link Rank (RANK):	5
TB congestion level (TBCONGLEVEL):	0
Periodic SLT procedure used (SLTUSE):	NO_PERSLT
S7MP_C>	

6.43 SHNRESET

Format

SHNRESET *entity_name/qualifiers*

Parameters

entity_name

The name of an entity. The *entity_name* parameter is comprised of the different elements of the path (from the root) in the hierarchy tree required to arrive at the entity class.

Qualifiers

list_of_characteristic_attributes

list_of_event_attributes

ALL

Displays all the attributes of the entity *next* to the entity specified by the *entity_name* parameter.

STATUS

Displays the attributes of the group STATUS for the entity *next* to the entity specified by the *entity_name* parameter.

CHARACTERISTICS

Displays the attributes of the group CHARACTERISTICS for the entity *next* to the entity specified by the *entity_name* parameter.

COUNTERS

Displays the attributes of the group COUNTERS for the entity *next* to the entity specified by the *entity_name* parameter.

EVENTS

Displays the attributes of the group EVENTS for the entity *next* to the entity specified by the *entity_name* parameter.

IDENTIFIERS

Displays the attributes of the group IDENTIFIERS for the entity *next* to the entity specified by the *entity_name* parameter. By using the appropriate qualifier, the command can be used to show a single group of attributes for the entity next to the entity specified by the *entity_name* parameter, or to show individual characteristics or events (see the *Bee-CUSP CSP entity reference guide* for more details of characteristics and events).

You can specify more than one qualifier at a time, but you cannot use another qualifier at the same time as the /ALL qualifier. The /ALL qualifier shows all the attribute groups for the entity next to the entity specified by the *entity_name* parameter.

Description

This command allows information to be displayed for the entity considered to be *next* to the entity specified by the *entity_name* parameter and then resets the counters to (zero) 0 for this entity. The command provides the same functions and behaves in the same way as the SHNEXT command.

Example

```
S7MP_C> SHNRESET FEP_SUBSYSTEM * MTP3 LINKSET 2 LINK 1/STATUS
```

Displays the STATUS attributes for LINK 2 of LINKSET 2 and then resets the counters to 0 for this entity.

Example

```
S7MP_C> SHNRESET FEF_SUBSYSTEM * MTP3 LINKSET * LINK */STATUS
```

As LINK 1 is the first LINK of the first LINKSET (LINKSET 0), displays the STATUS attributes for LINK 1 of LINKSET 0 and then resets the counters to 0 for this entity.

6.44 SHOW

Format

SHOW *entity_name* /*qualifiers*

Parameters**entity_name**

The name of an entity.

Qualifiers**list_of_characteristic_attributes****list_of_event_attributes****ALL**

Displays all the attributes of the entity.

STATUS

Displays all the attributes of the group STATUS.

CHARACTERISTICS

Displays all attributes of the group CHARACTERISTICS.

COUNTERS

Displays all attributes of the group COUNTERS.

EVENTS

Displays all attributes of the group EVENTS.

IDENTIFIERS

Displays all attributes of the group IDENTIFIERS.

This command accepts the /ALL qualifier, which shows all the attribute groups for the specified entity. The command can also be used to show a single group of attributes using the appropriate qualifier, or to show individual characteristics or events. You can combine the qualifiers, but if you use the qualifier /ALL, you cannot use another qualifier at the same time.

Description

Shows entity attributes. Refer to the *Bee-CUSP CSP entity reference guide* for further details.

This command supports the use of wildcards when SET MODE/SHOWWILD is set to ON (the default setting).

Depending on the consolidation mode (ZERO or NZERO), all the counters are displayed or only the counters not equal to 0.

See SET, SET MODE/SHOWWILD for more information.

Example

```
S7MP_C> SHOW FEP_SUBSYSTEM CHIMAY MTP2/ALL
```

Shows all the attributes related to the MTP2 entity located on all the FE process, and produces this type of response:

```
FEP target list                : FCKENO FCMICH FCPERT FCSOUL
Number of FEP target          : 4
Timestamp : 23-Feb-2010 11:55:53.87

Timestamp : 23-Feb-2010 11:55:53.87
Attribute Group : Identifiers
  MTP2 Name                : MTP2 0
Attribute Group : Characteristics
  Product Version          : V611
  DLk Tx buffer cg threshold (CGONSETTB) : 30
  DLk Tx buff abatement thres. (CGABATTB) : 50
  Report two duplicated FISUs (DUPLISU)   : OFF
Attribute Group : Status
  Mtp2 State                : SS7_K_MTP2_AVAILABLE
Attribute Group : Counters
  MTP2 Creation Time (CREATIME)           : SS7_K_NO_CONSO
  Nb Of software errors (SWNBERROR)       : 0
  MTP2 Reset Time (RESETIME)              : 14-Jan-1980 20:57:35.94 950194698 Sec
Attribute Group : Events
```

```

MTP2 Soft Failure (LEVELFAIL)          : SS7_K_EVT_DISABLED
MTP2 Soft Error (SWERROR)              : SS7_K_EVT_DISABLED

```

Example

```
S7MP_C> show f * mtp3 1 10 1 2/count
```

Produces this type of response:

```

FEP target list                        : FCKENO FCMICH FCPERT FCSOUL
Number of FEP target                  : 4
Timestamp : 23-Feb-2010 11:55:02.40

Timestamp : 23-Feb-2010 11:55:02.40
Attribute Group : Counters
LINK Creation Time (CREATIME)         : SS7_K_NO_CONSO
MSUs Received (MSURCV)                : 10506
SIF_SIO Received (SIFORCV)           : 1050522
Net. Test. & Maint. Msg Rcv (TMSGRCV) : 1
MSUs Invalid DPC Rcv (MRCVIDPC)       : 0
Terminating MSU Disc - Inv NI (MDINI) : 0
Terminating MSU Disc - Inv SIO (MDISIO) : 0
Terminating MSU Disc - Inv HC (MDIHC) : 0
MSUs Transmitted (MSUTRN)             : 11609
SIF_SIO Transmitted (SIFOTRN)        : 1184038
Net. Test. & Maint. Msg trans.(TMSGTRN) : 1
Automatic ChangeOver (NACHOVER)      : 0
Nb of unexpected message (NBRCVIPROT) : 0
Nb of UPU msg sent (UPUSNT)          : 0
Nb of UPU msg received (UPURCV)      : 0
MSUs Disc. in retrieval (MSUDISCRET) : 0
MSUs sent by STP (MSUROUTED)         : 0
MSUs rcvd from STP (MSUTOROUTE)      : 0
LINK Reset Time (RESETIME) : 14-Jan-1980 20:57:38.19 950194644 Seconds

```

Example

```
S7MP_C> show f * mtp3 1 10 1 2/count=(MSURCV,MSUTRN)
```

Shows only the counters MSURCV and MSUTRN of the MTP3 LINK entity and produces this type of response:

```

FEP target list                        : FCKENO FCPERT FCMICH FCSOUL
Number of FEP target                  : 4
Timestamp : 23-Feb-2010 11:55:19.70

Timestamp : 23-Feb-2010 11:55:19.70
Attribute Group : Counters
MSUs Received (MSURCV)                : 17428
MSUs Transmitted (MSUTRN)             : 19227

```

6.45 SHRESET

Format

SHRESET *entity_name/qualifiers*

Parameters

entity_name

The name of an entity.

Qualifiers

list_of_characteristic_attributes

list_of_event_attributes

ALL

Displays all the attributes of the entity

STATUS

Displays all the attributes of the group STATUS

CHARACTERISTICS

Displays all attributes of the group CHARACTERISTICS

COUNTERS

Displays all attributes of the group COUNTERS

EVENTS

Displays all attributes of the group EVENTS

IDENTIFIERS

Displays all attributes of the group IDENTIFIERS

This command accepts the /ALL qualifier, which shows all the attribute groups for the specified entity. The command can also be used to show a single group of attributes using the appropriate qualifier, or to show individual characteristics or events. You can combine qualifiers, but if you use the /ALL qualifier, you cannot use another qualifier at the same time.

Description

The **SHRESET** command shows entity attributes and then resets counters for that entity to (zero) 0.

This command supports the use of wildcards when **SET MODE/SHOWWILD** is set to ON (the default setting). (See **SET** for more information.)

Example

```
S7MP_C> shreset b bcingr t 32/count
```

The example below shows the result of this command on a BE process subsystem. The first command returns the value of the current counters, then reset them. The second command shows the counters reset:

```
Timestamp : 23-Feb-2010 12:06:05.42
Attribute Group : Counters
Creation Time (CREATIME) : 23-Feb-2010 11:51:29.30
Msg failed to be sent (NBFAILEDSENT) : 0
Rcv Msg disc for any reason (MSG_DISC) : 0
Msg discarded by CS (CSDISC) : 0
Simult. TCAP recovered active dialogs : 0
Invoke Received (NINVKRCV) : 26250
Invoke Sent (NINVKSNT) : 0
```

S7MP command descriptions

P_Abort Received (NPABORCV)	: 0
P_Abort Rcv:Unrec Msg Type (NABUMTRCV)	: 0
...	
U_Abort Received (NUABORCV)	: 0
U_Abort Sent (NUABOSNT)	: 0
Rej Rcv : Inv. Comp. Coding (NRJINCRCV)	: 0
Begin Received (NBEGINRCV)	: 26250
Begin Sent (NBEGINSNT)	: 0
Continue Received (NCONTRCV)	: 0
Continue Sent (NCONTSNT)	: 0
End Received (NENDRCV)	: 0
End Sent (NENDSNT)	: 26250
Result_Last Received (NRSLARCV)	: 0
Result_No_Last Received (NRSNLARCV)	: 0
Result_Last Sent (NRSLASNT)	: 26250
Result_No_Last Sent (NRSNLASNT)	: 0
...	
Simult. TCAP Init Dialogs (NTOTINIDIA)	: 0
Unidirectional Messages Sent (NUNUSNT)	: 0
Unidirectional Msg Received (NUNURCV)	: 0
Invoke_Last Sent (NILSNT)	: SS7_K_NOT_SUPPORTED
Invoke_No_Last Sent (NINLSNT)	: SS7_K_NOT_SUPPORTED
Handover sent (HNDOVSNT)	: SS7_K_NOT_SUPPORTED
Invoke_Last Received (NILRCV)	: SS7_K_NOT_SUPPORTED
Invoke_No_Last Received (NINLRCV)	: SS7_K_NOT_SUPPORTED
Reset Time (RESETIME)	: 23-Feb-2010 11:51:29.30 876 Secor

Example

S7MP_C> shreset b bcingr t 32/count

Timestamp : 23-Feb-2010 12:06:10.93	
Attribute Group : Counters	
Creation Time (CREATIME)	: 23-Feb-2010 11:51:29.30
Msg failed to be sent (NBFAILEDSNT)	: 0
Rcv Msg disc for any reason (MSG_DISC)	: 0
Msg discarded by CS (CSDISC)	: 0
Simult. TCAP recovered active dialogs	: 0
Invoke Received (NINVKRCV)	: 210
Invoke Sent (NINVKSNT)	: 0
P_Abort Received (NPABORCV)	: 0
P_Abort Rcv:Unrec Msg Type (NABUMTRCV)	: 0
...	
U_Abort Received (NUABORCV)	: 0
U_Abort Sent (NUABOSNT)	: 0
Rej Rcv : Inv. Comp. Coding (NRJINCRCV)	: 0
Begin Received (NBEGINRCV)	: 210
Begin Sent (NBEGINSNT)	: 0
Continue Received (NCONTRCV)	: 0
Continue Sent (NCONTSNT)	: 0
End Received (NENDRCV)	: 0
End Sent (NENDSNT)	: 210
Result_Last Received (NRSLARCV)	: 0
Result_No_Last Received (NRSNLARCV)	: 0
Result_Last Sent (NRSLASNT)	: 210
Result_No_Last Sent (NRSNLASNT)	: 0
...	

Bee-CUSP Convergent Signaling Platform operating guide

```
Simult. TCAP Init Dialogs (NTOTINIDIA) : 0
Unidirectional Messages Sent (NUNUSNT) : 0
Unidirectional Msg Received (NUNURCV) : 0
Invoke_Last Sent (NILSNT) : SS7_K_NOT_SUPPORTED
Invoke_No_Last Sent (NINLSNT) : SS7_K_NOT_SUPPORTED
Handover sent (HNDOVSNT) : SS7_K_NOT_SUPPORTED
Invoke_Last Received (NILRCV) : SS7_K_NOT_SUPPORTED
Invoke_No_Last Received (NINLRCV) : SS7_K_NOT_SUPPORTED
Reset Time (RESETIME) : 23-Feb-2010 12:06:05.41 5 Seconds
```

The current state helps to explain the next two examples

```
S7MP_C> show fep */count
```

```
FEP target list : VLZ13 VLZ16 VLZ15 VLZ14
Number of FEP target : 4
Timestamp : 14-Sep-2006 14:27:40.44
Attribute Group : Counters
FEP creation time (CREATIME) : SS7_K_NO_CONSO
Nb of software errors (SWNBERROR) : 0
Msg discarded by CS (MSG_DISC) : 0
Nb of free buffers (NBFREEBUFF) : 39984
FEP reset time (RESETIME) : 14-Sep-2006 14:07:57.49 1183 Seconds
```

```
S7MP_C> show fep vlz13/count
```

```
Timestamp : 14-Sep-2006 14:28:28.64
Attribute Group : Counters
FEP creation time (CREATIME) : 14-Sep-2006 14:07:57.49
Nb of software errors (SWNBERROR) : 0
Msg discarded by CS (MSG_DISC) : 0
Nb of free buffers (NBFREEBUFF) : 9996
FEP reset time (RESETIME) : 14-Sep-2006 14:07:57.49 1231 Seconds
```

```
S7MP_C> show fep vlz14/count
```

```
Timestamp : 14-Sep-2006 14:28:32.34
Attribute Group : Counters
FEP creation time (CREATIME) : 14-Sep-2006 14:07:57.51
Nb of software errors (SWNBERROR) : 0
Msg discarded by CS (MSG_DISC) : 0
Nb of free buffers (NBFREEBUFF) : 9996
FEP reset time (RESETIME) : 14-Sep-2006 14:07:57.51 1235 Seconds
```

```
S7MP_C> show fep vlz15/count
```

```
Timestamp : 14-Sep-2006 14:28:41.70
Attribute Group : Counters
FEP creation time (CREATIME) : 14-Sep-2006 14:07:57.47
Nb of software errors (SWNBERROR) : 0
Msg discarded by CS (MSG_DISC) : 0
Nb of free buffers (NBFREEBUFF) : 9996
FEP reset time (RESETIME) : 14-Sep-2006 14:07:57.47 1244 Seconds
```

```
S7MP_C> show fep vlz16/count
```

```
Timestamp : 14-Sep-2006 14:28:44.76
```


S7MP command descriptions

```
Attribute Group : Counters
FEP creation time (CREATIME) : 14-Sep-2006 14:07:57.49
Nb of software errors (SWNBERROR) : 0
Msg discarded by CS (MSG_DISC) : 0
Nb of free buffers (NBFREEBUFF) : 9996
FEP reset time (RESETIME) : 14-Sep-2006 14:07:57.49 1247 Sec
```

SHRESET using wildcarding

```
S7MP_C> shreset fep */count
```

```
FEP target list : VLZ13 VLZ14 VLZ15 VLZ16
Number of FEP target : 4
Timestamp : 14-Sep-2006 14:29:34.84
Attribute Group : Counters
FEP creation time (CREATIME) : SS7_K_NO_CONSO
Nb of software errors (SWNBERROR) : 0
Msg discarded by CS (MSG_DISC) : 0
Nb of free buffers (NBFREEBUFF) : 39984
FEP reset time (RESETIME) : 14-Sep-2006 14:07:57.49 1297 Sec
```

```
S7MP_C> show fep vlz13/count
```

```
Timestamp: 14-Sep-2006 14:29:44.90
Attribute Group: Counters
FEP creation time (CREATIME) : 14-Sep-2006 14:07:57.49
Nb of software errors (SWNBERROR) : 0
Msg discarded by CS (MSG_DISC) : 0
Nb of free buffers (NBFREEBUFF) : 9996
FEP reset time (RESETIME) : 14-Sep-2006 14:29:34.84 10 Second
```

```
S7MP_C> show fep vlz14/count
```

```
Timestamp : 14-Sep-2006 14:29:49.02
Attribute Group : Counters
FEP creation time (CREATIME) : 14-Sep-2006 14:07:57.51
Nb of software errors (SWNBERROR) : 0
Msg discarded by CS (MSG_DISC) : 0
Nb of free buffers (NBFREEBUFF) : 9996
FEP reset time (RESETIME) : 14-Sep-2006 14:29:34.85 15 Second
```

```
S7MP_C> show fep vlz15/count
```

```
Timestamp : 14-Sep-2006 14:29:51.27
Attribute Group : Counters
FEP creation time (CREATIME) : 14-Sep-2006 14:07:57.47
Nb of software errors (SWNBERROR) : 0
Msg discarded by CS (MSG_DISC) : 0
Nb of free buffers (NBFREEBUFF) : 9996
FEP reset time (RESETIME) : 14-Sep-2006 14:29:34.82 17 Second
```

```
S7MP_C> show fep vlz16/count
```

```
Timestamp : 14-Sep-2006 14:29:53.43
Attribute Group : Counters
FEP creation time (CREATIME) : 14-Sep-2006 14:07:57.49
Nb of software errors (SWNBERROR) : 0
Msg discarded by CS (MSG_DISC) : 0
```

Bee-CUSP Convergent Signaling Platform operating guide

```
Nb of free buffers (NBFREEBUFF) : 9996
FEP reset time (RESETIME) : 14-Sep-2006 14:29:34.84 19 Seconds

S7MP_C> show fep */count

FEP target list : VLZ13 VLZ14 VLZ15 VLZ16
Number of FEP target : 4
Timestamp : 14-Sep-2006 14:30:37.67
Attribute Group : Counters
FEP creation time (CREATIME) : SS7_K_NO_CONSO
Nb of software errors (SWNBERROR) : 0
Msg discarded by CS (MSG_DISC) : 0
Nb of free buffers (NBFREEBUFF) : 39984
FEP reset time (RESETIME) : 14-Sep-2006 14:29:34.84 63 Seconds
```

SHRESET without using wildcarding

```
S7MP_C> shreset fep vlz13/count

Timestamp : 14-Sep-2006 14:32:13.44
Attribute Group : Counters
FEP creation time (CREATIME) : 14-Sep-2006 14:07:57.49
Nb of software errors (SWNBERROR) : 0
Msg discarded by CS (MSG_DISC) : 0
Nb of free buffers (NBFREEBUFF) : 9996
FEP reset time (RESETIME) : 14-Sep-2006 14:29:34.84 159 Seconds

S7MP_C> show fep vlz13/count

Timestamp : 14-Sep-2006 14:32:17.60
Attribute Group : Counters
FEP creation time (CREATIME) : 14-Sep-2006 14:07:57.49
Nb of software errors (SWNBERROR) : 0
Msg discarded by CS (MSG_DISC) : 0
Nb of free buffers (NBFREEBUFF) : 9996
FEP reset time (RESETIME) : 14-Sep-2006 14:32:13.44 4 Seconds

S7MP_C> show fep vlz14/count

Timestamp : 14-Sep-2006 14:32:19.99
Attribute Group : Counters
FEP creation time (CREATIME) : 14-Sep-2006 14:07:57.51
Nb of software errors (SWNBERROR) : 0
Msg discarded by CS (MSG_DISC) : 0
Nb of free buffers (NBFREEBUFF) : 9996
FEP reset time (RESETIME) : 14-Sep-2006 14:29:34.85 165 Seconds

S7MP_C> show fep vlz15/count

Timestamp : 14-Sep-2006 14:32:22.22
Attribute Group : Counters
FEP creation time (CREATIME) : 14-Sep-2006 14:07:57.47
Nb of software errors (SWNBERROR) : 0
Msg discarded by CS (MSG_DISC) : 0
Nb of free buffers (NBFREEBUFF) : 9996
FEP reset time (RESETIME) : 14-Sep-2006 14:29:34.82 168 Seconds
```

```
S7MP_C> show fep vlz16/count
```

```
Timestamp : 14-Sep-2006 14:32:25.23
Attribute Group : Counters
FEP creation time (CREATIME) : 14-Sep-2006 14:07:57.49
Nb of software errors (SWNBERROR) : 0
Msg discarded by CS (MSG_DISC) : 0
Nb of free buffers (NBFREEBUFF) : 9996
FEP reset time (RESETIME) : 14-Sep-2006 14:29:34.84 171 Second
```

```
S7MP_C> show fep */count
```

```
FEP target list : VLZ13 VLZ15 VLZ14 VLZ16
Number of FEP target : 4
Timestamp : 14-Sep-2006 14:32:29.01
Attribute Group : Counters
FEP creation time (CREATIME) : SS7_K_NO_CONSO
Nb of software errors (SWNBERROR) : 0
Msg discarded by CS (MSG_DISC) : 0
Nb of free buffers (NBFREEBUFF) : 39984
FEP reset time (RESETIME) : 14-Sep-2006 14:30:14.49 135 Second
```

6.46 SPAWN

Format

SPAWN *[command]*

Parameters

COMMAND

Qualifiers

None.

Description

If no COMMAND is specified, the SPAWN command creates a subprocess without exiting from S7MP. The prompt for this subprocess is:

```
S7MP_SPAWNED_$
```

All the operating system functions are available at this prompt.

When you have finished using the subprocess and wish to return to S7MP, type EXIT. If a command is specified, S7MP executes the command, displays the result and returns to the S7MP prompt.

Example

```
S7MP_C> SPAWN ls /usr/local
```

StP	fvwm	lemacs	pnv
VRWeb	games	lemacs-19.10	ps2text
alpha-dec-osf3.2	gdb	lib	release_notes
app-defaults	ghostscript	lsof	sentinel

Bee-CUSP Convergent Signaling Platform operating guide

Executes the command `ls /usr/local` before continuing.

Example

```
S7MP_C> spa ls /var
```

Produces the following result:

```
account  cache  db      games  hp    local  log      mail      nis  preserve  run
agentx   crash  empty   gdm    lib   lock   lost+found  net-snmp  opt   racoon   spool
```

```
S7MP_C> spa date
```

Produces the following result:

```
Tue Feb 23 14:41:34 CET 2010
```

Example

6.47 SWITCH_TABLE

Format

`SWITCH_TABLE entity_name`

Parameters

entity_name

The name of the SCCP or NIF entity.

Qualifiers

None.

Description

Switches the active GTT or RK rule table with the standby rule table and the active rule file with the standby rule file. There is no system message when the command has completed successfully if you are operating in verbose mode.

This command does **not** support the use of wildcards.

Sample command for GTT rules

```
S7MP_C> SWITCH_TABLE FEP_SUBSYSTEM * SCCP
```

Sample command for RK rules

```
S7MP_C> SWITCH_TABLE FEP_SUBSYSTEM * NIF
```

6.48 TEST_GTT

Format

`TEST_GTT /qualifiers`

Parameters

None.

Qualifiers

PARTY_ADDRESS=<"list_of_bytes">

Party address contents to be translated.

ACTIVE

STANDBY

Description

This command is used for testing the GTT mechanism. The GTT engine attempts to match a rule in the active table with the GT contained in the PARTY_ADDRESS. If a GTT rule in the active table is matched with the PARTY_ADDRESS, the rule ID and the result of the translation are displayed.

Example

```
S7MP_C> TEST_GTT /PARTY_ADDRESS=88049001/ACTIVE
```

Produces the following result:

Rule matched:	10
Action:	
Distribution (T_DISTRIB):	LOADSHARING
Reserved National Bit (T_RNB):	1
Global Title Indicator (T_GTI):	3
Translation Type (T_TT):	2
Numbering Plan (T_NP):	2
Primary PC (T_PPC):	3
Secondary PC (T_SPC):	3
Primary SSN (T_PSSN):	8
Global Title Address Information (T_GTAI):	ABCD

6.49 TRACE

Format

TRACE *file_spec*

Parameters**file_spec**

The name you give the file where the trace information is placed.

Qualifiers

None.

Description

Creates a file where the user input commands and the responses to these commands are stored. You may give the file any name and extension that you wish. You can continue to specify input at the `S7MP>` prompt; the subsequent output is placed in the same file until a new trace file is created or the trace is closed.

Opening a trace automatically closes any previously opened trace. A trace can be also be closed using the `UNTRACE` command.



Note

The `TRACE` command does not trace output related to the `HELP` command.

Example

```
S7MP_C> TRACE TEST.TRACE
```

Activates S7MP trace mode.

All the commands and results are saved in the file `TEST.TRACE`.

6.50 UNINHIBIT

Format

`UNINHIBIT link_entity_name`

Parameters

link_entity_name

The name of a LINK entity.

Qualifiers

None.

Description

Uninhibits a link.

This command supports the use of wildcards.

Example

```
S7MP_C> UNINHIBIT FEP_SUBSYSTEM PILOTE MTP3 LINKSET 1 LINK 1
```

Produces the following result:

```
@SS7-S-NORMAL, normal successful completion
Entity Name:      FEP_SUBSYSTEM PILOTE MTP3 LINKSET 1 LINK 1
Timestamp:        5-MAR-2003 14:50:25.14
```

Example

```
S7MP_C> UNINHIBIT FEP_SUBSYSTEM PILOTE MTP3 LINKSET 2 LINK 1
```

Produces the following result if the LINK entity is in a disabled state:

```
@SS7-E-NOTDONE, action not performed
Entity Name:      FEP_SUBSYSTEM PILOTE MTP3 LINKSET 2 LINK 1
Timestamp:        8-MAR-2003 15:51:48.87
Action Status:    SS7_K_STATE_INVALID
Reason:           SS7_K_LINK_DISABLED
```

Example

```
S7MP_C> UNINHIBIT FEP_SUBSYSTEM PILOTE MTP3 LINKSET 1 LINK 1
```

Produces the following result if the parent entity of the LINK does not exist:

```
@SS7-E-NOSUCHPARENT, no such parent
Class:            LINKSET
Instance_id:      1
Timestamp:        5-MAR-2003 14:50:25.14
```

6.51 UNTRACE

Format

UNTRACE

Parameters

None.

Qualifiers

None.

Description

Closes the trace file that was specified with the TRACE command.

Example

```
S7MP_C> UNTRACE
```

When the UNTRACE command has completed successfully, no message is returned. If you enter the UNTRACE command when a trace has not been previously opened, the following message is returned:

```
@SS7S7MP-I-NOTRF, No trace has been opened
```

6.52 WAIT

Format

WAIT *period_in_seconds*

Parameters

period_in_seconds

The waiting period, specified in seconds.

Qualifiers

None.

Description

Specifies, in seconds, the period to wait before executing the next command.

Example

```
S7MP_C> WAIT 11
```