NAGRAVISION®

DIGITAL PAY-TV SYSTEMS

CONFIDENTIAL

SMS-SM

VERSION 3.0.4 (or above)

User Guide

ISSUE 1.0.3

SAS

NAGRAVISION S.A.

First published, 2001
Revised, 11 Oct 2001
Part number: SasSmmUse 1.0.3

Nagravision S.A. is a member of the Kudelski Group of Companies.
Tel.: (41) (21) 732-0311 Fax: (41) (21) 732-0300

Contents–Summary

Contents–Full listing

List of figures

List of tables

Conventions used in this guide

Pull-quotes

Pull-quotes are used in this document to clearly draw your attention to some parts of the text. See below for the pull-quotes used in this document. The name of the pull-quote or symbol is on the left (For clarity, these appear in the margins, clear of the main body text) and its purpose is detailed to the right.

CAUTION	Provides information to avoid undesirable effects or indicates that an operation or action could give unexpected results or is irreversible (e.g., data loss etc…).
Important	Information that must not be ignored when carrying out some task or tasks.
Note	Further information, advice or exceptions etc…
NOTICE	Information that Nagravision S.A. respectfully requires its customers and/or partners to observe.
	Technical stuff that only need be read by technical staff.
	Provides information by way of a 'TIP' to carry out a task more effectively or efficiently.
	**Indicates advice, which if not observed may result in injury and/or equipment damage.**

Convention for Windows

Item	Description
Menu commands	In **bold** type: e.g., Select **Save.**
Field names, radio buttons and check boxes	In **bold** type: e.g., Select the **Needs publishing** check box.
Items selected in a list box	Items selected are shown inverted
Unselected items appear normal	Items unselected are shown without any treatment.

Convention used

- On-screen **buttons**, menu names, and menu items are in bold type.
- *Directories* and *filenames* are in italic type.
- **Commands** that should be entered exactly as shown are in **bold** type.
- Command ***Variables*** are in italic bold type.
- **Optional items** are in underlined bold type.
- Ellipses "…" indicate that more than one element may be placed on the command line.
- Keys to be pressed are shown in the < and > brackets. For example, the delete key appears as <Del>.
- Pressing <Enter> at the end of a command line assumes that an entry has been effected.
- Screen examples appear in a box like this one:

```
SystemPrompt>Command Parameters1 Parameters2 …
Answer
SystemPrompt>
```

Acknowledgements

Printing or viewing online

NOTICE It is strictly prohibited to print this document if it is marked "for online use only" or to disseminate this document with a screen dump/capture utility or similar tool or to view it on a machine that is not part of the System that it was supplied for use on.

Also, refer to the notice at the bottom this document's front cover.

This document is supplied in Portable Document Format (PDF) format and it requires Adobe Acrobat Reader 3.0 (or later) to be printed (or viewed online). It is recommended to print this document double-sided on A4 paper (Also see note) using a laser printer. If your printer does not have double-sided mode—See your printer's documentation.

Note To print on letter size paper, in Acrobat reader, click **Print…** from the **file** menu (Print dialogue box appears), and then select **shrink to fit** check box.

For your notes

Acronyms and abbreviations

Acronym Abbreviation	Definition	Description
ANI	Automatic Number Identification	Module obtaining the caller's phone number during an IRD callback through a PSTN.
APDU	Application Protocol Data Unit	-
CAS	Conditional Access System	A generic term for a system used in pay television.
Cipher	Ciphering application	-
DOCM	Data Oriented Communication Module	-
DNASP	Digital Nagravision Advanced Security Processor	Name given to Nagravision CAS product.
EIS	External Interface Specification	The EIS Formatter is an internal component of the SMS-SM responsible for converting the format of SMS commands and acknowledgements from the external format used by SMS to the internal format used by the ITM component.
EME	Entitlement Message Encryptor	Device (computer or card) that encrypts or decrypts messages.
EMGR	Entitlement Message Manager	SAS process that stores the EMM in an EMM database, orders the EMM encryption and sends the EMM to the EMM broadcaster.
IIOP	Internet Inter-ORB Protocol	Protocol that allows various ORB to communicate.
IMS	Information Management System	Set of applications in charge of gathering program and conditional access information.
IRD	Integrated Receiver Decoder	Device that allows receiving a signal, demodulates it and de-scrambles it (if inserted smart card gets corresponding rights). Smart card is inserted directly inside. Also known as Set Top Box.
ISD	Integrated Security Device	Also known as smart card or ICC.
ITM	Interactive Transaction Manager	The part of the CAS that manages interactive requests.
MAX	Ascend Product Line	Ascend equipment (modem/router) used between public telephone network and CC to handle CCM. There are many MAX models (1800, 4000, 4004, etc.) and also non-Ascend similar equipments.
MO	Managed Object	Object needed for NSM
NAS	Network Access Server	-
NSM	Nagravision System Management	The subsystem used to monitor and control every component of the CAS.
ORB	Object Request Broker	A software component able to access remote objects complying with the CORBA architecture.
PA	Positive Addressing	Supplementary security in CAS to limit problems in case of right cancellation. For a long right period (e.g. 1 year), right expiration is put to a shorter time interval (e.g. 1 month), and has to be renewed (confirmed) at the end of each interval.
PSTN	Public Switched Telephone Network.	-
QoS	Quality of Service	The quality of service can be negotiated between two

Acronym Abbreviation	Definition	Description
		modules during a service negotiation.
RADIUS	Remote Authentication Dial-In User Service	Interface which allows connection from CC to the MAX with login and password control.
RTM	Regular Transaction Manager	The part of the CAS that manages regular requests and EMM generation.
SEP	Software Environment Platform	The Software Environment Platform provides services to the application level.
SM	Session Manager	The part of the CAS that enables communication with the external world.
SMS-SM	SMS Session Manager	Also known as SMM.
SRS	Software Requirements Specifications	The Software Requirements Specifications document presents the specifications for a particular software component.
SSM	Subscriber Session Manager	The part of the CAS that enables communication with IRD's.
STB	Set-Top Box	The cable TV box "sitting on top" of the TV set that allows receiving signal, demodulates it and de-scrambles it.
T1	Communication Protocol	ISO7816 data link standard protocol (used between IRD and SSM).
UAS	User Application Software	The part of the Zermatt project that is application-specific and not generic.
UDP	User Datagram Protocol	A protocol within the TCP/IP protocol suite that is used in place of TCP when a reliable delivery is not required.
UML	Unified Modeling Language	A modeling method for OO projects.

Table 1 – Acronyms and abbreviations

1. Introduction

This document provides guidance to operate the SMS-SM component from support and engineering perspectives. The SMS-SM is part of Nagravision CAS applications acting as a routing application able to dispatch SMS commands to the various CAS components and, conversely, to dispatch SMS feedback commands to the various SMS currently connected.

The operation of the SMS-SM application, described in this user guide, encompasses the user operation, the maintenance and the troubleshooting aspects.

1.1 Purpose

The objective of this user guide is to provide the operator of the SMS-SM application with the necessary knowledge so that he would be able to establish and maintain the operational mode of the application, as well as to be able to troubleshoot this application.

1.2 Audience

This guide is intended for persons that will operate the SMS-SM product.

The reader is assumed to have a basic understanding of the following:

- The Nagravision CAS,
- Pay TV terminology,
- Basic CORBA knowledge,
- C++ knowledge (optionally).

1.3 References

[1] NSM Product Installation Check List
[2] NSM Applications User Guide
[3] SEP Product Installation Check List
[4] SMS Gateway Interface Definition
[5] ITM DNASP-2 Software User Guide
[6] ITM DNASP-3 Software User Guide
[7] RTM Software User Guide.

1.4 Your comments

We at Nagravision make every endeavor to produce quality and accurate documentation to meet the needs of our customers. Therefore, if you as our valued customer have any suggestion or note a discrepancy in this document we would be pleased if you let us know for inclusion in a subsequent issue.

Important Please send comments to "publications@nagra.com" with subject "Feedback"

1.5 Document history

Version	Date		Description
1.0.3	October 15, 2001	J.-N. Chabaud	• The SMS command 110 'EMM cleanup' is now routed to the SMS Gateway in a DNASP-2 CAS.
1.0.2	October 11, 2001	J.-N. Chabaud	• Corrected the schema of the SMS-SM internal architecture • Added a remark regarding the fact that SMS commands are now processed even if connection with all TM applications is not successful. • Added information about what happens when the connection with the SMS Gateway is lost.
1.0.1	September 19, 2001	J.-N. Chabaud	Described the new parameter SMM_nackGlobalCmds
1.0.0	August 20, 2001	J.-N. Chabaud	First draft based on the information previously contained in the ITM DNASP-2 and ITM DNASP-3 User Guides.

2. SMS-SM overview

This chapter gives an overview of the SMS-SM use within a CAS that exists in two distinct DNASP version (i.e. DNASP-2 or DNASP-3) in accordance with the type of smart card (further named ISD) to be managed.

2.1 SMS-SM within a typical DNASP-2 CAS configuration

The following schema shows a typical DNASP-2 CAS configuration involving the SMS-SM component:

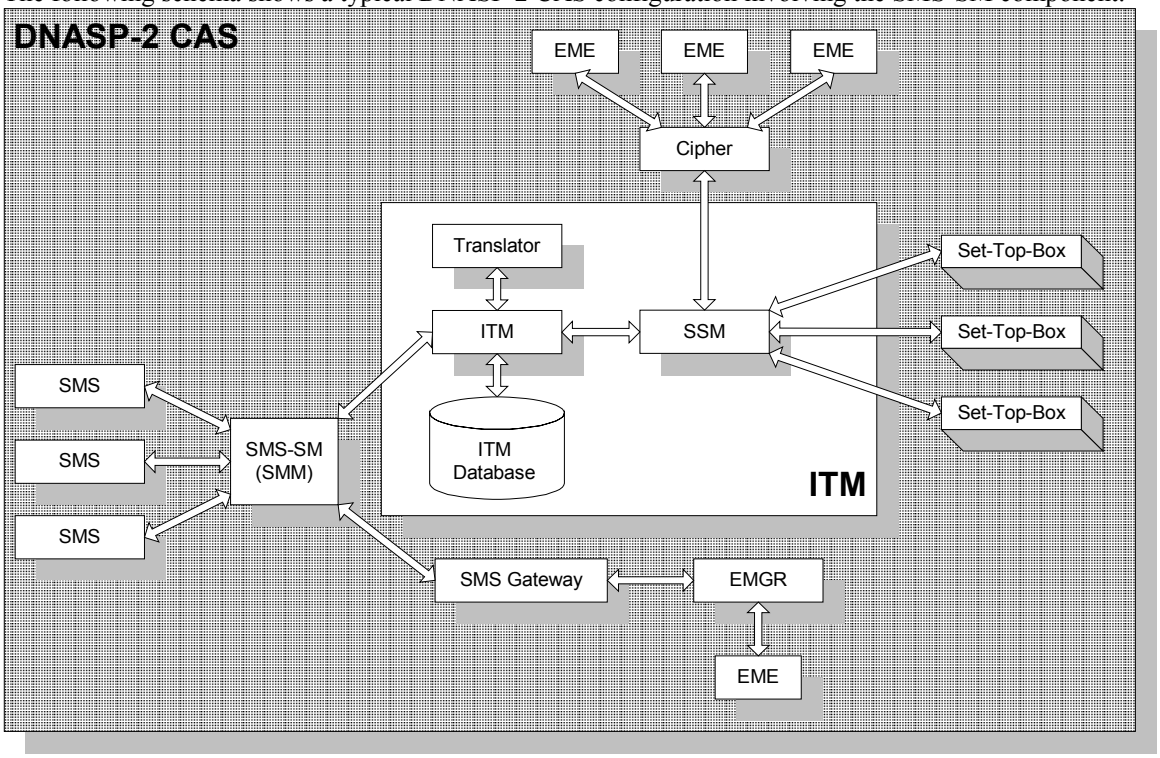

Fig. 2.1 – Typical DNASP-2 CAS configuration

2.2 SMS-SM within a typical DNASP-3 CAS configuration

The following schema shows a typical DNASP-3 CAS configuration involving the SMS-SM component:

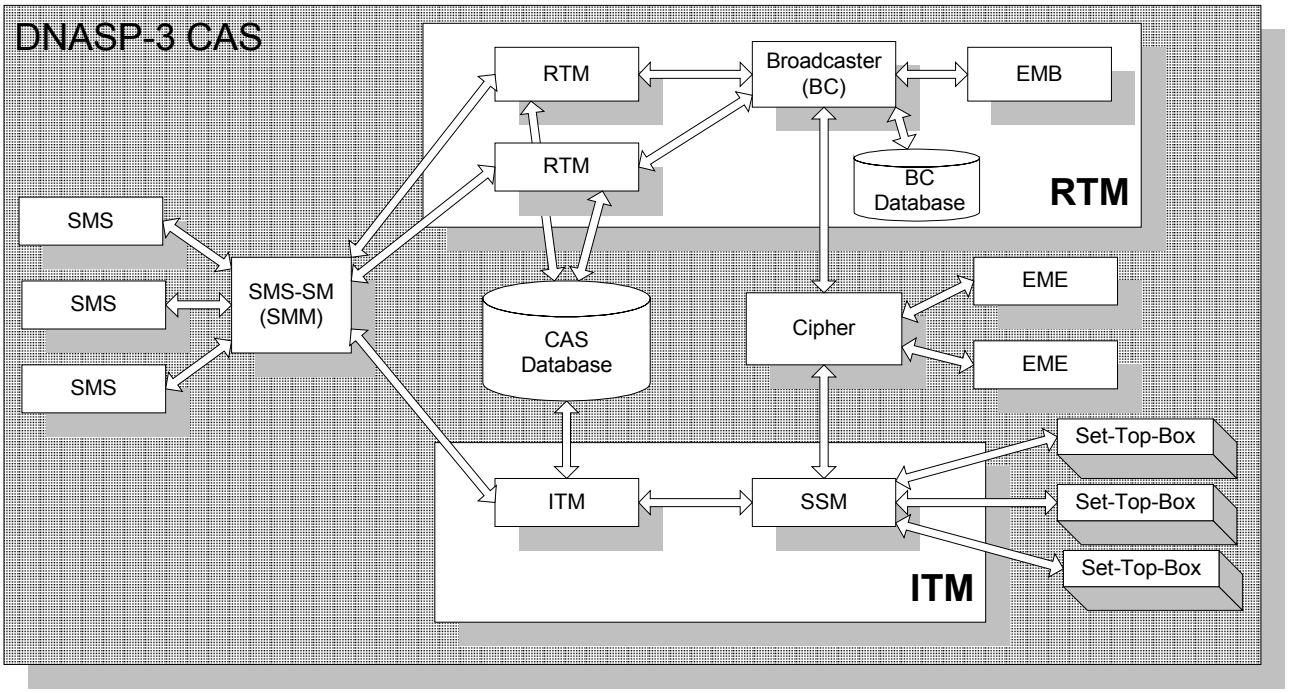

Fig. 2.2 – Typical DNASP-3 CAS configuration

2.3 Metrics and limitations

The SMS-SM is a multi-threaded application that uses one thread for each connected SMS. Furthermore, additional threads are used to handle the CORBA requests forwarded to the TM application(s), and there is a pool of threads to process the responses of this (these) TM application(s). Therefore, the Unix kernel limit to the number of threads per process will define the maximum number of components that may be configured.

The SMS-SM has been successfully tested with up to 15 simulated SMS. It should work with even more SMS, provided the system it runs on has a sufficiently large number of processors and is suitably configured.

2.4 Connection and reconnection strategy

The next schema shows the roles of the various components around the SMS-SM. In the following figure, an arrow from component c1 to c2 means c1 retries to connect to c2 when connection is not successful.

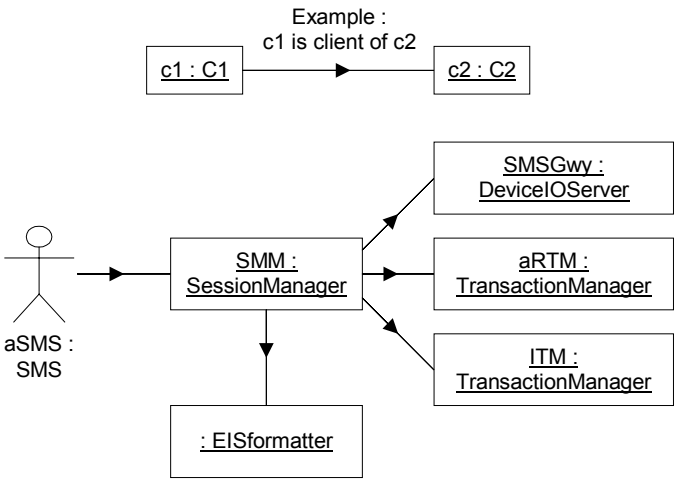

Fig. 2.3 – Reconnection strategy

In this schema, the client of the SMS-SM is the SMS front end application. To be noticed that the EIS Formatter component is internal to the SMS-SM application so that no network connection is involved between these two components.

From a SMS application's perspective, the SMS-SM acts as a server. It waits for TCP/IP connections from the SMS clients. Requests are made within the context of this connection. If this connection is broken, the SMS-SM keeps in its internal communication queues all feedback commands and acknowledgements destined to the various unreachable SMS and resumes sending these messages once the connection is re-established.

Note Depending on the usage of the SMS-SM component in either a DNASP-2 or a DNASP-3 CAS, the SMS-SM will establish a connection with the legacy SMS Gateway component or with the RTM component but not with both at the same time…

From a CAS component's perspective, the SMS-SM acts as a client:

- The connection with the legacy SMS Gateway (within a DNASP-2 CAS) is established by the SMS-SM when the SMS connects to the SMS-SM. The Device IO handshake established between the SMS and the SMS-SM is then reproduced between the SMS-SM and the SMS Gateway. If, for any reason, the SMS Gateway is unavailable then the connection with the SMS is closed at once. Similarly, if the connection with the SMS Gateway is lost during the normal processing of SMS commands, then the SMS-SM will close the connection with the SMS. This disconnection will occur once all SMS commands being currently processed by the SMS Gateway have experienced a timeout (accordingly to the SMM_GWYtimeout parameter).
- The connection with the TM application (i.e. ITM and / or RTM) is established by the SMS-SM during its startup time. If the CORBA interface of a TM application is not available, the SMS-SM will skip this connection and try to connect to the other ones. At the end of its initialization, the SMS-SM starts processing SMS commands even if one, several or all connection(s) to the TM application(s) could not be established. Once a CORBA connection is established with a TM application, all requests are made within the context of this connection. If the connection is broken during the normal processing of SMS commands, there is no way to retrieve the result of a previous request and this latter must be submitted again. The SMS-SM is

responsible for re-establishing the connection if it is broken. This is done by an internal ‘trigger’ (one for each TM application) responsible for regularly polling ‘its’ CORBA interface and warning the SMS-SM whenever the interface is available again.

3. SMS-SM architecture

The following schema gives an overview of the internal architecture of the SMS-SM application:

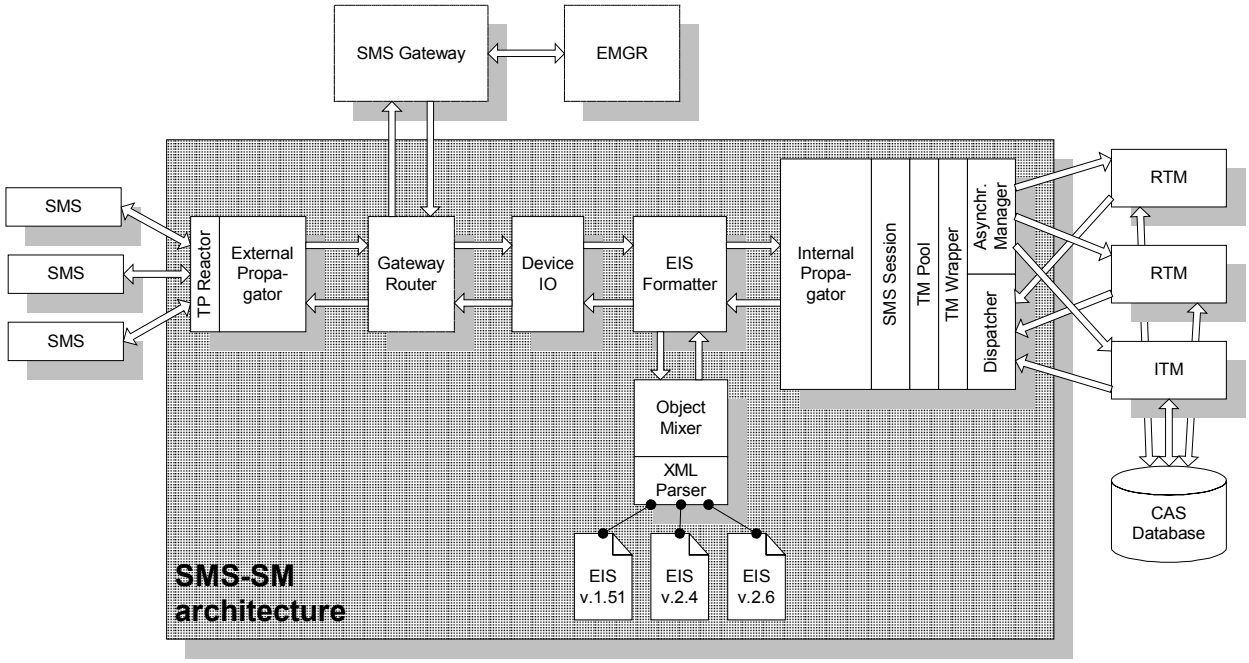

Fig. 3.1 – SMS-SM internal architecture

The following paragraphs describe the various modules of the SMS-SM application.

3.1 External Propagator Reader

This module manages the low level TCP/IP communication with the SMS. An internal component named TP Reactor (where TP stands for Threads Pool) manages efficiently the multiple SMS connections with a pool of threads, each thread serving the successive SMS commands in a recursive way.

3.2 Gateway Router

This module is created within the SMS-SM stream of modules only if a connection with the legacy SMS Gateway is required (i.e. in a DNASP-2 CAS configuration). This module determines which commands are of relevance to the legacy SMS Gateway, to the ITM, or both.

The following constraints are respected by the Gateway Router module:

- The SMS-SM opens one control connection on the SMS Gateway for each SMS connection, and one feedback connection for each SMS feedback connection,
- The sequence in which the commands are sent by the SMS is preserved for each subscriber (except for SMS OPERATION commands),
- Commands that must be sent to both the legacy SMS Gateway and to the ITM are first sent to the SMS Gateway, and then to the ITM, but only if they were positively acknowledged by the SMS Gateway,

- The legacy system (i.e. EMGR then SMS Gateway) does not necessarily send acknowledgements in order,
- The SMS Gateway interface is asynchronous, i.e. the Gateway Router doesn't wait for previous acknowledgements to send the next SMS command,
- The number of open commands on the SMS Gateway is configurable (cf. parameter 'SMM_GWYopenCmds'). If the number of non-acknowledged SMS commands exceeds this limit, the SMS-SM will temporarily stop processing SMS commands on this connection. Either the SMS Gateway acknowledgements will lower this number, or these SMS commands will time out (cf. parameter 'SMM_GWYtimeout ') in the Gateway Router queue if the SMS Gateway acknowledgements are lost.

3.3 Device IO

This module handles the low-level 'Device IO' protocol which encapsulates every SMS command (a description of this protocol is given in [4]).

3.4 EIS Formatter

This module translates incoming and outgoing SMS commands and acknowledgements from their external format (described in document [4]) to the internal format (i.e. sequences of objects) common to the various CAS components, and vice-versa.

Actually, the translation is achieved by the Object Mixer component which applies formatting rules contained into a particular XML file. Any change in the external (or internal) format of SMS commands can thus be easily managed by modifying the corresponding XML rules and restarting the SMS-SM application.

Currently, the supported EIS formats of SMS commands are v.1.51, v.2.4 and v.2.6.

3.5 Internal Propagator

This module manages the communication with the Transaction Manager applications which are the ITM and the RTM applications. Several internal components are involved in the management of the various CORBA interfaces used to communicate with the TM applications:

- **SMS Session**: this component manages the SMS sessions and keeps track of their interfaces (external for the SMS and internal for the TM application) as well as their current states (control port active or inactive, feedback port active or inactive).
- **TM Pool**: this component records and manages the configuration of the TM applications within the CAS. It is involved when a change occurs in this configuration like the addition a new TM Wrapper within a TM pool or the modification of a TM application state (active / inactive).
- **TM Wrapper**: this component manages the communication with the TM application through a CORBA interface. It deals with things like connection, disconnection, reconnection, timeouts, requests sending and TM applications notification (when a configuration change occurs).
- **Asynchronous Manager**: this component manages to send requests to the TM applications by putting them into an internal activation queue which is served by a pool of threads responsible for recursively getting the requests out of the queue and sending them to the appropriate TM application.
- **Dispatcher**: this component manages to receive requests from the TM applications by putting them into an internal activation queue which is served by a pool of threads

responsible for recursively getting the requests out of the queue and sending them back to the appropriate SMS.

4. Data flows managed by the SMS-SM

This chapter describes the various data flows that are managed by the SMS-SM.

4.1 Data flow between the SMS and the SMS-SM

4.1.1 SMS connection

The communication protocol between the SMS and the CAS system is the 'Device IO' protocol (a description of this protocol is given in [4]). In this protocol, the SMS is a client that connects to the CAS system and once the connections have been established, it's then possible for the SMS to issue business commands on one link as well as retrieving feedback commands on another link.

Note The entry communication point the SMS has to connect to are two TCP/IP sockets corresponding to the 'SMM_address', 'SMM_port' and 'SMM_feedbackPort' parameters of the SMS-SM component.

4.1.2 SMS business commands

Once the 'Device IO' connection has been established, the SMS-SM is then ready to accept and process business commands. The description of each of these commands is given in document [4]. These commands will be dispatched in different ways over the CAS components, depending on their type.

Note The field 'source_ID' the SMS must provide within any business command is a parameter of the SMS-SM component. It must be either :

- Equal to the 'SMM_uniqueSID' parameter if the mono SMS configuration is used,
- Referenced in the 'SMM_smsList' parameter list if the multi SMS configuration is used.

4.1.3 SMS command acknowledgement

An acknowledgement (positive or negative) is generated when the CAS system receives a business command. On the other hand, acknowledgements are also generated by the SMS to acknowledge feedback commands. Once again, document [4] gives a full description of theses acknowledgements also known as OPERATING commands.

Nominal mode

In the nominal operating mode, the SMS-SM only responds to SMS business commands with positive acknowledgements (operating command 1000), indicating that the command was successfully processed. In that mode, the SMS also has to positively acknowledge the feedback commands forwarded by the SMS-SM.

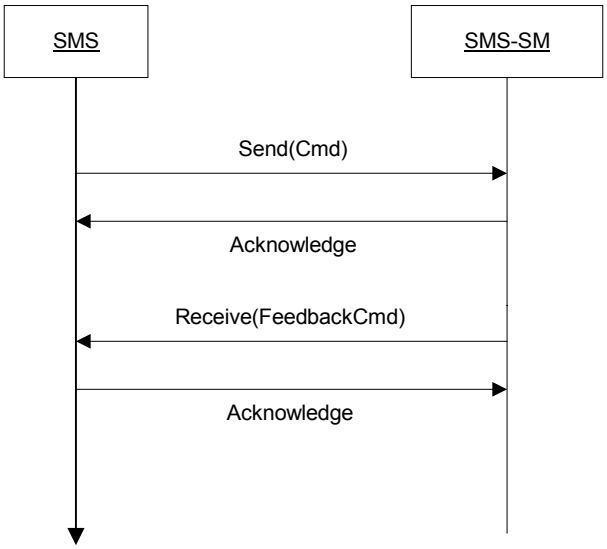

Fig. 4.1 – SMS acknowledgement: nominal mode

Non-nominal mode

In the non-nominal situation, the SMS-SM responds with negative acknowledgements (operating command 1001) to the SMS. It also may happen that the communication link is closed if a further component of the CAS is missing.

The transaction number returned within the negative acknowledge allows to know which component generated the acknowledgement:

Transaction number	Negative acknowledgment comes from
'0xxxxxxxx'	SMS Gateway (DNASP-2 CAS only)
'5xxxxxxxx'	SMS-SM
'6xxxxxxxx'	ITM
'7xxxxxxxx'	RTM (DNASP-3 CAS only)

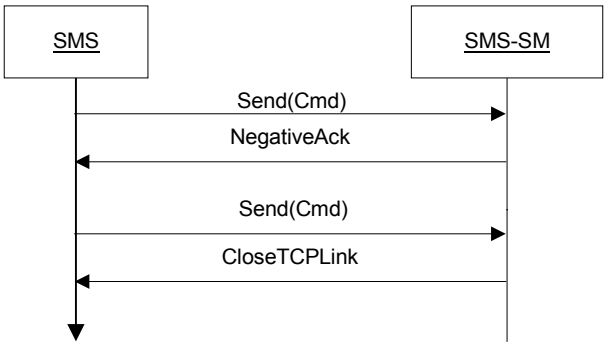

Fig. 4.2 – Acknowledgement to SMS command: non-nominal mode

For unknown feedback commands, the SMS may also respond with negative acknowledge.

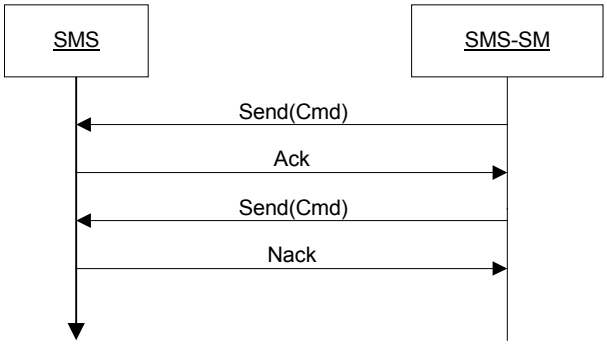

Fig. 4.3 – Acknowledgement to feedback command

4.1.4 SMS disconnection

In order to disconnect from the CAS system, the SMS only has to close the TCP/IP communication link with the SMS-SM component.

4.1.5 Monitoring

All the events generated by the ITM components may be forwarded to one or several distinct event reporting application(s). In this way, events such as SMS connection / disconnection or SMS commands hits are reported to the operator through the event reporting tools.

4.1.6 Error handling

Negative acknowledgements are the way to signal to the SMS any trouble with a given command. Nevertheless, some situations of trouble on the components themselves may bring to communication closure (in case of crash of the SMS-SM component for instance).

4.2 SMS commands routing

Depending on the DNASP version of the CAS (either DNASP-2 or DNASP-3) the Gateway Router module is activated or not and the routing of SMS commands is modified accordingly.

Note Within a DNASP-2 CAS, most of the SMS business commands are sent to the ITM, whereas, within the DNASP-3 CAS, these commands are managed by the RTM application and not any more by the ITM application!

4.2.1 Routing within a DNASP-2 CAS

When the SMS sends commands to the SMS-SM, this latter dispatches them to the SMS Gateway and to the ITM. Some of the commands are only forwarded to the SMS Gateway, some others are only forwarded to the ITM, and some to both the SMS Gateway and the ITM. These latter SMS commands are sent to the SMS Gateway in a first time and then forwarded to the ITM upon reception of the SMS Gateway acknowledgement (but only if the received acknowledgement is positive). If the SMS Gateway acknowledgement is negative, it is directly sent back to the SMS.

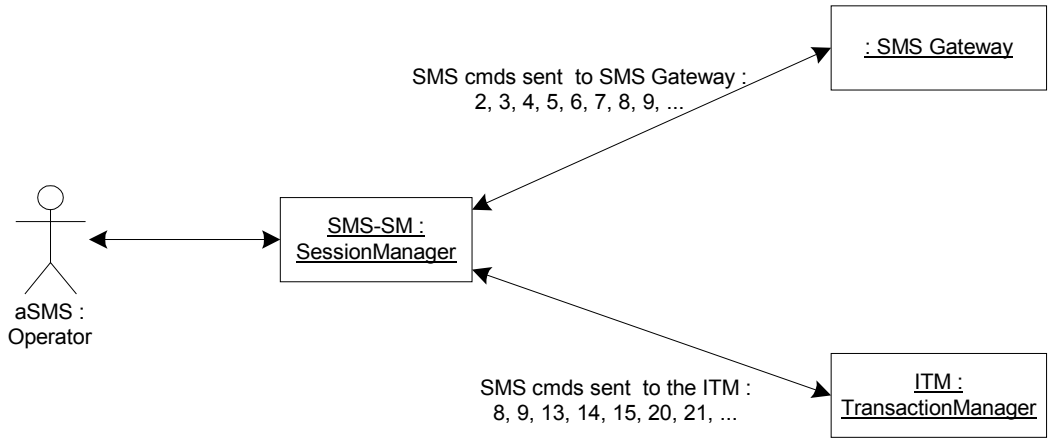

Fig. 4.4 – SMS commands routing in DNASP-2 CAS

SMS commands are routed according to the following rules:

- **Commands sent to the SMS Gateway**:
  - EMM commands : 2, 3, 4, 5, 6, 7, 8, 9, 10, 13, 14, 15, 20, 21, 48, 49, 50, 51, 52, 53, 54, 62, 64, 69, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 93
  - CONTROL commands : 110
  - PRODUCT_DEF commands: 300, 301, 302, 303, 304, 305, 306, 307, 308, 309
  - OPERATION commands: 1002

- **Commands sent to the ITM**:
  - EMM commands : 8, 9, 13, 14, 15, 20, 21, 49, 50, 51, 52, 54, 62, 93
  - CONTROL commands : 100, 101, 104, 105, 111
  - OPERATION commands : 1000, 1001, 1002

4.2.2 Routing within a DNASP-3 CAS

When the SMS sends commands to SMS-SM, these commands are routed either to the RTM or to the ITM. In fact, all SMS business commands (i.e. EMM and CONTROL commands) are routed to the RTM whereas the OPERATION commands (i.e. SMS commands 1000, 1001 and 1002) are routed to the ITM.

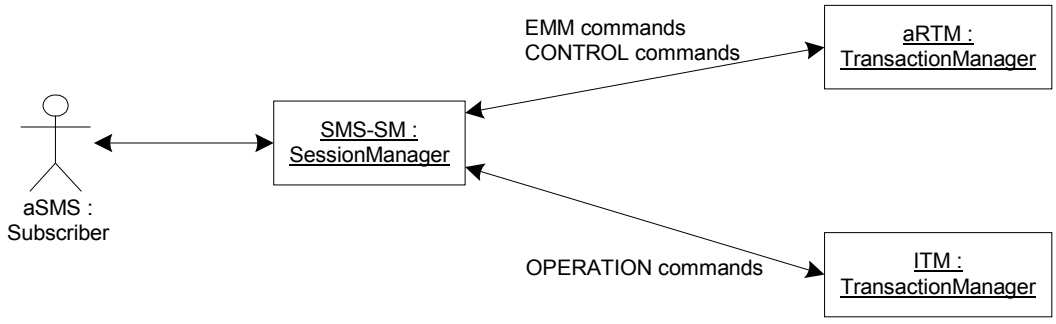

Fig. 4.5 – SMS-SM routing table

To be noticed that both commands 1000 and 1001 are acknowledgements returned by the SMS upon reception of feedback commands generated by the ITM, whereas the SMS command 1002 is a 'no operation' command sent by the SMS only once at connection time to notify the CAS that it is currently connected and to provide its source id.

4.3 Data flow between the SMS-SM and the SMS Gateway

This chapter only applies to a DNASP-2 CAS configuration.

4.3.1 SMS-SM connection

The connection between the SMS-SM and the SMS Gateway is automatically established when the SMS connects to the SMS-SM component. This connection relies on an exchange of messages using the Device IO protocol.

4.3.2 Monitoring

The commands forwarded to the SMS Gateway and their relative acknowledgements are logged within a file called `SMS_GATEWAY_IO.LOG-X` (where `X` stands for an incremental number) in the `$acs_log` directory. The command `sig 3 sms_gateway` can be used to flush the last processed commands in the last version of the file.

4.3.3 Error handling

Any error raised by the SMS Gateway or by the EMGR generates a negative acknowledgement that is returned to the SMS without affecting the ITM component.

4.4 Data flow between the SMS-SM and the RTM

This chapter only applies to a DNASP-3 CAS configuration.

4.4.1 SMS-SM connection

Depending on the list of CORBA interface names provided by the 'SMM_rtm' parameter, the SMS-SM can work with one or several RTM applications at the same time. The connection(s) between the SMS-SM and the RTM(s) is (are) automatically enabled at SMS-SM startup (or later if unavailable at SMS-SM startup time) by means of one (several) CORBA interface(s). A particular state is granted by the SMS-SM to each distinct RTM. This state can be one of the following:

- **Not connected**: the RTM CORBA interface couldn't be found by the SMS-SM and no connection is established with the RTM,
- **Inactive**: a CORBA connection could be established with the RTM but it is considered as 'inactive' and no SMS command is forwarded to it.
- **Active**: a CORBA connection could be established with the RTM which is considered as 'active' and SMS commands are forwarded to it.

Note If the 'SMM_rtmAutoActivate' startup parameter is set to true, the 'Active' state is automatically granted to the RTM at connection time, whereas only the 'Inactive' state is granted if this parameter is set to false. In this latter situation, one must manually call the 'activate' method of the TM Wrapper Managed Object (MO) for the 'Active' state to be granted to the corresponding RTM.

4.4.2 Commands routing (again)

When a RTM is considered by the SMS-SM as 'active', SMS commands can be forwarded to it. If only one RTM is configured in the CAS system, all SMS commands are sent to it. If several RTM are configured, SMS commands are routed to the various RTM using a modulo on the Unique Address (UA) found in the SMS command. For instance, if two RTM are used, the first one (i.e. the one corresponding to the first name of the 'SMM_rtm' startup list) will receive SMS commands for UA 1, 3, 5, 7, 9, etc…, whereas the second RTM will receive SMS commands for UA 2, 4, 6, 8, 10, etc.

4.4.3 SMS-SM reconfiguration

Whenever one of the RTM crashes, a specific trigger (regularly polling the RTM CORBA interfaces) notifies the SMS-SM which enters a reconfiguration phase where the following actions are taken :

- The stream of incoming SMS commands is suspended (i.e. TM Wrappers are no more fed with new SMS commands),
- The SMS-SM waits for the queues of all TM Wrappers to be empty (i.e. SMS commands remaining in the queues are sent to the corresponding RTM for processing),
- SMS commands present in the queue of the TM Wrapper corresponding to the crashed RTM are negatively acknowledged with a 'Postponed' status,
- The SMS-SM computes the number of RTM remaining in the CAS,
- Each RTM is then notified (through a specific 'signal' CORBA interface) the following values :
  - The total number of 'active' RTM,
  - The position (i.e. modulo) of the RTM among the pool of 'active' RTM.

Note These two parameters are used by the RTM to reload the database of the Broadcaster (BC) component in case this latter crashes and has to be restarted.

  To be noticed that the SMS-SM tries also to notify the crashed RTM with a pair of null values for it to know that it is excluded from the pool of 'active' RTM.
- The stream of SMS commands is resumed and the queues of the TM Wrappers are fed again (each queue receiving SMS commands with UA corresponding to the newly calculated modulo).

The same reconfiguration phase equally occurs whenever one RTM is manually set to the 'inactive' state by means of a call to the 'activate' method (with 'false' parameter) of the corresponding TM Wrapper Managed Object (MO). The only difference is that SMS commands which are present at this time in the queue of the TM Wrapper are sent to the RTM for processing (before its deactivation) instead of being negatively acknowledged.

Finally, the same process occurs in a similar way when a previously deactivated RTM is reactivated. This one is then reintegrated in the pool of active RTM after the SMS-SM has reconfigured itself.

4.4.4 Commands processing

The description of the processing of SMS commands by the RTM is out of the scope of this document. Please refer to document [7] for more information.

Note If no RTM is currently available, all SMS commands will be negatively acknowledged by the SMS-SM component with a POSTPONED status.

4.5 Data flow between the SMS-SM and the ITM

4.5.1 SMS-SM connection

The connection between the SMS-SM and the ITM is automatically enabled at SMS-SM startup (or later if unavailable at SMS-SM startup time) by means of a CORBA interface. Like for the RTM, a particular state is granted by the SMS-SM to the ITM. This state can be one of the following:

- **Not connected** : the ITM CORBA interface couldn't be found by the SMS-SM and no connection is established with the ITM,
- **Inactive**: a CORBA connection could be established with the ITM but it is considered as 'inactive' and no SMS command is forwarded to it.
- **Active**: a CORBA connection could be established with the ITM which is considered as 'active' and SMS commands are forwarded to it.

Note If the 'SMM_itmAutoActivate' startup parameter is set to true, the 'Active' state is automatically granted to the ITM at connection time, whereas only the 'Inactive' state is granted if this parameter is set to false. In this later situation, one must manually call the 'activate' method of the TM Wrapper Managed Object (MO) for the 'Active' state to be granted to the ITM.

At connection time, the ITM is notified of the list of every SMS currently connected to the SMS-SM component. This ensures that the ITM can start to generate feedback commands for these SMS (cf. document [5] or [6] for more details).

4.5.2 SMS business commands processing

In a DNASP-2 CAS configuration, the SMS business commands received by the ITM (either EMM or CONTROL commands) are processed according a two-phases commit procedure and all relevant data are stored in the database. See document [5] for more information about this processing.

4.5.3 SMS acknowledgements processing

As indicated in §4.2 the SMS OPERATION commands are routed to the ITM whatever the CAS configuration is (either DNASP-2 or DNASP-3):

- **SMS commands 1000 and 1001**: these are acknowledgements (either positive or negative) returned by the SMS upon reception of feedback commands previously generated by the ITM. Please refer to document [5] or [6] for a detailed description of the processing of these acknowledgements by the ITM.
- **SMS command 1002**: this is a 'no operation' command sent by the SMS only once at connection time to notify the CAS that it is currently connected and to provide its source id. Upon reception of this SMS command, the ITM simply sends back a positive acknowledgement to the corresponding SMS.

4.5.4 Error handling

A negative acknowledgement is generated by the ITM (DNASP-2 CAS) or by the RTM (DNASP-3 CAS) for any problem occurring during the processing of a SMS business command. If, for any reason, the ITM or respectively the RTM is down, a negative acknowledgement is generated by the SMS-SM component with a POSTPONED status. It is then the SMS responsibility to send again the command later.

4.5.5 Monitoring

All the SMS-SM processing can be monitored thanks to the traces described in §6.3 or through the event reporting tools.

5. Starting and stopping the SMS-SM

5.1 Setting up configuration parameters

All configuration parameters of the SMS-SM component are provided in a configuration file named `init_smm.dat` and located in the directory of the service into which the SMS-SM is installed, i.e.:

- `/soft/<service>/<bundle>/smm/current/param`

Note The `init_smm.dat` configuration file is only a **template** file that will not be actually used when starting the SMS-SM component. All parameters present in this file must be copied within the configuration file of the service into which the SMS-SM is installed, e.g. in `init_itm.dat` if the SMS-SM is used within the `itmsoft` service.

Before starting the SMS-SM component, the service configuration file must be edited in order to verify the configuration parameters and adapt them if necessary.

The next paragraphs give the description of all configuration parameters needed by the SMS-SM component.

Note The configuration parameters are split in two categories:
- some parameters should usually be customized for a successful installation. These are marked with a specific [CFG] tag,
- other parameters have default values which should prove satisfactory in most circumstances. They should not be changed unless a specific 'expert' configuration is required (for performance enhancement for instance).

5.1.1 Customized environment parameters

The following table gives the list of environment parameters to be defined:

Parameter	Default value	Remark
BUNDLE_NAME	itm1	Name of the software bundle the SMS-SM belongs to.
SEP_VERSION	1.5.0-D4.0g	Version of the Software Environment Platform (SEP)

Table 5-1 – Customized environment parameters

5.1.2 System parameters

The following table gives the list of system parameters to be defined:

Parameter	Default value	Remark
VISIBROKER_HOME	${VISIBROKER33_HOME}	This definition can be commented out if the environment variable is defined elsewhere.
SEP_HOME	/soft/sep/SEPv ${SEP_VERSION}	This definition can be commented out if the environment variable is defined elsewhere.
ITM_COMMON_HOME	/soft/${SERVICE_NAME}/${BUNDLE_NAME}/common_itm/current	Must be defined if the SMS-SM is installed within the ITM service.
RTM_COMMON_HO	/soft/${SERVICE_NAME}/${BUN	Must be defined if the SMS-SM is installed

Parameter	Default value	Remark
ME	DLE_NAME}/common_rtm/current	within the RTM service.
TM_HOME	/soft/${SERVICE_NAME}/${BUN DLE_NAME}/common_ims/current	Location of the TM utility scripts.
CIPHER_ HOME	/soft/${SERVICE_NAME}/${BUN DLE_NAME}/smm/current	Location of the SMS-SM application.
ROOT_DATA	/users/${SERVICE_NAME}/${BUN DLE_NAME}/data	Location of the log files (by default log files of all applications are centralized there).
thisHost	$HOSTNAME	Host name.

Table 5-2 – System parameters

5.1.3 Environment expert parameters

The following table gives the list of environment expert parameters to be defined:

Parameter	Default value	Remark
id	"<id>"	Parameter used for internal substitution of the instance identifier. **This value must not be changed!**
PATH	$PATH:$TM_HOME/shell	The current PATH is complemented with the location of the TM utility scripts. **This value must not be changed!**
LD_LIBRARY_PATH	$TM_HOME/shlib:$SEP_HOME/sh lib:$ORACLE_HOME/lib:$VISIBR OKER_HOME/param	Path where dynamic libraries can be found. This definition can be commented out if the environment variable is defined elsewhere.
instanceId	${BUNDLE_NAME}${thisHost}	Instance identifier that will be present in the CORBA name declared by the SMS-SM application. The instance identifier must be specified after the '-i' mandatory parameter of the starting script.
startAppArgs_smm	"-i\s$instanceId\s-c\s-L$ROOT_DATA\s-eCLOG:200\s-b1"	List of default parameters being used only for the SMS-SM application. These parameters can be overridden by parameters specified when calling the starting script.
startAppArgs	"-i\s$instanceId\s-c\s-L$ROOT_DATA\s-eCLOG:200\s-b1"	List of default parameters being used by all applications of the service. These parameters can be overridden by parameters specified when calling the starting script.

Table 5-3 – Environment parameters for expert configuration

5.1.4 SEP global parameters

The following table gives the list of SEP global parameters to be defined:

Parameter	Default value	Remark
SEP_sepInfo	NONE	SEP Internal Info. If set to 'ALL' the list of SEP shared libraries loaded at startup is displayed.
SEP_appInstanceName	$id	Application instance name.
SEP_eventReportingName	${id}ER	CORBA name of the event reporting application.

Parameter	Default value	Remark
SEP_moEventOnCerr	1	0: The events are not written in log files 1: The events are written in log files
SEP_moEventBodyMax	1024	Maximum length of Managed Object event body.
SEP_ORBagentport	14001	ORB Agent port (Visibroker 3.0: 14000, Visibroker 3.3: 14001).
SEP_moGlobalPath	/ITM/	Managed object global path used to locate the MO of the SMS-SM application.
SEP_eventDbPath	$SEP_HOME/param/	Location of the NSM events database.
SEP_defaultMOVerbosity	LOW	Minimum level for NSM events to be generated.
SEP_OAthreadStackSize	100000	Stack size allocated to threads created by the BOA of Visibroker.
SEP_OAthreadMax	0	Maximum number of concurrent client threads allowed for Visibroker (usually 0, i.e. unlimited).

Table 5-4 – SEP global parameters for expert configuration

5.1.5 SEP Event Reporting Subscriber parameters

The following table gives the list of SEP Event Reporting Subscriber parameters to be defined. These parameters are used when starting the 'show_events' script which launches a subscriber to the Event Reporting application

Parameter	Default value	Remark
ERS_appInstanceName	${id}ERSub	
ERS_life	0	
ERS_appl_type	0	
ERS_source	0	
ERS_time	1	
ERS_discriminator	TRUE	
ERS_moEventBodyMax	1000	
ERS_line	1	

Table 5-5 – SEP Event Reporting Subscriber parameters for expert configuration

5.1.6 SMS-SM parameters

The following table gives the list of the configuration parameters necessary to the SMS-SM component:

Parameter	Default value	Remark
SMM_appInstanceName	${id}	SMS-SM application instance name.
SMM_rtmConnect	1	Activate the RTM connection (0 = false, 1 = true).
SMM_rtmAutoActivate	1	Enable automatic RTM activation when it is restarted. If this parameter is false (0), the RTM activation must be manually requested by calling the 'activate' method of the TM Wrapper Managed Object (MO). See § 4.4 for more details…

Parameter	Default value	Remark
SMM_rtm	${id}RTMSM_TM	Name of the RTM CORBA interface for communication with the SMS-SM. If several RTM applications are to be used, this parameter accepts a list of names separated with comas like for instance RTM1,RTM2,RTM3 **(CAUTION: no space character must be inserted between names !)**
SMM_itmConnect	1	Activate the ITM connection (0 = false, 1 = true).
SMM_itmAutoActivate	1	Enable automatic ITM activation when it is restarted. If this parameter is false (0), the ITM activation must be manually requested by calling the 'activate' method of the TM Wrapper Managed Object (MO). See §4.5 for more details.
SMM_multiSMS	1	Activate the multi SMS feature (enable the use of the 'SMM_smsList' parameter).
SMM_uniqueSID	1234	Unique SMS source id to be used when the multi SMS feature is disabled.
SMM_smsList	'\(0001,0002,0003,0004,0256,1234\)'	List of authorized SMS source ids (only if 'SMM_multiSMS' parameter is activated).
SMM_address	itmsoft	IP address of the machine hosting the SMS-SM.
SMM_port	60002	Port for SMS control connection.
SMM_feedbackPort	60003	Port for SMS feedback connection.
SMM_useLegacy	0	Activate the connection to the legacy SMS Gateway (used only in a DNASP-2 CAS configuration).
SMM_GWYaddr	10.0.2.7	IP address of the SMS Gateway (used only in a DNASP-2 CAS configuration).
SMM_GWYport	20000	Port for the SMS gateway control connection (used only in a DNASP-2 CAS configuration).
SMM_GWYfeedbackPort	20001	Port for the SMS gateway feedback connection (used only in a DNASP-2 CAS configuration).

Parameter	Default value	Remark
SMM_EISProtocol	2.6	EIS protocol version being used (authorized values are 1.51, 2.4 and 2.6). The CAS components (ITM or RTM) generates error codes (in negative acknowledgements) using the latest version of EIS protocol. These codes are further mapped by the SMS-SM to authorized error codes depending on the value of this parameter.
SMM_nackGlobalCmds	true	When activated, this parameter enables the blocking of SMS global commands, i.e. of SMS commands which 'address_type' filed is 'G'. The SMS-SM then builds and sends back to the related SMS a negative acknowledgement for every incoming SMS global command.
SMM_reactorThreads	2	Number of Reactor threads. Recommended values are computed by multiplying the number of SMS expected to be connected simultaneously to the SMS-SM by the factor given in this table: SMM_useLegacy | false | true | Synchronous SMS | 0.5 | 0.7 | Asynchronous SMS | 1.0 | 1.2 | If you experience negative acknowledgments due to timeouts, try increasing this number of threads.

Table 5-6 – SMS-SM parameters to customize

The following table gives the list of the 'expert' configuration parameters necessary to the SMS-SM component:

Parameter	Default value	Remark
SMM_corbaInterface	UNDEFINED	CORBA interface of the SMS-SM for incoming DNASP commands from the ISD Synchronizer (ISS) and from the DNASP Command Generator (DCG) applications (this interface is currently in development and should not be used in the current release of the SMS-SM).
SMM_corbaFeedbackInterface	UNDEFINED	CORBA interface of the SMS-SM for outgoing acknowledgements destined to the ISD Synchronizer (ISS) and to the DNASP Command Generator (DCG) applications (this interface is currently in development and should not be used in the current release of the SMS-SM).
SMM_name	${SMM_appInstanceName}SMM	SMS-SM name.
SMM_formatter	${id}EISFormatter	EIS Formatter name.
SMM_itm	${ITM_SMMInterface}	Name of the ITM CORBA interface for communication with the SMS-SM.
SMM_itmOutgoingThreads	5	Number of threads sending data to the ITM (i.e. SMS feedback command acknowledgements).
SMM_itmIncomingThreads	5	Number of threads receiving data from the ITM (i.e. SMS feedback commands).
SMM_smsWrapper	${id}SMSWrapper	Name of the SMS interface.
SMM_eventQueueMaxSize	500	Event queue max size for Event Reporting

Parameter	Default value	Remark
		application.
SMM_GWYtimeout	150	Timeout of the SMS Gateway [sec] (used only in a DNASP-2 CAS configuration).
SMM_rtmOutgoingThreads	15	Number of threads to send data to the RTM (i.e. SMS commands). For optimal performances, the 'RTM_submitThreadPoolNumber' parameter of the RTM should be given the same value as this current parameter.
SMM_rtmIncomingThreads	${SMM_rtmOutgoingThreads}	Number of threads to receive data from the RTM (i.e. SMS command acknowledgements). For performance reasons, this parameter is usually given the same value as the 'SMM_rtmOutgoingThreads' parameter.
SMM_immediateOffset	10	Offset in minutes for an immediate callback. This parameter is used when processing a SMS command 60 'Immediate Callback' and when the connection to the SMS Gateway is activated. This offset is then added to the current date and time to get the date and time of the next immediate callback of the ISD.
SMM_startHour	23	Start hour of the callbacks. This parameter is used when processing a SMS command 61 'Enable Automatic Callback' and when the connection to the SMS Gateway is activated.
SMM_deltaTime	24	Number of hours during which callbacks are authorized. This parameter is used when processing a SMS command 61 'Enable Automatic Callback' and when the connection to the SMS Gateway is activated.
SMM_tmBindTimeout	0	Timeout when connecting to a Transaction Manager like ITM or RTM [sec].
SMM_tmSendTimeout	30	Timeout when sending data to a Transaction Manager like ITM or RTM [sec].
SMM_tmReceiveTimeout	30	Timeout when receiving data from a Transaction Manager like ITM or RTM [sec] .
SMM_xmlFileName	${SMM_HOME}/param/EIS_${SMM_EISProtocol}_formatter.xml	XML file name for EIS Formatter. This parameter is automatically defined upon the version of the EIS protocol used and should not be modified.
SMM_cfgFileName	${SMM_HOME}/param/EIS_${SMM_EISProtocol}_formatter.cfg	Configuration file name for EIS Formatter. This parameter is automatically defined upon the version of the EIS protocol used and should not be modified.
SMM_smsCmdLog	true	Activation of SMS commands logging.
SMM_smsAckLog	true	Activation of SMS acknowledgements logging.
SMM_fbCmdLog	true	Activation of Feedback commands logging.

Parameter	Default value	Remark
SMM_fbAckLog	true	Activation of Feedback acknowledgements logging.
SMM_GWYopenCmds	100	Maximum number of asynchronous commands still not acknowledged by the SMS Gateway (used only in a DNASP-2 CAS configuration).
SMM_minEventLevel	MEDIUM	Minimum level required to generate an event. The authorized values are: DEBUG, VERY_LOW, LOW, MEDIUM, HIGH and VERY_HIGH. To be noticed that events of DEBUG level are not produced under the form of NSM events but are directly sent to the Unix standard output. These events will then be only visible in the circular logs of the SMS-SM.
SMM_minEventSeverity	INFO	Minimum severity required to generate an event. The authorized values are: INFO, WARNING, ERROR and FATAL.
SMM_eventFilterOption	FILTER_OR	Event filtering option. The authorized values are: FILTER_OR and FILTER_AND (which is more restrictive). For instance, if FILTER_AND is used, an event will be produced only if its level is at least MEDIUM **and** if its severity is at least INFO.
SMM_eventOnCout	false	Display events on standard output. If enabled, this option discards the generation of NSM events!
SMM_moDictionaryFileName	${SMM_HOME}/param/smm_counters.dat	Name of the SMS-SM MO dictionary used for counter translation (to be used in conjunction with the Statistics and Events Managed Object, or SEM MO). This dictionary is not used at the present time but will be used in a further release…
SMM_semMOName	${id}SMM_SEM_MO	Name of the CORBA interface of the Statistics and Events Management Managed Object, or SEM MO.
SMM_timing	false	Enable timestamps for performance measurements.
SMM_OAthreadStackSize	100000	Stack size allocated to threads created by the BOA of Visibroker.
SMM_OAthreadMax	0	Maximum number of concurrent client threads allowed for Visibroker. Zero means unlimited.

Table 5-7 – SMS-SM parameters for expert configuration

5.2 Optional environment variables

There is no need to set any environment variable to start the SMS-SM application. All the environment variables needed by the application are defined within the startup scripts using the global configuration script *init_<service>.dat* located in the common section of the service where the SMS-SM is installed.

However, the following optional environment variables may be set to diagnose the starting steps of the SMS-SM application.

Variable name	Description
TM_PARAM_ID	If this environment variable is set : • The file *init_<bundle_type>${TM_PARAM_ID}.dat* located in */soft/<service>/<bundle name>/common_<bundle_type>/current/param* is used instead of *init_<bundle_type>.dat* • The location of the SMS-SM data directory is set to: */users/<service>/<bundle name>/data/${TM_PARAM_ID}* For instance, if the value of TM_PARAM_ID is '_test' and if the service is 'itm', then • The configuration file being used is: */soft/itmsoft/itm1/common_itm/current/param/init_itm_test.dat* • The location of the SMS-SM data directory is set to: */users/itmsoft/itm1/data/_test*
SHOW_CONFIG_FILE	If set to 'YES' the configuration source file being used is displayed on the current terminal whenever an application is started (the configuration file is the file containing all the environment variables requested by the SMS-SM application).
SHOW_INIT_ENV	If set to 'FULL' all the defined environment variables are displayed when starting the SMS-SM application.

Table 5-8 – Optional environment variables

5.3 Starting up the component

Important Make sure the bundle identifier is effectively present at the end of all commands indicated in this chapter and in the following chapters. For instance, `start_smm` must be typed `start_smm_itm1` if the bundle name is `itm1`.

As a highly recommended prerequisite, an Event Reporting application should be up and running before starting the SMS-SM component. The command **`start_er`** (i.e. `start_er_<bundle name>`) starts the event reporting where the events are redirected.

The command **`start_smm`** (i.e. `start_smm_<bundle name>`) starts the SMS-SM component.

Note All startup scripts (start_smm, start_er, etc.) use the generic **_startApp** command and inherits the following options:

- **-h** Display this help page.
- **-p** Display application parameters.
- **-c** Check application parameters.
- **-l** List running applications.
- **-k** Kill application.
- **-r** Restart the application if it was already running.
- **-V** Display the command to be performed WITHOUT starting the application.
- **-H** Display the application starting directory WITHOUT starting the application.
- **-s** Show command line options at startup.
- **-i** Application Instance name (MANDATORY) (automatically set by the global startup option variable startAppArgs).

-L	Set the data root directory ($ROOT_DATA). Default is $TM_HOME/data.
-o <target>	Redirect standard output to the target.
-e <target>	Redirect standard error to the target. Here are the allowed targets : **NULL** Redirection to /dev/null. **TERMINAL** No redirection (default value). **FILE** Regular text file in $ROOT_DATA/data/log, i.e. : **<AppName>.log** for standard output, **<AppName>.dia** for standard error. **FILE_APPEND** Regular text file (in append mode). **CLOG:<fileSize>** Generate a single circular log file with the specified size (in KBytes). **FILE_SET:<fileSize>:<fileNumber>[:<targetApplication>]** Generates a set of regular text files. The default '_fileSet' manager is used unless a <targetApplication> (i.e. a target log file manager) is specified. See § 6.3.2 for more information… **UDP[:<host>:<port>:<appId>]** Sends the output to a UDP port on the given host.
-b <flag>	If the flag's value is 1, backups the old application log files into the archive directory before starting the application. If the flag's value is zero, existing log file is not backed up.

5.4 Stopping the component

The command **`stop_smm`** (i.e. `stop_smm_<bundle name>`) shutdowns the SMS-SM (a kill signal –15 is sent to the process). All data stream currently managed by the SMS-SM will be suddenly interrupted.

6. Monitoring tools

Several tools are included in the TM utility package, consisting in:

- Showing the TM environment,
- Checking the presence of the processes,
- Displaying the component's output.

6.1 Showing the TM environment

The command **`showenv_tm`** (i.e. `showenv_tm_<bundle name>`) displays the system environment variables currently defined for the service. Here is an example of the resulting information being displayed by this command:

```
~ > showenv_tm
%_initenv_g.sh-I, Config file: /soft/itmsoft/itm2/common_itm/current/param/init_itm.dat
VISIBROKER_HOME  /soft/local/visibroker_33/current (1.0.1)
SEP_HOME         /soft/imssoft/operator/sep/current/
TM_HOME          /soft/itmsoft/itm2/tmutil/current (1.0.0-D4.0g)
ORACLE_HOME      /soft/local/oracle/product/8.1.6.2.0
TNS_ADMIN        /soft/itmsoft/itm2/common_itm/current/param
TWO_TASK         casdb1.sas03
ORACLE_USER      sasdba
ORACLE_PASS      nagra
TM_PARAM_ID
instanceId       itm2IT2CAS01
startAppArgs     -i itm2IT2CAS01 -c –e CLOG:200 –b1 –L /users/itmsoft/itm2/data

PATH
                 /usr/bin
                 /usr/bin/X11
                 /usr/local/bin
                 /usr/sbin
                 /sbin
                 /users/local/operator/bin
                 .
                 /usr/ccs/bin
                 /users/operator/bin
                 /soft/itmsoft/itm2/tmutil/current/shell

LD_LIBRARY_PATH
                 /soft/itmsoft/itm2/tmutil/current/shlib
                 /soft/imssoft/operator/sep/current//shlib
                 /soft/local/oracle/product/8.1.6.2.0/lib
                 /soft/local/visibroker_33/current/param
```

6.2 Checking the processes

The next two commands display the processes with their memory and the elapsed time since they were started:

- **`sys_smm_<Bundle_name>`** : shows only the SMS-SM process,
- **`sysapp_<Bundle_name>`** : shows all SEP based processes (SMS-SM, RTM, ITM, etc.).

These two commands can specify the following parameters:

-h Display this help list.

-i Application Instance name pattern: only processes of an instance id

-u User name.

-p Process list (separated by comma, without space).

-n Process name pattern.

-l List searched process.

-k Kill application.

-f Do not ask for confirmation (used with -k option).

-a Display all process arguments.

Here are two examples of the resulting information being displayed by the **sysapp_<Bundle_name>** command:

- **sysapp_itm1**

```
IT2CAS01>operator% sysapp_itm1
  pid user         instance       application              vsiz  rsiz   cpu

28809 operator     itm1IT2CAS01   EventReportingApp       17.7M  2.7M   0.0
28948 operator     itm1IT2CAS01   CipherDnasp3App         28.4M  3.4M   0.0
28881 operator     itm1IT2CAS01   ITMDnasp3App            63.1M  3.3M   0.0
28960 operator     itm1IT2CAS01   SMMApp                  98.4M   19M   0.0
26843 operator     itm1IT2CAS01   SSMDnasp3App            46.1M  2.7M   0.0
27102 operator     rtm1IT2CAS01   BCApp                   66.4M   12M   0.0
30654 operator     rtm1IT2CAS01   DnaspGroupComposer      27.4M  8.8M   0.0
30920 operator     rtm1IT2CAS01   ISSApp                  48.0M  8.9M   0.0
30856 operator     rtm1IT2CAS01   MainPa                  51.2M  7.7M   0.0
30803 operator     rtm1IT2CAS01   RTMApp                  68.0M   11M   0.0
30595 operator     rtm1IT2CAS01   SmsDnaspTranslator      54.9M   20M   0.0
29223 operator     rtm1IT2CAS01   Subscriber              16.3M  2.7M   0.0
```

- **sysapp_itm1 -n SMM -a**

```
IT2CAS01>operator% sysapp_itm1 -n SMM -a

  pid user         instance       application              vsiz  rsiz   cpu
28960 operator     itm1IT2CAS01SMM SMMApp                  98.4M   19M   0.0
###
### Application: /soft/itmsoft/itm1/smm/current/bin/SMMApp
###
GWYaddr                  = 10.0.2.7
GWYfeedbackPort          = 20001
GWYopenCmds              = 100
GWYport                  = 20000
GWYtimeout               = 150
OAthreadMax              = 0
OAthreadStackSize        = 100000
ORBagentport             = 14001
address                  = 10.0.7.6
appInstanceName          = itm1IT2CAS01SMM
cfgFileName              = /soft/itmsoft/itm1/smm/current/param/EIS_1.51_formatter.cfg
defaultMOVerbosity       = LOW
deltaTime                = 24
eisProtocol              = 1.51
eventDbPath              = /soft/imssoft/operator/sep/current/param/
eventFilterOption        = FILTER_OR
eventOnCout              = false
eventQueueMaxSize        = 500
eventReportingName       = itm1IT2CAS01ER
fbAckLog                 = true
fbCmdLog                 = true
feedbackPort             = 60003
formatter                = itm1IT2CAS01EISFormatter
immediateOffset          = 10
itm                      = ProdSMMIF
itmAutoActivate          = 1
itmConnect               = 1
itmIncomingThreads       = 5
itmOutgoingThreads       = 5
minEventLevel            = MEDIUM
minEventSeverity         = INFO
moDictionaryFileName     = /soft/itmsoft/itm1/smm/current/param/smm_counters.dat
```

```
moEventBodyMax          = 1024
moEventOnCerr           = 1
moGlobalPath            = /ITM/
multiSMS                = 1
name                    = itm1IT2CAS01SMM
port                    = 60002
reactorThreads          = 2
rtm                     = rtm1IT2CAS01RTMSM_TM
rtmAutoActivate         = 1
rtmConnect              = 1
rtmIncomingThreads      = 15
rtmOutgoingThreads      = 15
sepInfo                 = NONE
smsAckLog               = true
smsCmdLog               = true
smsList                 = (0001,0002,0003,0004,0256,1234,0200)
smsWrapper              = ProdSMSWrapper
startHour               = 23
timing                  = false
tmBindTimeout           = 15
tmReceiveTimeout        = 30
tmSendTimeout           = 30
uniqueSID               = 1234
useLegacy               = 0
xmlFileName             = /soft/itmsoft/itm1/smm/current/param/EIS_1.51_formatter.xml
```

6.3 Displaying the component's output

Depending on the startup parameter being used when starting the SMS-SM application, there are several ways to access the application's output which are described in the following paragraphs.

6.3.1 Displaying the content of regular log files

Regular log files may be generated by using the **–e** or **-o** options when starting the application. The size of these regular log files is unlimited (i.e. is only limited by the disk space currently available!). The location of these files is (by default) `$ROOT_DATA/log`. To read the content of these log files, the standard `'more'` or `'tail'` Unix commands can be used. Here is an example:

- `more $ROOT_DATA/log/SMMApp.log`
- `tail -f $ROOT_DATA/log/SMMApp.dia`

6.3.2 Displaying the content of circular log file

A circular log file may be generated by using the **–e CLOG** option when starting the application (this is the default value). The standard error and the standard output of the application being started are then redirected to a circular log file. When using this option, the size of the generated log file will not exceed a maximum specified value. The syntax is:

- **-e CLOG:<file size in Kbytes>**

The log file manager is then a binary application named 'clog'. Only one log file is created and any new information generated by the application is replacing the oldest information in the same file. It is recommended not to exceed 200 KB for the log file size when using this log file manager.

The command **`clog_<bundle_name>`** may be used to view the last information being written in the log file. This command accepts the following parameters:

- **`-r`** or **`–read`**: indicates that the circular log file must be read and displayed.

Once the end of log is reached, the clog utility terminates.

- **`-f`** or **`-follow`**: indicates that the circular log file must be constantly read and that new traces currently written to the file must be displayed. This parameter can be used only when the `-r` or `-read` parameter is also specified.
- **`-rf`**: combination of the two previous parameters.

Here is an example of clog use:

- `clog_itm1 -rf $ROOT_DATA/log/SMMApp.clog`

Note Type CTRL-C to exit the **clog** utility when the follow up mode is used…

CAUTION Due to a technical constraint, the full path of the log file must be specified when using the command **`clog_<bundle_name>`** or this utility will not be able to find the log file and will crash.

6.3.3 Displaying the content of a revolving file set

A revolving file set may be generated by using the **–e FILE_SET** option when starting the application. The syntax is:

- **-e FILE_SET:<file size in Kbytes>:<number of files>**

The standard error and the standard output of the application being started are redirected to a revolving file set manager (i.e. the '_fileSet' perl application) that will create and manage the set of files.

For instance: the option –e FILE_SET:200:5, will generate 5 versions of log files with a maximum of 200 KB each. The name of the log file ends with a version number starting from 1 and incremented by 1 for each new version. Only the 5 last versions are kept. Older versions are deleted (e.g., for the SMS-SM, the log file names will be SMMApp.clog-<version_id>). There is no limit to the log file size and version, except the disk space.

The command **`ctail_<bundle_name>`** can be used to view the last information being written in the log files. This command accepts the following parameters:

- **`-h`** : display the help page.
- **`-f`** : continuously display new output data (same as option -f in UNIX tail command).
- **`-n`** : specify the number of lines to be displayed up to the end of file.

Here is an example of **ctail** use:

- `ctail_itm1 $ROOT_DATA/log/SMMApp.clog`

Note

- The version number of the circular log file must not be specified when using the **ctail** command. This command automatically selects the last log file and shows the most recent information…
- Type CTRL-C to exit the **ctail** utility when the follow up mode is used.

6.3.4 Displaying UDP traces

UDP traces may be generated by using the **–e UDP** option when starting the application. The command **`read_udp_<bundle_name>`** may be used to catch and display the

content of UDP packets being broadcast on the network.

Here is an example of **`read_udp`** used to display UDP traces being broadcast on port number 4008 (i.e. when option '-e UDP:<node_name>:4008:SMM' is specified when starting the SMS-SM):

- `read_udp_itm1 4008`

Note Type CTRL-C to exit the **read_udp** utility…

6.4 Nagravision System Management (NSM) interface

The SMS-SM allows several system management actions to be performed while it is active. There are three distinct management interfaces offered by the SMS-SM:

- A first management interface is related to the SMS-SM instance and is published by the **SMS Wrapper** Managed Object (MO).

Note To be noticed that the name of this MO (i.e. SMS Wrapper) is quite misleading as one could think that one SMS Wrapper MO is created for each SMS being connected. This might be the case in a future release of the SMS-SM, but there is currently only one instance of this MO, whatever the number of connected SMS is.

- Another management interface is bound to every Transaction Manager application (i.e. ITM or RTM) to which the SMS-SM is connected. The Managed Object publishing this interface is named TM Wrapper. There is one distinct MO created for each connected TM application.
- The third management interface is published by the Session Manager Managed Object (MO) and gives information about the various SMS sessions being managed by the SMS-SM.

The next three paragraphs describe the various management actions triggered through these interfaces.

6.4.1 SMS Wrapper management interface

The SMS Wrapper managed object implements an interface that allows:

- Locking or unlocking the stream of SMS commands (currently not implemented).

It is also possible to query and reset the following counters (for statistics purpose):

- Number of SMS commands,
- Number of successful or failed requests sent to the EIS Formatter,
- Number of SMS commands sent to the SMS Gateway,
- Number of SMS feedback commands,
- Number of positive or negative acknowledgements.

The following parameters can also be queried:

- Internal queue size of SMS control and feedback commands,
- SMS Gateway timeout (i.e. the value of the 'SMM_GWYtimeout' parameter),
- Immediate callback offset (i.e. the value of the 'SMM_immediateOffset' parameter),
- Callback start hour (i.e. the value of the 'SMM_startHour' parameter),
- Callback delta time (i.e. the value of the 'SMM_deltaTime' parameter).

6.4.2 TM Wrapper management interface

The TM Wrapper managed object implements an interface that allows, for each related Transaction Manager application:

- Getting the number of threads sending data to the Transaction Manager (i.e. the value of either 'SMM_itmOutgoingThreads' or 'SMM_rtmOutgoingThreads' parameter),
- Getting the number of threads receiving data from the Transaction Manager (i.e. the value of either 'SMM_itmIncomingThreads' or 'SMM_rtmIncomingThreads' parameter),
- Getting or setting the delay [s] between two successive attempts to connect to the Transaction Manager,
- Getting or setting the maximum variation [s] randomly added or subtracted to the connection retry delay,
- Getting the state of the associated Transaction Manager,
- Activating or deactivating the associated Transaction Manager,
- Getting or setting the current state of the automatic activation of the associated Transaction Manager,
- Getting the number of seconds attributed to the Transaction Manager connection, send or receive timeouts (i.e. the value of the 'SMM_tmBindTimeout', 'SMM_tmSendTimeout' or 'SMM_tmReceiveTimeout' parameter).

It is also possible to query and reset the following counters (for statistics purpose):

- Number of requests successfully sent to the Transaction Manager,
- Number of requests that couldn't be sent to the Transaction Manager,
- Number of answers received from the Transaction Manager and successfully sent to the IRD.

6.4.3 Session Manager management interface

The Session Manager managed object implements an interface that allows:

- Getting the list of internal interfaces corresponding to the Transaction Manager applications (i.e. the value of both 'SMM_itm' and 'SMM_rtm' parameters),
- Getting the list of external interfaces corresponding to the IRD (a list of Unique Addresses or UA is provided),
- Getting the list of SMS identifiers (i.e. SMS source id) related to sessions currently established,
- Getting the number of SMS sessions currently established.

Note

At the present time, sessions are created when the SMS-SM starts up, not when the corresponding SMS actually connects. Thus there is no way to know which SMS are effectively connected. This behavior will be changed in a further version of the SMS-SM.

—— END OF DOCUMENT ——