DIGITAL PAY-TV SYSTEMS

CONFIDENTIAL

ITM DNASP-2

Version 2.5.12

User Guide

ISSUE 1.2.0

SAS

NAGRAVISION SA

First published: Aug 2000.

Revision Dates: Nov 2001

Part number: SasItmDn2Use 1.2.0

NagraVision is a division of NAGRAVISION SA.
Tel.: +41.21.732.03.11 Fax: +41.21.732.03.00

Security Policy of NagraVision (Nagravision SA)

A recipient(s) of this document, without exception is subject to a

Non Disclosure Agreement (NDA) of NagraVision (Nagravision SA) prior to delivery.

NOTICE

This document is supplied with an understanding that the notice(s) herein or any other contractual agreement(s) made that instigated the delivery of a hard copy, electronic copy, facsimile or file transfer of this document are strictly observed and maintained.

Polite notice and request to an unintended recipient

Should this document come into your possession and you are not the intended recipient: NagraVision kindly requests and thanks you in advance for making contact at your earliest convenience for instructions on how to proceed with its disposal.

Summary of Contents

Full listing of Contents iv

Conventions used in this guide viii

Acknowledgements ix

Printing or viewing online ix

Your comments ix

Acronyms and abbreviations x

1. Introduction 1

2. Basic business mechanisms 3

3. Communication with the IRD 7

4. Starting and stopping components 15

5. Monitoring tools 25

6. Example of use 31

7. ITM Deployment 41

8. Troubleshooting 43

9. Maintenance 47

Contents–Full listing

Conventions used in this guide viii
Pull-quotes viii
Convention for Windows viii

Acknowledgements ix

Printing or viewing online ix

Your comments ix

Acronyms and abbreviations x

1. Introduction 1
1.1. Purpose 1
1.2. Scope 1
1.3. Audience 1
1.4. Prerequisite 1
1.5. Further reading 2
1.6. Document history 2

2. Basic business mechanisms 3
2.1. Services 3
2.1.1. ITM services 3
2.1.2. SEP services 3
2.1.3. NSM services 4
2.2. Classes and components 4
2.3. Connection and reconnection strategy 5

3. Communication with the IRD 7
3.1. Callback data 7
3.2. Data flow between the IRD and the ITM 7
3.2.1. ANI Provider 7
3.2.2. IRD logon 8
3.2.3. Message processing 8
3.2.4. Monitoring 9
3.2.5. Callback reasons 9
3.3. Data flow between the ITM and the SMS (feedback commands) 10
3.3.1. SMS connection 10
3.3.2. Feedback commands generation 10
3.3.3. Standalone generation of feedback command 206 12
3.3.4. SMS feedback commands acknowledgement 12
3.3.5. SMS downtime management 12
3.3.6. Monitoring 13

4. Starting and stopping components 15
4.1. Setting up configuration parameters 15
4.1.1. Customized environment parameters 15
4.1.2. System parameters 15
4.1.3. Environment expert parameters 16
4.1.4. SEP global parameters 16
4.1.5. SEP Event Reporting Subscriber parameters 17

4.1.6. ITM parameters 17
4.1.7. SSM parameters 20
4.1.8. Translator parameter 21
4.1.9. Optional environment variables 21
4.2. Starting up components 22
4.3. Stopping components 23

5. Monitoring tools 25
5.1. Showing the TM environment 25
5.2. Checking the ITM processes 25
5.3. Checking the ACS processes 27
5.4. Checking the whole application (obsolete) 27
5.5. Monitoring the application 27
5.5.1. Displaying the content of regular log files 27
5.5.2. Displaying the content of circular log file 27
5.5.3. Displaying the content of a revolving file set 28
5.5.4. Displaying UDP traces 29
5.6. Nagravision System Management (NSM) interface 29

6. Example of use 31
6.1. Checking the ITM components 31
6.2. Use case : SMS commands processing 31
6.2.1. Use case prerequisites 31
6.2.2. Displaying traces of the SMS-SM 31
6.2.3. Displaying traces of the ITM 31
6.2.4. Connecting to the SMS-SM 31
6.2.5. Sending SMS commands 32
6.3. Use case : feedback commands processing 34
6.3.1. Use case prerequisites 34
6.3.2. Displaying traces of the ITM components 34
6.3.3. Connecting to the SMS-SM 34
6.3.4. Generating an ISD callback 36

7. ITM Deployment 41
7.1. Initial installation/upgrade 41
7.2. Data migration from Call Collector to ITM 41

8. Troubleshooting 43
8.1. Operating system problems 43
8.2. SMS related problems 43
8.2.1. The SMS connection cannot be established 43
8.2.2. Negative acknowledgements are returned to the SMS 43
8.2.3. No acknowledgement is returned to the SMS 44
8.2.4. No feedback command are generated 44
8.3. IRD related problems 45
8.3.1. The IRD cannot call back 45
8.3.2. The IRD connects to the SSM but there are ciphering problems 45

9. Maintenance 47
9.1. Product support 47
9.2. Application maintenance 47
9.3. ITM database maintenance 47

9.3.1. Database backup ... 47
9.3.2. Database restore ... 47
9.3.3. Database purge ... 48
9.4. Reporting anomalies to NagraVision ... 49

Appendix A

Glossary of terms ... 50

Appendix B

Sequence diagrams ... 51
B.1 Basic mechanisms ... 51
B.1.1 Two phases commit transaction ... 51
B.1.2 Request coordinator ... 51
B.1.3 Acknowledgement strategy ... 52
B.2 Feedback command generation ... 53
B.2.1 Acknowledgement manager ... 53
B.2.2 Immediate mode ... 53
B.2.3 Batch mode ... 53

Appendix C

Differences between Call Collector and ITM ... 55

Figures

Fig. 2.1 – Services... 3
Fig. 2.2 – Classes and components 4
Fig. 2.3 – Connection and reconnection strategy 5
Fig. 3.1 – Data flow between the IRD and the ITM... 9
Fig. 3.2 – Data flow between the ITM and the SMS... 12
Fig. 6.1 – JSim connection properties... 32
Fig. 6.2 – JSim ‘Connect’ button 32
Fig. 6.3 – JSim display after SMS commands are sent.. 33
Fig. 6.4 – JSim display after SMS commands 1002 is sent 35
Fig. 6.5 – JSim display after feedback commands are received.......................... 36
Fig. 9.1 – Two phases commit transaction mechanism....................................... 51
Fig. 9.2 – Request coordinator 52
Fig. 9.3 – Acknowledgement strategy... 52
Fig. 9.4 – Acknowledgement manager 53
Fig. 9.5 – Immediate mode 53
Fig. 9.6 – Batch mode.. 54

Tables

Table 0-1 – Acronyms and abbreviations xi
Table 1-1 – Document history 2
Table 3-1 – SQL request for callback reason.. 9
Table 3-2 – Callback reason bitmap.. 10
Table 4-1 – Environment customized parameters.. 15
Table 4-2 – Environment cutomized parameters 16
Table 4-3 – Environment expert parameters... 16
Table 4-4 – SEP global parameters for expert configuration............................... 17
Table 4-5 – SEP Event Reporting parameters for expert configuration................ 17
Table 4-6 – ITM parameters to customize 17
Table 4-7 – ITM parameters for expert configuration 19
Table 4-8 – SSM parameters to customize 20
Table 4-9 – SSM parameters for expert configuration... 21
Table 4-10 – Translator parameter for expert configuration................................ 21
Table 4-11 – Optional environment variables.. 22
Table 8-1 – Troubleshooting : SMS commection 43
Table 8-2 – Troubleshooting : negative acknowledgement returned to the SMS ... 44
Table 8-3 – Troubleshooting : no feedback command generated 45
Table 8-4 – Troubleshooting : the IRD cannot call back 45
Table 8-5 – Troubleshooting : ciphering problems... 45
Table 9-1 – ITM database backup.. 47
Table 9-2 – ITM database restore 48
Table 9-3 – ITM database purge 48
Table 9-4 – Purge criteria for ITM data.. 48
Table 9-5 – Modification of the purge parmeters 48
Table 9-6 – Glossary of terms.. 50
Table 9-7– Differences between CallCollector and ITM DNASP-2...................... 56

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.
TIP	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.

Acknowledgements

Trademarks: Any company's or product name(s) found herein may be the trademarks or registered trademarks of their respective companies.

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 of 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.

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"

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.
IRD	Integrated Receiver Decoder	Device that allows receiving 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 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 equipment.
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 period right (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 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.

Acronym Abbreviation	Definition	Description
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 "sits on top" of the TV set that allows receiving signal, demodulates it and de-scrambles it
T1		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 0-1 – Acronyms and abbreviations

1. Introduction

This user guide document provides guidance to operate the NagraVision Interactive Transaction Manager application (hereafter **ITM**).

The ITM application, as part of the NagraVision conditional access system (**CAS**), has to be considered as the secured, connection oriented interface between the terminal subscriber equipment (also known as the Integrated Receiver Decoder or **IRD**) and the head end system (**CAS**). In that sense, it enables all the return path related activities, e.g. interactive purchases, home banking, home shopping, statistical report.

The operation of the ITM application, described in this user guide document, encompasses the user operation, the maintenance and the troubleshooting aspects.

From the functional point of view, the ITM application described in the present document addresses the interactive purchase functionality.

1.1. Purpose

The objective of this user guide document is to provide the operator of the ITM application the necessary knowledge to be able to establish and maintain the operational mode of the application, as well as to be able to maintain and troubleshoot the ITM application.

1.2. Scope

The scope of the ITM application is

- **Advanced Interactivity**: possibility of using a “content on demand” service.
- **Bidirectional PrivateChannel**: point-to-point and bi-directional communication.
- **Distributed architecture**: possibility to dispatch services over the network.
- **Scalability**: easiness in the way of calculating ideal dimensions of the system.
- **High system integrity**: availability to recover the system after any crash.
- **High system security**: secured communication entry points and encrypted information.
- **Centralized system supervision**: monitoring the system may be done from one centralized console.

1.3. Audience

This guide is intended for persons that will operate the ITM product.

1.4. Prerequisite

This guide is intended for persons that have a basic understanding of the following:

- The NagraVision CAS,
- Use of Unix,
- UML diagrams.

1.5. Further reading

[1] SEP User Guide
[2] NSM Log Control User Guide
[3] SMS Gateway interface definition
[4] SEP Product Installation Check List
[5] NSM Product Installation Check List
[6] ITM DNASP-2 Software Installation Guide
[7] CC – ITM Installation Sheet
[8] Cipher DNASP-2 Installation Guide
[9] Cipher DNASP-2 User Guide
[10] SMS-SM Installation Guide
[11] SMS-SM User Guide

1.6. Document history

Version	Date	Author(s)	Description
1.2.0	20.09.2001	Jean-Noël Chabaud	• Removed information regarding the Cipher DNASP-2 and the SMS-SM applications which are now shipped within independent packages (cf. documents [9] and [11]), • Updated information regarding starting script (cf. § 4.1.9 and 4.3) and monitoring scripts (cf. § 5.5) which have been improved. • Added new parameters for the SSM.
1.1.3	17.08.2001	Jean-Noël Chabaud	Updated the SEP_HOME environment variable.
1.1.2	25.06.2001	Jean-Noël Chabaud	Added remark about the fact that SMS feedback commands are not grouped by callback date (cf. § 3.3.2)
1.1.1	20.06.2001	Jean-Noël Chabaud	Added new value '2.4' for the SMM_EISProtocol startup parameter.
1.1.0	01.06.2001	Jean-Noël Chabaud	The document is entirely reshaped. To be noticed that : • The monitoring tool 'check_itmsoft' is no more supported and its related description has been suppressed. • A new chapter (cf. § 4.1) has been added to describe every configuration parameter of all the ITM components.
1.0.2	30.05.2001	Jean-Noël Chabaud	Added Appendix C 'Differences between Call Collector and ITM'.
1.0.1	10.07.2000	Sébastien Ruffy	Added § 3.3.5 'SMS downtime management'. Added § 3.2.1 'ANI Provider'.
1.0.0	15.03.2000	Sébastien Ruffy, Nicolas Pauli, Michel Buri	First version

Table 1-1 – Document history

2. Basic business mechanisms

2.1. Services

In addition to SEP and NSM services, a set of services has been developed to meet the ITM requirements.

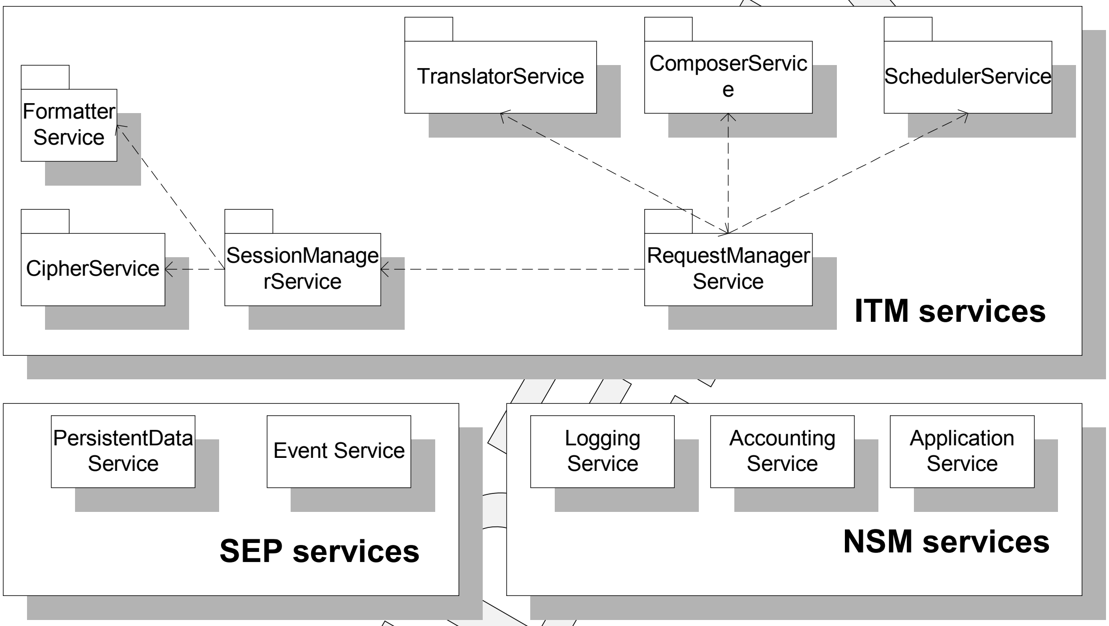


Fig. 2.1 – Services

2.1.1. ITM services

Formatter service: checks the format of commands and transforms them into internal representation.
Cipher service: secures messages.
Session manager service: manages the sessions in terms of communication and authentication.
Translator service: translates internal messages from a language to another one.
Composer service: arranges data in containers.
Scheduler service: lets create jobs with a given schedule.
Request manager service: manages the requests in term of ordering, scheduling and acknowledgements.

2.1.2. SEP services

Persistent data service: offers a mechanism to keep data in a persistent storage in case of application crash.
Event service: offers a mechanism that reports application events. See [1] for a full description of this service.

2.1.3. NSM services

Logging service: offers a mechanism that logs the events. See [2] for a full description of this service.
Account service: manages the user accounts of the CAS system.
Application service: monitors the whole CAS application.

2.2. Classes and components

The following figure shows some software classes that will be used further on. The second figure is the ITM components view.

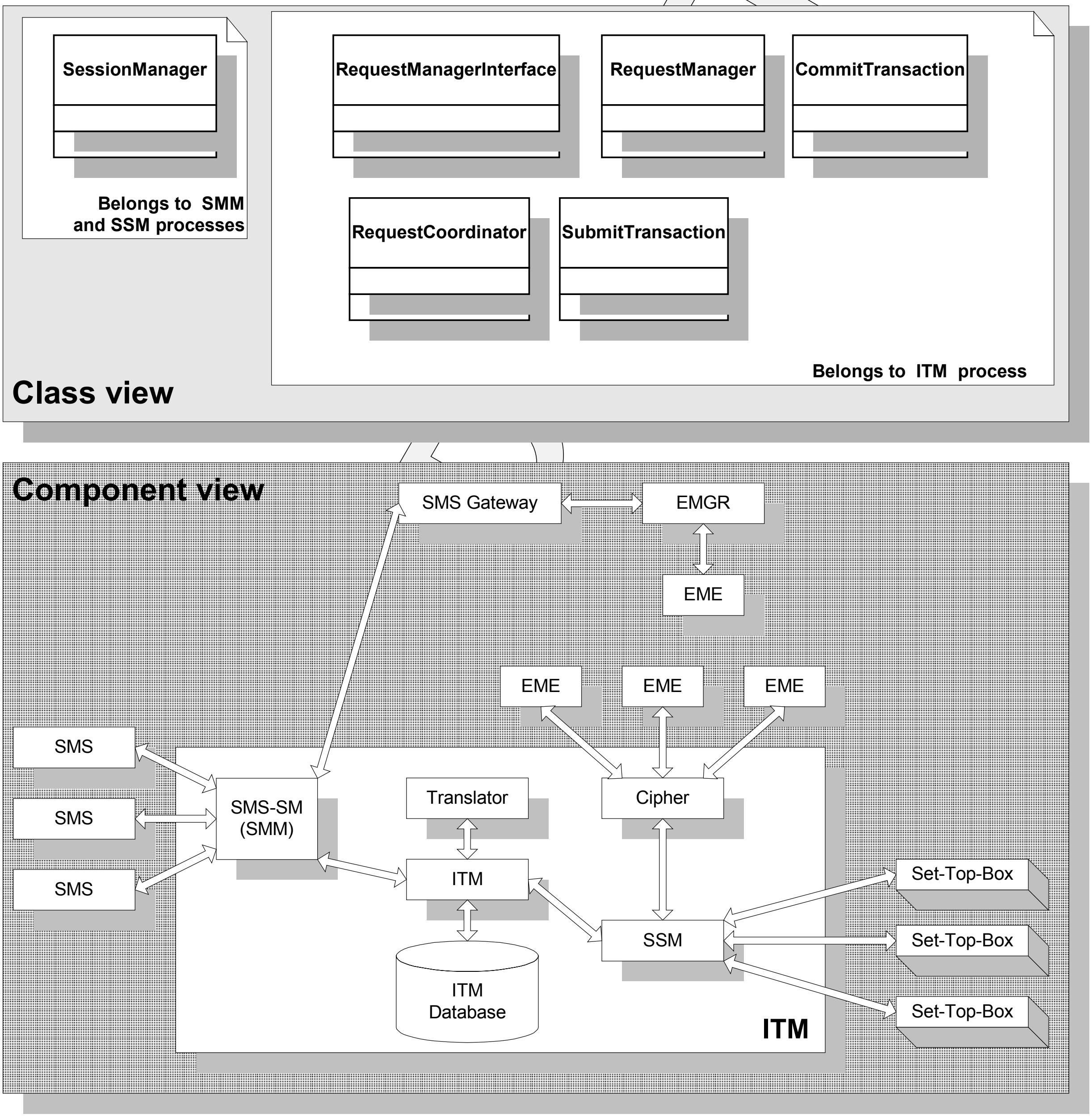


Fig. 2.2 – Classes and components

2.3. Connection and reconnection strategy

In case of communication loss, ITM components act as clients that connect to servers. In the following figure, an arrow from component c1 to c2 means c1 retries to connect to c2 when connection is not successful.

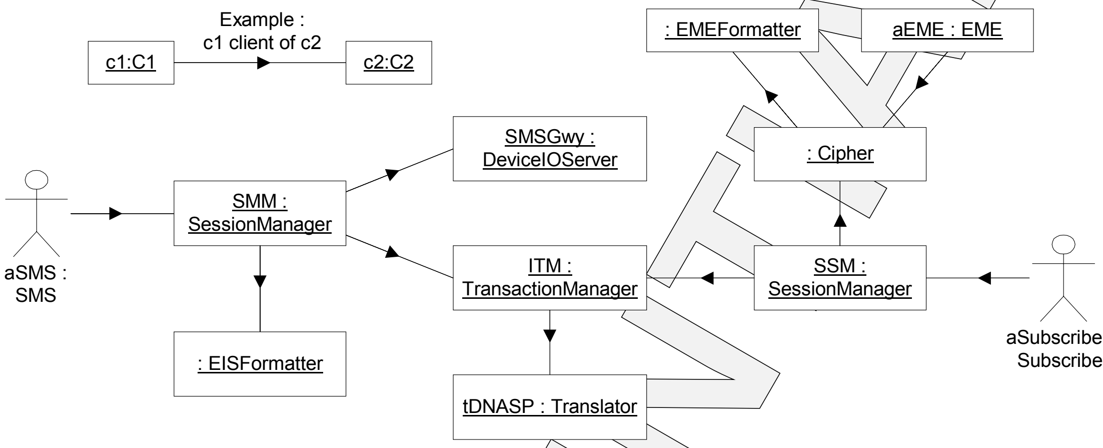


Fig. 2.3 – Connection and reconnection strategy

3. Communication with the IRD

The purpose of this section is to describe mechanisms involved in the ITM when an Integrated Receiver Decoder or IRD (sometimes also called Set Top Box or STB) calls the CAS system.

3.1. Callback data

The data involved during a callback are CCM's and EMM's. CCM's are going from the IRD towards the ITM and EMM's are issued from the ITM to the IRD. For security reasons, the description of the fields of both categories of messages (EMM's and CCM's) is beyond the scope of the present document.

3.2. Data flow between the IRD and the ITM

This section describes the data flow occurring between the IRD and the ITM during a regular callback. Another section (see § 3.3) will present the data flow between the ITM and the SMS.

3.2.1. ANI Provider

The ITM has been designed and implemented to work either with cable or phone return channel. In the first configuration, the IRD uses TCP/IP protocol over the cable network. In the second configuration, the T1 protocol is implemented over the telephone line.

In order to identify the caller, a module called 'ANI Provider' (where ANI stands for Automatic Number Identification) has been developed. The following diagrams show the modules and the sequences involved during the IRD callback for the PSTN configuration.

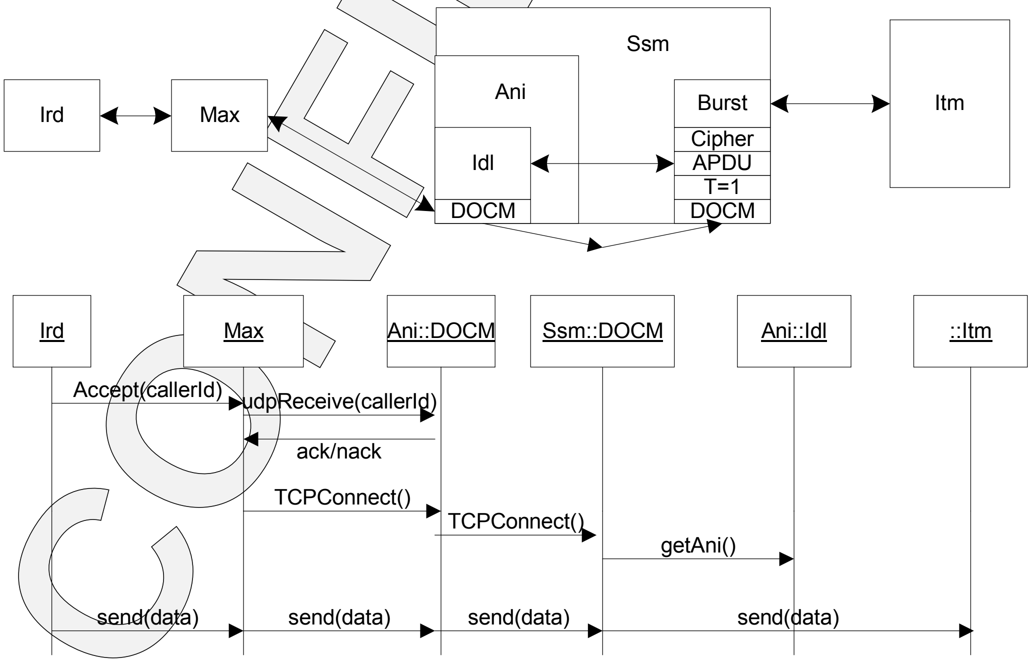


Fig. 3.11 – ANI provider

With the ANI Provider module, the phone number of the calling IRD can be obtained by the ITM and is further stored in the ITM database. This allows the ITM to generate (either immediately or periodically depending on the generation mode of feedback commands) the feedback command 205 'Phone Discrepancies' if the phone number used by the IRD is not present in the list of the authorized phone numbers.

3.2.2. IRD logon

The first thing executed by the IRD is to connect to the SSM component and to initialize a logon procedure during which the IRD must provide a user name and a password to identify itself. This user name and password must correspond to the 'SSM_loginName' and 'SSM_loginPassword' parameters of the SSM. If the login fails, the connection with the IRD is closed by the SSM. Once connected and identified, the IRD can start to send one or several CCM separated by APDU messages.

3.2.3. Message processing

The message processing mainly consists in exchanging CCM's and EMM's. The following steps are observed during a nominal session (several internal processes are not described for the sake of simplicity) :

- The IRD sends a username and a password to open the session (log on procedure).
- These values are checked and the connection is closed if wrong values are provided.
- The IRD sends one or several CCM's.
- Every CCM is deciphered by means of a deciphering request sent by the SSM to the Cipher component through its dedicated CORBA interface. If a deciphering problem occurs the connection with the IRD is closed by the SSM.
- The IRD sends a special 'EMM Request' APDU message.
- The set of deciphered CCM's previously received from the IRD is sent by the SSM to the ITM through its dedicated CORBA interface.
- The ITM processes the set of CCM's following to a 'two phases commit' mechanism (see § B.1.1 for more information). Data contained in the CCM's are stored in the ITM database. If an error occurs during this processing, the ITM will never send back any EMM to the SSM, and the IRD session will be closed by the SSM due to a timeout.
- The ITM generates a set of DNASP commands and sends it to the Translator for these commands to be translated into one or several EMM(s).
- The ITM sends the resulting set of EMM's to the SSM.
- The set of EMM's is forwarded by the SSM to the Cipher component for ciphering.
- The Cipher returns to the SSM the set of ciphered EMM's.
- The SSM forwards the set of EMM's to the IRD and closes the connection.

The following figure summarizes communication exchange involved during a callback.

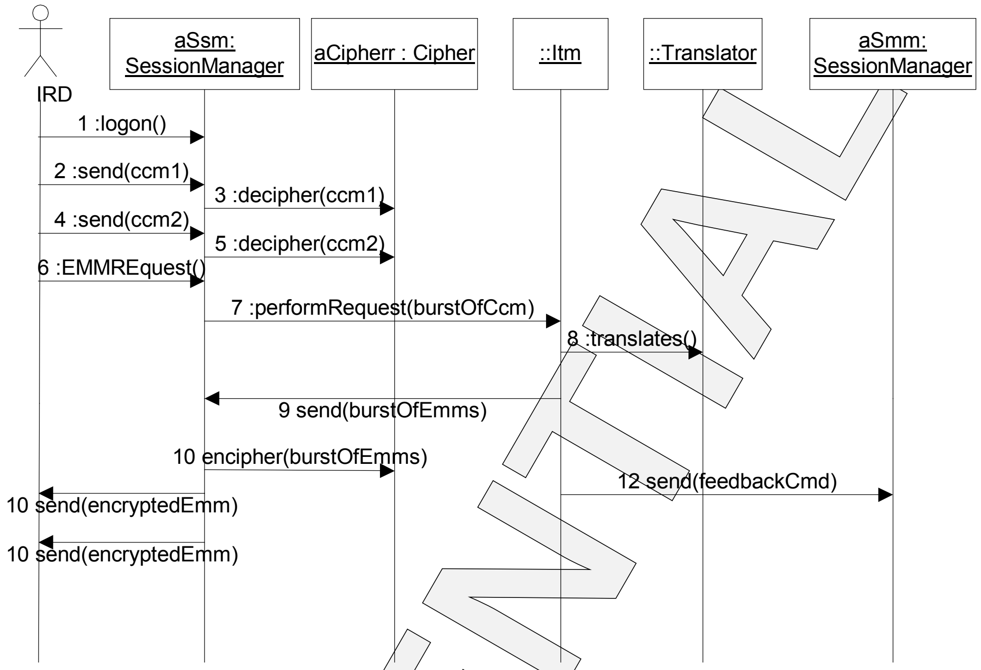


Fig. 3.1 – Data flow between the IRD and the ITM

3.2.4. Monitoring

Any relevant action (connection / disconnection, data hit, exceptions…) generates a NSM event and is possibly reported to the component traces. Additionally, it is possible to fetch some information from the database to obtain, for instance, the callback reason as described in the next paragraph.

3.2.5. Callback reasons

There are six reasons for an IRD to call the CAS system:

- *Immediate*: executed just after a request of the SMS,
- *Regular*: executed every n days,
- *Low memory*: generated if there is a lack of memory inside the ISD,
- *Low credit*: generated if the available credit is under the credit threshold,
- *Special event:* generated when a specific event is being watched,
- *Maintenance:* not described here because customer specific.

This callback reason can be obtained by means of a specific SQL request to be issued against the database, as indicated below:

Command	Description
`rlogin –l oracle itmora`	Log on to the oracle account on the Oracle server.
`export ORACLE_SID=<ITM_SID>`	Export the ORACLE_SID environment variable to point to the <ITM_SID> of the ITM database.
`sqlplus itmdba/<password>`	Open SQL*Plus session.
`SQL > select datetime, callbacktype from callback where ua = <ua_nbr> order by datetime;`	Get the list of every callback (still not purged) for a given UA. The `<ua_nbr>` field must be, of course, replaced with the real UA number. The last record is the most recent.
`SQL > quit`	Exit the SQL*Plus session.

Table 3-1 – SQL request for callback reason

The following table gives the list of the various callback reason codes that can be displayed by the previous SQL request. The returned value is a bitmap value on one byte where each bit represents a distinct callback reason (as they can be combined):

Bit	Meaning	Remark
0	Regular	
1	Immediate	
2	Low Credit	
3	Memory Full	
4	Maintenance	
5	Special Event	
6	None	Reserved for future use
7	None	Reserved for future use

Table 3-2 – Callback reason bitmap

3.3. Data flow between the ITM and the SMS (feedback commands)

This section describes the data flow occurring between the ITM and the SMS, i.e. the generation and the acknowledgement of feedback commands related to previous IRD callbacks.

3.3.1. SMS connection

In order to receive feedback commands from a given TCP port of the SMS-SM, the SMS must identify itself by issuing a command 1002 containing its source id.

Note If the SMS does not send a command 1002 with the right source id, it will not be able to receive the feedback commands.

However, there is an exception when the 'SMM_multiSMS' parameter of the SMS-SM is deactivated (i.e. the component works in a mono SMS mode). In such a circumstance, the 'SMM_uniqueSID' parameter of the SMS-SM provides the source id of the SMS and this latter doesn't need to send a SMS command 1002 to identify itself…

3.3.2. Feedback commands generation

After the ITM has processed a burst sequence of CCM's, it translates them into feedback commands and sends them to the SMS-SM. The SMS-SM then forwards these feedback commands to the right SMS.

The type and the order of feedback commands to be generated are given by the startup parameter 'ITM_FeedbackList'. If, for instance, this parameter is set to '211,201,202,206,212', the following SMS feedback commands will be generated for each UA being processed:

- One SMS command 211 'Start of Report',
- One SMS command 201 'Current Debit and Credit',
- No, one or several SMS command(s) 202 'PPV Purchase List' depending on the interactive purchase(s) effectively done by this UA.
- One SMS command 206 'STU Responding Status'
- One SMS command 212 'End of Report'

Note If the EIS protocol version being used (cf. startup parameter ‘SMM_EISProtocol’) is version 1.51, the SMS commands 211 and 212 should be suppressed from the ‘ITM_FeedbackList’ parameter as these commands are only supported with version 2.6 of the EIS protocol.

Actually, the ITM manages different ways to generate feedback commands depending on its startup parameter ‘ITM_GenerateFeedbackOnCB’ (see also § 4.1.6):

- **ITM_GenerateFeedbackOnCB = 0** : periodic mode where feedback commands are periodically generated. Every N minutes (where N is given by the startup parameter ‘ITM_FeedbackInterval’); a thread is started which scans the database for any impulsive purchase data to be reported to the SMS. In order to avoid the generation of too many feedback commands at one time, this scanning process is divided into as many slices as indicated by the ‘ITM_FeedbackHash’ parameter. For instance, if this parameter has a value of five, the first periodic generation will report feedback commands for UA 0, 5, 10, etc… the second periodic generation will report feedback commands for UA 1, 6, 11, etc… and so on. Thus, in that case, the whole database will be fully scanned after five passes of periodic generation and within a period of time around 50 minutes.

Note The duration (in minutes) for the whole database to be totally scanned is approximately given by the product of both parameters ‘ITM_FeedbackHash’ and ‘ITM_FeedbackInterval’.

- **ITM_GenerateFeedbackOnCB = 1** : immediate mode where feedback commands are generated at the end of every ISD callback. At the end of the callback, the database is scanned for any impulsive purchase data to be reported to the SMS but **only for the UA that called back**. Then feedback commands are generated and sent to the SMS-SM.
- **ITM_GenerateFeedbackOnCB = 2** : combined mode where feedback commands are generated periodically and also upon every callback occurring.

Note Whatever the feedback command generation mode is, no feedback command will be generated for a given UA if the SMS managing this UA is not currently connected to the SMS-SM component.

According the date and time information given in the SMS feedback command 211 ‘Start of Report’, the various SMS feedback commands 202 ‘PPV Purchase List’ generated between the initial SMS command 211 and the final SMS command 212 ‘End of Report’ should be related to a **unique** and **same callback** of the ISD. Actually, this is not the case in the current release of the ITM where the following rules apply :

- During the periodic generation of SMS feedback commands, the database is scanned for all unreported interactive purchases without any consideration for the date and time when these data were inserted into the database. Thus, if several callbacks are achieved by the same ISD since the last generation of feedback commands for this ISD, it may occur that interactive purchases reported by the ISD within distinct callbacks are merged into the same burst of SMS feedback commands…
- In the immediate generation mode, a burst of SMS feedback commands is usually related to the same ISD callback (because generation intervenes at the end of the callback). But, here again, it may occur that interactive purchases reported by the ISD within distinct callbacks are merged together if, for instance, the concerned SMS was down during the last generation of feedback commands…

Note It is the intention of Nagravision to modify the behavior of the ITM in order to group SMS feedback commands together into one distinct burst of SMS feedback commands for each corresponding ISD callback using the date and time of the ISD callback as discriminator.

3.3.3. Standalone generation of feedback command 206

Additionally to the generation of SMS feedback commands, the ITM application offers the possibility to periodically generate a set of SMS feedback commands 206 'STU Responding Status' for every ISD found in the ITM database with a 'late responding' status (i.e. for every ISD having a next callback date in the past). This periodic generation may be enabled or not depending on the value of the startup parameter 'ITM_Generate206' (see also § 4.1.6).

If the periodic generation of SMS feedback commands 206 'STU Responding Status' is enabled, the mechanism is similar to the one being used for the periodic generation of feedback commands. Every N minutes (where N is given by the startup parameter 'ITM_Interval206'), a thread is started which scans the database for any ISD with a 'late responding' status to be reported to the SMS. In order to avoid the generation of too many feedback commands 206 at one time, this scanning process is divided into as many slices as indicated by the 'ITM_Hash206' parameter. For instance, if this parameter has a value of five, the first periodic generation will report feedback commands 206 for UA 0, 5, 10, etc… the second periodic generation will report feedback commands for UA 1, 6, 11, etc… and so on. Thus, in that case, the whole database will be fully scanned after five passes of periodic generation.

Note The duration (in minutes) for the whole database to be totally scanned is approximately given by the product of both parameters 'ITM_Hash206' and 'ITM_Interval206'.

Note Even if the periodic generation of SMS feedback commands 206 'STU Responding Status' is enabled, no feedback command 206 will be generated for a given UA if the SMS managing this UA is not currently connected to the SMS-SM component.

3.3.4. SMS feedback commands acknowledgement

Each feedback command must be acknowledged by the SMS. If the SMS responds with a negative acknowledgement, the command is not immediately sent again by the ITM (in immediate mode, it will be sent again at the end of the next callback of the same ISD; whereas, in periodic mode, it will be sent again during the next pass processing the same UA). The following figure shows this message processing:

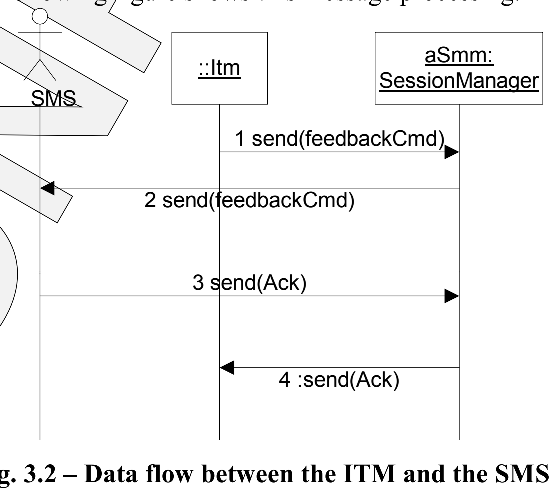


Fig. 3.2 – Data flow between the ITM and the SMS

3.3.5. SMS downtime management

For any unexpected reason, it may appear that the SMS cannot acknowledge the feedback commands. To handle such a situation, all the data necessary to generate feedback commands are persistently stored into the ITM database. So, even if a callback is done during a SMS downtime period, the corresponding feedback commands will be generated later on by the ITM.

By the way, a specific management policy has been implemented in the ITM to deal with such a situation where the SMS does not acknowledge feedback commands:

- The startup parameter ‘ITM_OutstandingAckLimit’ indicates the maximum number of non-acknowledged feedback commands that should be generated by the ITM. Once this limit is reached, no more feedback command is generated by the ITM unless some acknowledgements are received by the SMS. To be noticed that this limit is global to the whole set of SMS currently connected.
- Another startup parameter ‘ITM_OutstandingAckTime’ gives the delay after which a non-acknowledged feedback command is considered by the ITM as being negatively acknowledged.

Finally, a notification mechanism exists between the SMS-SM and the ITM for this latter to be constantly warned of which SMS are currently connected to the SMS-SM component…

3.3.6. Monitoring

The generation of feedback commands can be monitored thanks to the traces of the ITM component (see §5.5.4, 6.2.2, 6.2.3 and 6.3.2 for more information about displaying traces). Additionally, traces regarding feedback commands are generated by the SMS-SM component if the ‘SMM_FbCmdLog’ and the ‘SMM_FbAckLog’ parameters are enabled.

4. Starting and stopping components

4.1. Setting up configuration parameters

All configuration parameters of the various ITM components (i.e. ITM and SSM) are managed together in a unique configuration file named `init_itm.dat` and located in the following directory:

- If the multi-SEP feature is being used:
  - `/soft/${SERVICE_NAME}/${BUNDLE_NAME}/common_${SERVICE}/current/param`
    (e.g. `/soft/itmsoft/itm1/common_itm/current/param`)
- If the multi-SEP feature is not being used:
  - `/soft/${SERVICE_NAME}/${BUNDLE_NAME}/ITMDnasp2/current/param`
    (e.g. `/soft/itmsoft/itm1/ITMDnasp2/current/param`)

Before starting any ITM component, you should edit this configuration file to verify the configuration parameters and adapt them if necessary.

The next paragraphs give the description of all configuration parameters of the ITM components.

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).

4.1.1. Customized environment parameters

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

Parameter	Default value	Remark
BUNDLE_NAME	itm1	Name of the software bundle where the ITM software is installed.
SEP_VERSION	1.5.1_UNIX4.0g	Version of the Software Environment Platform being used.
ORACLE_VERSION	7.3.4	Version of the Oracle Database Management System.
TNS_ADMIN	/soft/${SERVICE_NAME}/${BUNDLE_NAME}/common_${SERVICE}/current/param	Environment variable used by Oracle to determine the location of the network configuration file `tnsnames.ora`. This definition can be commented out if the environment variable is defined elsewhere.

Table 4-1 – Environment customized parameters

4.1.2. System parameters

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

Parameter	Default value	Remark
ORACLE_HOME	/soft/local/oracle/product/	This definition can be commented out if the

Parameter	Default value	Remark
	${ORACLE_VERSION}	environment variable is defined elsewhere.
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.
TM_HOME	/soft/${SERVICE_NAME}/${BUNDLE_NAME}/tmutil/current	TM_HOME is needed to find the TM utility scripts.
ITM_HOME	/soft/${SERVICE_NAME}/${BUNDLE_NAME}/ITMDnasp3/current	This definition can be commented out if the environment variable is defined elsewhere.
ROOT_DATA	/users/${SERVICE_NAME}/${BUNDLE_NAME}/data	Location of the log files (by default log files of all applications are centralized there).
thisHost	$HOSTNAME	Host name for internal use.

Table 4-2 – Environment cutomized parameters

4.1.3. Environment expert parameters

The following table gives the list of the environment 'expert' parameters that shouldn't be normally modified:

Parameter	Default value	Remark
NLS_LANG	""	The NLS_LANG Oracle variable should be reset to prevent connection problems with the database.
id	"<id>"	Instance identifier for internal use. Do not modify this parameter!
PATH	$PATH:$TM_HOME/shell	The PATH is complemented with the location of TM utility scripts.
LD_LIBRARY_PATH	$TM_HOME/shlib:$SEP_HOME/shlib:$ORACLE_HOME/lib:$VISIBROKER_HOME/param	Path where shared libraries can be found
instanceId	${BUNDLE_NAME}${thisHost}	Instance identifier being used in CORBA name of every application.
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 4-3 – Environment expert parameters

4.1.4. 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.
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

Parameter	Default value	Remark
		the ITM DNASP-2 application.
SEP_eventDbPath	$SEP_HOME/param/	Location of the NSM events database.
SEP_defaultMOVerbosity	LOW	Minimum level for NSM events to be generated.

Table 4-4 – SEP global parameters for expert configuration

4.1.5. 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 that 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 4-5 – SEP Event Reporting parameters for expert configuration

4.1.6. ITM parameters

The following table gives the list of the configuration parameters necessary to the ITM component:

Parameter	Default value	Remark
ITM_appInstanceName	${id}	ITM application instance name.
ITM_twoTask	itm1net.world	Name of the database the ITM must connect to. This parameter must reference to the name of a section within the `$TNS_ADMIN/tnsnames.ora` configuration file.
ITM_oracleUser	itmdba	Oracle name to be used by ITM to connect to the database.
ITM_oraclePass	nagra	Oracle password to be used by ITM to connect to the database.
ITM_generateFeedbackOnCB	1	Generation mode of feedback commands (cf. § 3.3.2). Possible values are: 0 (periodic mode), 1 (immediate mode), 2 (combined mode).
ITM_feedbackList	211,201,202,206,212	List of feedback (2xx) commands to be generated. If protocol EIS 1.51 is used, feedback commands 211 and 212 should be removed from the list as they don’t exist in this version of the EIS protocol.
ITM_feedbackCmdsPPID	12345	PPID to use when reporting feedback commands in periodic mode.

Table 4-6 – ITM parameters to customize

The following table gives the list of the ‘expert’ configuration parameters necessary to the ITM component:

Parameter	Default value	Remark
ITM_outstandingAckLimit	1000	Max number of feedback commands waiting for an acknowledgement. Once this number is reached, no more feedback commands are generated by the ITM.
ITM_outstandingAckTime	1	Timeout for non-acknowledged feedback commands [hours].

Parameter	Default value	Remark
		Once this timeout occurs, the related feedback commands are considered as negatively acknowledged and are discarded. They will be generated again later on.
ITM_phoneCheckLen	10	Number of digits to be compared between phone numbers when generating feedback command 205. The phone number comparison is done on numerical characters only, starting from the end.
ITM_regularCBRule	61	Bitmap value indicating which callback reasons will trigger the regular callback date update. **Warning! This parameter is not used in ITM DNASP-2 but only concerns ITM DNASP-3.**
ITM_updateCreditRule	61	Bitmap value indicating which callback reasons will trigger the credit amount update. **Warning! This parameter is not used in ITM DNASP-2 but only concerns ITM DNASP-3.**
ITM_generate206	false	Enable periodic generation of feedback command 206 'STU Responding Status'. If set to true, this parameter enables the ITM to generate periodically this feedback command for all UAs that are in late responding status. This generation is totally independent from the possible periodic generation of other feedback commands (see § 0 for more information).
ITM_hash206	48	Hashing value applied on UA numbers for the periodic generation of feedback command 206 'STU Responding Status' (see § 0 for the description of this parameter).
ITM_interval206	30	Interval [in minutes] between two successive generations of feedback command 206 'STU Responding Status'. Minimum value is 1. If the value is greater than 30, the maximum interval of 60 minutes is used instead.
ITM_lateOffset	3600	Late responding authorized offset [sec], i.e. the authorized offset between regular and effective callback time.
ITM_feedbackHash	72	Hashing value applied on UA numbers for periodic feedback command generation (see § 3.3.2 for the complete description of this parameter).
ITM_feedbackInterval	10	Interval [in minutes] between two successive generations of feedback commands (minimum value is 1). If the value is greater than 30, the maximum interval of 60 minutes is used instead.
ITM_checkCCMIRDSN	false	When set to true, reject the callback if pairing is incorrect.
ITM_checkCCMDate	true	When set to true, reject the callback if the date of the CCM has not been incremented since the last callback. **Warning! This parameter is not used in ITM DNASP-2 but only concerns ITM DNASP-3.**
ITM_handleSMSCommandsNotStrictly	true	Accept SMS commands even if the specified IRD (i.e. UA) does not exist in the database. Any SMS command related to a non-existing IRD will then automatically create this IRD in the database.
ITM_historyOffset	60	History offset [days] for SMS command 111 'GetHistory from Call Collector'.
ITM_crPeriod	180	Callback retry period [minutes] sent to the ISD, i.e. delay to be observed by the ISD before it tries to call back again if the current callback fails.
ITM_SMMInterface	${id}SMM_IF	CORBA name of the SMS-SM interface.
ITM_SSMInterface	${id}SSM_IF	CORBA name of the SSM interface.
ITM_eventQueueMaxSize	500	Event queue max size for Event Reporting application.
ITM_translator	${id}ITMTranslator	Translator name (external to the ITM DNASP-2 process whereas it is internal with the ITM DNASP-3).

Parameter	Default value	Remark
ITM_ssmHdlPoolSize	15	Size of the thread pool reserved for SSM Handler Manager. For optimal performances, this parameter should be given the same value as the 'SSM_outgoingThreads' parameter of the SMS-SM.
ITM_smmHdlPoolSize	5	Size of the thread pool reserved for SMS-SM Handler Manager. For optimal performances, this parameter should be given the same value as the 'SMM_itmOutgoingThreads' parameter of the SMS-SM.
ITM_ssmCoordPoolSize	${ITM_ssmHdlPoolSize}	Size of the thread pool reserved for SSM Coordinators. For performance reasons, this parameter is usually given the same value as the 'ITM_ssmHdlPoolSize' parameter.
ITM_smmCoordPoolSize	${ITM_smmHdlPoolSize}	Size of the thread pool reserved for SMM Coordinators. For performance reasons, this parameter is usually given the same value as the 'ITM_smmHdlPoolSize' parameter.
ITM_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 ITM. **Warning! This parameter is not used in ITM DNASP-2 but only concerns ITM DNASP-3.**
ITM_minEventSeverity	INFO	Minimum severity required to generate an event. The authorized values are: INFO, WARNING, ERROR and FATAL. **Warning! This parameter is not used in ITM DNASP-2 but only concerns ITM DNASP-3.**
ITM_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. **Warning! This parameter is not used in ITM DNASP-2 but only concerns ITM DNASP-3.**
ITM_eventOnCout	false	Display events on standard output. If enabled, this option discards the generation of NSM events! **Warning! This parameter is not used in ITM DNASP-2 but only concerns ITM DNASP-3.**
ITM_moDictionaryFileName	${ITM_HOME}/param/itm_counters.dat	Name of the ITM 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… **Warning! This parameter is not used in ITM DNASP-2 but only concerns ITM DNASP-3.**
ITM_dumpObjects	false	Dump business messages (mainly SMS commands and acknowledgments) to log files.
ITM_OAthreadStackSize	100000	Stack size allocated to threads created by the BOA of Visibroker.
ITM_OAthreadMax	0	Maximum number of concurrent client threads allowed for Visibroker (usually 0, i.e. unlimited).
ITM_translatorBindTimeout	15	Timeout when connecting to the Translator [sec].
ITM_translatorSendTimeout	30	Timeout when sending data to the Translator [sec].
ITM_translatorReceiveTimeout	120	Timeout when receiving data from the Translator [sec].

Table 4-7 – ITM parameters for expert configuration

4.1.7. SSM parameters

The following table gives the list of the configuration parameters necessary to the SSM component:

Parameter	Default value	Remark
SSM_appInstanceName	${id}	SSM application instance name.
SSM_ani	0	Enable Radius protocol with the MAX (set for PSTN only).
SSM_aniPort	60005	Radius protocol port number.
SSM_login	1	Activation of login check (set for cable only).
SSM_loginName	nagra	Authorized username for IRD connection in either cable or PSTN configuration
SSM_loginPassword	nagra	Authorized password for IRD connection in either cable or PSTN configuration
SSM_address	$thisHost	IP address the IRD or MAX connects to.
SSM_port	2002	Port number the IRD connects to.

Table 4-8 – SSM parameters to customize

The following table gives the list of the 'expert' configuration parameters necessary to the SSM component:

Parameter	Default value	Remark
SSM_irdInterface	${id}IRDWrapper	Name of the IRD interface.
SSM_cipher	${CIP_appInstanceName}	Name of the CORBA interface of the Cipher.
SSM_itm	${ITM_SSMInterface}	Name of the CORBA interface of the ITM.
SSM_outgoingThreads	15	Number of threads sending data to ITM.
SSM_incomingThreads	15	Number of threads receiving data from ITM.
SSM_eventQueueMaxSize	500	Event queue max size for Event Reporting application.
SSM_name	${SSM_appInstanceName}	SSM name
SSM_itmBindTimeout	15	Timeout when connecting to the ITM [sec]
SSM_itmSendTimeout	30	Timeout when sending data to the ITM [sec]
SSM_itmReceiveTimeout	120	Timeout when receiving data from the ITM [sec]
SSM_cipherBindTimeout	15	Timeout when connecting to the Cipher [sec]
SSM_cipherSendTimeout	30	Timeout when sending data to the Cipher [sec]
SSM_cipherReceiveTimeout	120	Timeout when receiving data from the Cipher [sec]
SSM_mo	true	Activation of Managed object.
SSM_maxSessions	200	Maximum allowable simultaneous IRD sessions.
SSM_inputThreads	15	Number of threads in the DOCM (this value was previously accessible only through the DOCM Managed Object interface).
SSM_aniPoolSize	500	Size of the ANI Provider internal pool to store open sessions.
SSM_aniQueueSize	96	Size of the ANI Provider incoming message buffer.
SSM_aniTimeout	5	Timeout in second for incoming NAS TCP connect request on ANI Provider [sec].
SSM_aniSecret	NAGRA	Radius secret used by ANI Provider to secure data exchange with Radius client
SSM_aniThreads	15	ANI Provider thread pool size
SSM_timeout	60	ITM answer timeout [sec]. If this timeout occurs the connection with the IRD is closed.

Parameter	Default value	Remark
SSM_connectionTimeout	60	IRD connection timeout [sec] regarding the overall connection duration with the IRD. If this timeout occurs, the connection with the IRD is closed.
SSM_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 SSM. **Warning! This parameter is not used in ITM DNASP-2 but only concerns ITM DNASP-3.**
SSM_minEventSeverity	INFO	Minimum severity required to generate an event. The authorized values are: INFO, WARNING, ERROR and FATAL. **Warning! This parameter is not used in ITM DNASP-2 but only concerns ITM DNASP-3.**
SSM_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. **Warning! This parameter is not used in ITM DNASP-2 but only concerns ITM DNASP-3.**
SSM_eventOnCout	false	Display events on standard output. If enabled, this option discards the generation of NSM events! **Warning! This parameter is not used in ITM DNASP-2 but only concerns ITM DNASP-3.**
SSM_moDictionaryFileName	${ITM_HOME}/param/ssm_counters.dat	Name of the SSM 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… **Warning! This parameter is not used in ITM DNASP-2 but only concerns ITM DNASP-3.**
SSM_OAthreadStackSize	100000	Stack size allocated to threads created by the BOA of Visibroker.
SSM_OAthreadMax	0	Maximum number of concurrent client threads allowed for Visibroker (usually 0, i.e. unlimited).

Table 4-9 – SSM parameters for expert configuration

4.1.8. Translator parameter

The following table gives the list of the ‘expert’ configuration parameters necessary to the Translator component:

Parameter	Default value	Remark
Translator_name	${id}	Translator application instance name.

Table 4-10 – Translator parameter for expert configuration

4.1.9. Optional environment variables

There is no need to set any environment variable elsewhere than in the service configuration script *init_itm.dat*. However, the following optional environment variables may be set to diagnose the starting steps of the ITM DNASP-2 application.

Variable name	Description
TM_PARAM_ID	If this environment variable is set : • The file *init_<bundle_type>${TM_PARAM_ID}.dat* located in */soft/<service>/<bundle name>/common_<bundle_type>/current/param* is used instead of *init_<bundle_type>.dat* • The location of the ITM DNASP-2 data directory is set to: */users/<service>/<bundle name>/data/${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 ITM DNASP-2 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 ITM DNASP-2 application).
SHOW_INIT_ENV	If set to 'FULL' all the defined environment variables are displayed when starting the ITM DNASP-2 application.

Table 4-11 – Optional environment variables

4.2. Starting up components

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_itm` must be typed `start_itm_itm1` if the bundle name is `itm1`.

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

The command `start_itmsoft` starts the whole set of ITM components. This command sequentially launches:

- The Cipher component that acts as an interface with one or several EME(s),
- The SMS-SM component ready to accept SMS commands,
- The SSM component ready to accept IRD calls,
- The Translator component ready to translate DNASP commands into EMM's,
- The ITM component ready to process IRD callbacks and to generate SMS feedback commands.

Note

Each of these components can be independently started with a distinct command :

- `start_CipherDnasp2`
- `start_smm`
- `start_ssmdnasp2`
- `start_translator`
- `start_itmdnasp2`

Note

All startup scripts (start_itm, start_er, etc.) use the generic **_startApp** command and inherit 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 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 § 5.5 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.

4.3. Stopping components

The command `stop_itmsoft` stops the whole set of ITM components. This command sequentially stops:

- The Cipher component,
- The SSM component,
- The SMS-SM component,
- The ITM component,
- The Translator component.

Note Each of these components can be independently stopped with a distinct command :

- `stop_CipherDnasp2`
- `stop_ssmdnasp2`
- `stop_smm`
- `stop_itmdnasp2`
- `stop_translator`

After all ITM components are stopped, it is possible to stop the Event Reporting application with the command `stop_er`.

5. 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 components' output.

5.1. 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_itm2
%_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 (1.5.1-D4.0g)
TM_HOME          /soft/itmsoft/itm2/tmutil/current (1.7.2-D4.0g)
ORACLE_HOME      /soft/local/oracle
TNS_ADMIN        /soft/itmsoft/itm2/common_itm/current/param
TWO_TASK
ORACLE_USER
ORACLE_PASS
TM_PARAM_ID
instanceId       itm2IT2CAS01
startAppArgs     -i itm2IT2CAS01 -c -L/users/itmsoft/itm2/data -eCLOG:200 -b1

PATH
                 /users/local/operator/bin
                 /usr/bin
                 /usr/bin/X11
                 /usr/local/bin
                 /usr/sbin
                 /sbin
                 .
                 /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/lib
                 /soft/local/visibroker_33/current/param
```

5.2. Checking the ITM processes

The command `sys_itm` displays the ITM processes with their memory and the elapsed time since they were started.

Note

Each of these components can be independently checked with a distinct command :

- `sys_itmdnasp2`
- `sys_translator`
- `sys_ssmdnasp2`
- `sys_smm`
- `sys_CipherDnasp2`

`The command` **`sysapp_<Bundle_name>`** shows all SEP based processes (ITM DNASP-2, Cipher DNASP-2, SMS-SM, SSM DNASP-2, Event Reporting, etc.).

All these commands can be specified 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_itm2`**

```
IT2CAS01>operator% sysapp_itm2
pid user         instance       application              vsiz  rsiz   cpu

24746 operator     itm1IT2CAS01   EventReportingApp       17.9M  1.8M   0.0
21455 operator     itm2IT2CAS01   CipherDnasp2App         28.3M  6.8M   8.6
21323 operator     itm2IT2CAS01   EventReportingApp       14.4M  3.3M   0.2
21446 operator     itm2IT2CAS01   ITMDnasp2App            51.0M  9.5M  19.1
20411 operator     itm2IT2CAS01   SMMApp                  26.9M  7.7M  32.0
21398 operator     itm2IT2CAS01   SSMDnasp2App            28.8M  7.1M  13.2
21397 operator     itm2IT2CAS01   Translator              13.8M  3.3M   0.6
```

- **`sysapp_itm2 -n ITM -a`**

```
IT2CAS01>operator% sysapp_itm2 -n ITM -a

pid user         instance       application              vsiz  rsiz   cpu
21446 operator     itm2IT2CAS01   ITMDnasp2App            53.7M  9.7M   0.0
###
### Application: /soft/itmsoft/itm2/ITMDnasp2/current/bin/ITMDnasp2App
###
OAthreadMax               = 0
OAthreadStackSize         = 100000
ORBagentport              = 14001
SMS_SMInterface           = itm2IT2CAS01SMM_IF
SSMInterface              = itm2IT2CAS01SSM_IF
appInstanceName           = itm2IT2CAS01
compareSTUWhenCB          = false
crPeriod                  = 180
defaultMOVerbosity        = LOW
dumpObjects               = false
eventDbPath               = /soft/imssoft/operator/sep/current/param/
eventQueueMaxSize         = 500
eventReportingName        = itm2IT2CAS01ER
feedbackCmdsPPID          = 12345
feedbackHash              = 72
feedbackList              = 211,201,202,206,212
generate206               = false
generateFeedbackOnCB      = 0
handleSMSCommandsNotStrictly = true
hash206                   = 48
historyOffset             = 60
interval206               = 30
intervalFB                = 10
lateOffset                = 3600
moEventBodyMax            = 1024
moEventOnCerr             = 1
moGlobalPath              = /ITM/
oraclePass                = nagra
oracleUser                = itmdba
outstandingAckLimit       = 1000
outstandingAckTime        = 1
```

```
phoneCheckLen            = 10
sepInfo                  = NONE
smmCoordPoolSize         = 5
smmHdlPoolSize           = 5
ssmCoordPoolSize         = 15
ssmHdlPoolSize           = 15
translator               = itm2IT2CAS01ITMTranslator
translatorBindTimeout    = 15
translatorReceiveTimeout = 120
translatorSendTimeout    = 30
twoTask                  = itm1net.world
```

5.3. Checking the ACS processes

The command `sysacs` displays the ACS processes (i.e. the SMS Gateway, the EMGR and possibly the PA processes) with their memory and the elapsed time since they were started.

5.4. Checking the whole application (obsolete)

Due to technical reasons, the script `check_itmsoft` cannot be maintained with the current release of the ITM DNASP-2 and has been removed from the package...

5.5. Monitoring the application

Depending on the startup parameter being used when starting the ITM DNASP-2 application, the traces can be managed in the various following ways:

- No trace is generated,
- Traces are sent to a regular log file,
- Traces are sent to a circular log file (default mode),
- Traces are sent to a revolving file set,
- Traces are broadcast on the network within UDP packets.

Independently from the kind of trace management being used, NSM events are also sent by the ITM components and these events can be displayed with the NSM tools. Please refer to [2] for more information….

The following paragraphs describe the various ways to access the application's output...

5.5.1. 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 `$ROOT_DATA/log` (when the default value of the –L option is used). 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/ITMDnasp3App.dia`

5.5.2. 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 to not 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/ITMDnasp3App.clog
```

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

5.5.3. 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 ITM DNASP-2, the log file names will be `ITMDnasp3App.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.

5.5.4. 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 4010 (i.e. when option '-e UDP:<node_name>:4010:ITM' is specified when starting the ITM DNASP-2):

- `read_udp_itm1 4010`

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

5.6. Nagravision System Management (NSM) interface

The ITM DNASP-2 allows several system management actions to be performed while it is active. The ITM managed object (MO) implements an interface that allows:

- Checking whether the ITM database can be accessed (currently not implemented),
- Checking whether the Translator is connected,
- Getting the number of seconds attributed to the Translator connection, send or receive timeouts (i.e. the value of the 'ITM_translatorBindTimeout', 'ITM_translatorSendTimeout' or 'ITM_translatorReceiveTimeout' parameter),
- Getting a list of interface names corresponding to connected SMS-SM (currently not implemented),
- Getting a list of interface names corresponding to connected SSM (currently not implemented).

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

- Number of requests received from the SMS-SM, i.e. the number of SMS commands,
- Number of requests received from the SSM, i.e. the number of CCM coming from IRD,
- Number of Submit Coordinators which processing has succeeded or failed and the total of both,
- Number of Commit Coordinators which processing has succeeded or failed and the total of both,
- Number of feedback commands positively acknowledged,
- Number of feedback commands negatively or not acknowledged,
- Total number of feedback commands generated.

6. Example of use

6.1. Checking the ITM components

Before sending and tracing a SMS command, you must ensure that all the ITM components are up and running. For this, two things may be done:

- Use the command `sys_itm_<bundle_name>` to check that all the ITM components are running,
- Either connect a subscriber to the Event Reporting application (the command `show_events` can be used) and check that no error event is generated by any component or display the traces generated by any component thanks to the appropriate command as indicated in § 5.5.

6.2. Use case : SMS commands processing

This section shows how the SMS commands are going through the CAS system. Another section will describe a use case regarding the feedback command generation.

6.2.1. Use case prerequisites

Prior to execute this test, the very last SMS simulator JSim (version 2.1.2 or above) should be installed on any available Windows NT workstation. This SMS simulator can be obtained from Nagravision support team if necessary. At the same time, the ITM database, the SMS-SM, the Cipher DNASP-2, and the ITM DNASP-2 applications have to be installed as described in documents [10], [8] and [6].

6.2.2. Displaying traces of the SMS-SM

In a new terminal, enter the command `clog_<bundle_name> -rf <ROOT_DATA>/SMMApp.clog` to display the traces of the SMS-SM component.

6.2.3. Displaying traces of the ITM

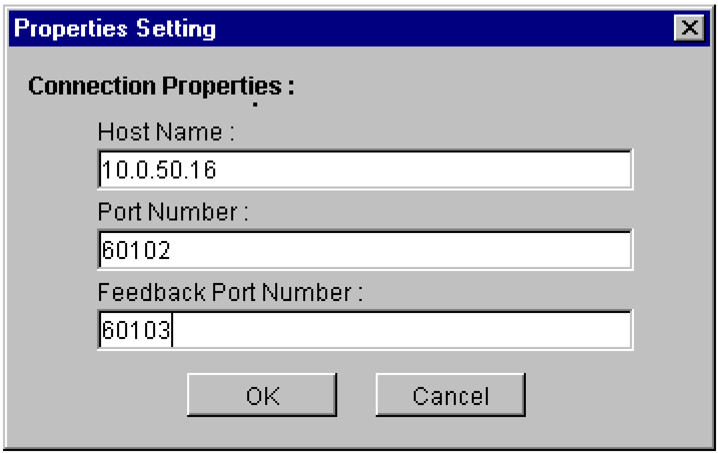
In a new terminal, enter the command:

```
clog_<bundle_name> -rf <ROOT_DATA>/ITMDnasp2App.clg
```

to display the traces of the ITM component.

6.2.4. Connecting to the SMS-SM

Before sending any SMS command, the SMS simulator must connect to the SMS-SM component. As a prerequisite, you must define the hostname (or IP address) and port number where the SMS simulator must connect to. The connection parameters can be set using the option ‘Properties’ in the ‘Tools’ menu:



Properties Setting

Connection Properties :

Host Name : 10.0.50.16

Port Number : 60102

Feedback Port Number : 60103

Fig. 6.1 – JSim connection properties

Once these parameters are set, the only thing to do is to click on the 'Connect' button shown below:

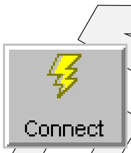


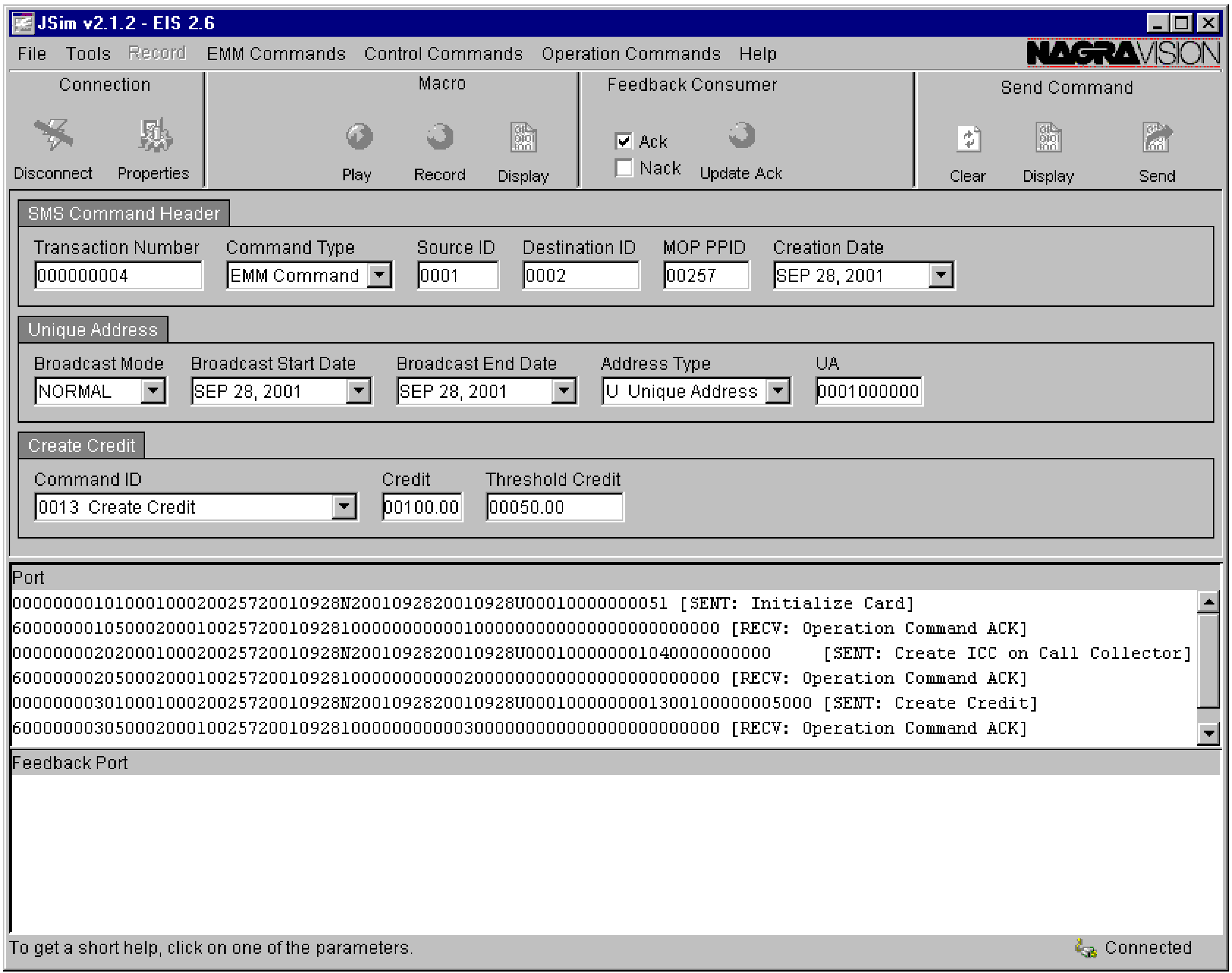
Fig. 6.2 – JSim 'Connect' button

6.2.5. Sending SMS commands

For this use case, the following SMS commands are successively sent to the SMS-SM component:

- SMS EMM command 51 'Initialize Card',
- SMS CONTROL command 104 'Create ICC on Call Collector',
- SMS EMM command 13 'Create Credit for Impulse Purchase'.

The following traces are visible in the JSim display:



SMS Command Header

Transaction Number	Command Type	Source ID	Destination ID	MOP PPID	Creation Date
000000004	EMM Command	0001	0002	00257	SEP 28, 2001

Unique Address

Broadcast Mode	Broadcast Start Date	Broadcast End Date	Address Type	UA
NORMAL	SEP 28, 2001	SEP 28, 2001	U Unique Address	0001000000

Create Credit

Command ID	Credit	Threshold Credit
0013 Create Credit	00100.00	00050.00

Port

```
00000000101000100020025720010928N2001092820010928U00010000000051 [SENT: Initialize Card]
6000000010500020001002572001092810000000000001000000000000000000000000 [RECV: Operation Command ACK]
00000000202000100020025720010928N2001092820010928U0001000000010400000000000     [SENT: Create ICC on Call Collector]
6000000020500020001002572001092810000000000002000000000000000000000000 [RECV: Operation Command ACK]
00000000301000100020025720010928N2001092820010928U0001000000001300100000005000 [SENT: Create Credit]
6000000030500020001002572001092810000000000003000000000000000000000000 [RECV: Operation Command ACK]
```

Feedback Port

To get a short help, click on one of the parameters.

Connected

Fig. 6.3 – JSim display after SMS commands are sent

At the same time, the following traces are generated by the SMS-SM component:

```
|EVENT: /ITM/jncheraSMSWrapperMO  28 Sep 2001 14:13:26 MEDIUM | ClassName: 'SMSSMApp' -
MethodName: 'initialize' - Reason: 'I-INITSMM-The SMM application is up and ready.'
|none      MESSAGE      1 28 Sep 2001 14:13:26 ::                    main            HIGH   |
##### Entering SEP::Process::run() ...
|EVENT: /ITM/jncheraSMSWrapperMO  28 Sep 2001 14:14:10 LOW    | ClassName: 'SMSSvcHandler' -
MethodName: 'SMSSvcHandler' - Reason: 'I-SMSSVCHDL-SMS connection detected.'
|EVENT: /ITM/jncheraSMSWrapperMO  28 Sep 2001 14:14:10 LOW    | ClassName: 'SMSSvcHandler' -
MethodName: 'SMSSvcHandler' - Reason: 'I-SMSSVCHDL-SMS connection detected.'
|EVENT: /ITM/jncheraSMSWrapperMO  28 Sep 2001 14:14:47 LOW    | SMS
command:'00000000101000100020025720010928N2001092820010928U00010000000051'
|EVENT: /ITM/jncheraSMSWrapperMO  28 Sep 2001 14:15:34 LOW    | ClassName: 'SMSSession' -
MethodName: 'controlActive' - Reason: 'I-SMSSESS-SMS with source id: 0001 is active on
control port.'
|EVENT: /ITM/jncheraSMSWrapperMO  28 Sep 2001 14:15:34 LOW    | ClassName: 'SMSSession' -
MethodName: 'state' - Reason: 'I-SMSSESS-Persistent queues for SMS with source id: 0001 are
being emptied.'
|EVENT: /ITM/jncheraSMSWrapperMO  28 Sep 2001 14:15:34 LOW    | ClassName: 'SMSSession' -
MethodName: 'state' - Reason: 'I-SMSSESS-SMS with source id: 0001 is active on control port
only.'
|EVENT: /ITM/jncheraSMSWrapperMO  28 Sep 2001 14:15:34 MEDIUM | ClassName:
'InternalPropagatorReader' - MethodName: 'setActive' - Reason: 'I-INTPROP-Session for SMS
with source id: 0001 is active on the control port.'
|EVENT: /ITM/jncheraSMSWrapperMO  28 Sep 2001 14:15:37 LOW    | SMS ack
:'6000000010500020001002572001092810000000000001000000000000000000000000'
```

```
|EVENT: /ITM/jncheraSMSWrapperMO  28 Sep 2001 14:16:05 LOW    | SMS
command:'00000000202000100020025720010928N2001092820010928U0001000000001040000000000    '
|EVENT: /ITM/jncheraSMSWrapperMO  28 Sep 2001 14:16:05 LOW    | SMS ack
:'6000000020500020001002572001092810000000000002000000000000000000000000'
|EVENT: /ITM/jncheraSMSWrapperMO  28 Sep 2001 14:16:34 LOW    | SMS
command:'00000000301000100020025720010928N2001092820010928U00010000000001300100000005000'
|EVENT: /ITM/jncheraSMSWrapperMO  28 Sep 2001 14:16:34 LOW    | SMS ack
:'6000000030500020001002572001092810000000000003000000000000000000000000'
```

6.3. Use case : feedback commands processing

This section shows how feedback commands are generated by the ITM component upon processing an IRD callback. All ITM components are involved in this use case.

6.3.1. Use case prerequisites

Prior to execute this test, the very last SMS simulator JSim (version 2.1.2 or above) should be installed on any available Windows NT workstation. This SMS simulator can be obtained from Nagravision support team if necessary. At the same time, the ITM database, the SMS-SM, the Cipher DNASP-2, and the ITM DNASP-2 applications have to be installed as described in documents [10], [8] and [6].

Note that the test described hereafter was done using an IRD simulator as well as an EME simulator. As these pieces of software are highly security sensitive, they cannot be delivered to any customer and real hardware should be used by any customer realizing a similar test.

6.3.2. Displaying traces of the ITM components

In several new terminal windows, enter the following commands to display the traces of the corresponding components:

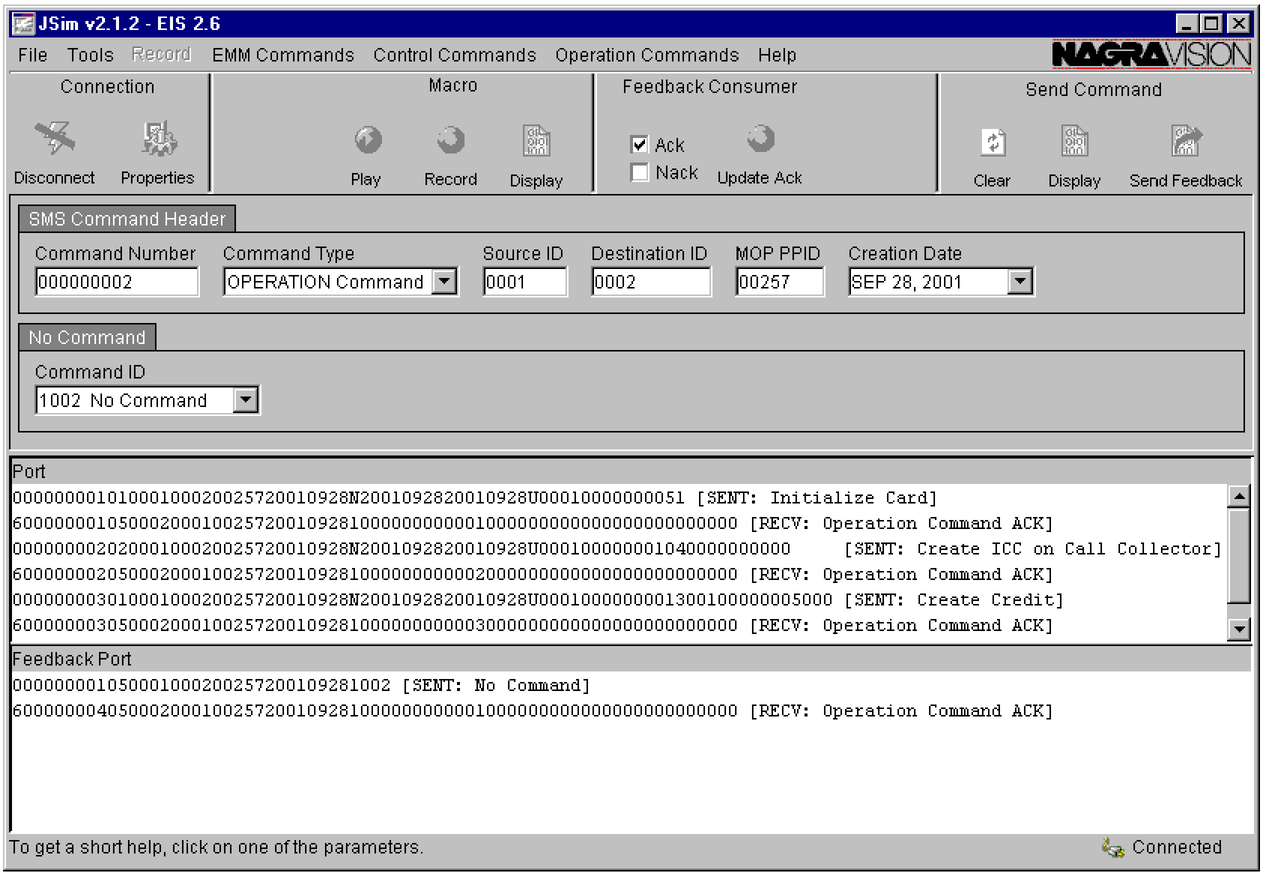
- `clog_<bundle_name> –rf <ROOT_DATA>/SSMDnasp2App.clog`
- `clog_<bundle_name> –rf <ROOT_DATA>/CipherDnasp2App.clog`
- `clog_<bundle_name> –rf <ROOT_DATA>/ITMDnasp2App.clog`
- `clog_<bundle_name> –rf <ROOT_DATA>/SMMApp.clog`

6.3.3. Connecting to the SMS-SM

Before receiving any feedback command, the SMS simulator must connect to the SMS-SM component. The same procedure as the one described in § 6.2.4 must be applied.

After the SMS simulator is connected, an SMS OPERATION command 1002 ‘No Command’ must be sent to the SMS-SM that will be forwarded to the ITM for it to know that feedback commands can be generated for this SMS system (identified by its source id).

The following traces appear in the JSim simulator window:



SMS Command Header

Command Number: 000000002
Command Type: OPERATION Command
Source ID: 0001
Destination ID: 0002
MOP PPID: 00257
Creation Date: SEP 28, 2001

No Command

Command ID: 1002 No Command

Port

```
00000000101000100020025720010928N2001092820010928U00010000000051 [SENT: Initialize Card]
60000000105000200010025720010928100000000000100000000000000000000000000 [RECV: Operation Command ACK]
00000000202000100020025720010928N2001092820010928U0001000000001040000000000    [SENT: Create ICC on Call Collector]
60000000205000200010025720010928100000000000200000000000000000000000000 [RECV: Operation Command ACK]
00000000301000100020025720010928N2001092820010928U00010000000013001000000005000 [SENT: Create Credit]
60000000305000200010025720010928100000000000300000000000000000000000000 [RECV: Operation Command ACK]
```

Feedback Port

```
000000001050001000200257200109281002 [SENT: No Command]
60000000405000200010025720010928100000000000100000000000000000000000000 [RECV: Operation Command ACK]
```

To get a short help, click on one of the parameters.   Connected

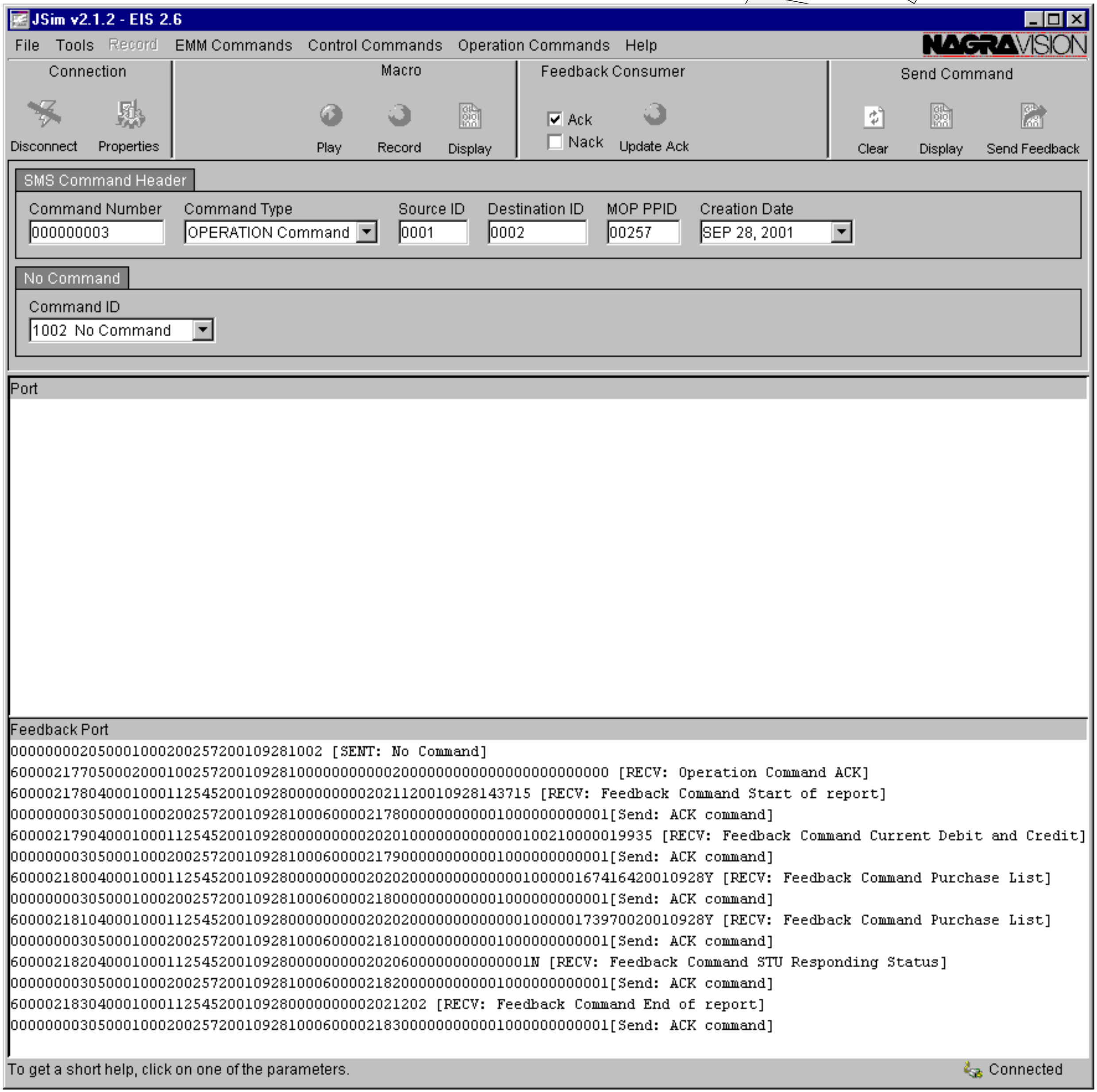
Fig. 6.4 – JSim display after SMS commands 1002 is sent

At the same time, the following ITM traces can be observed:

```
|none      MESSAGE     4 28 Sep 2001 14:28:17 SMS_SMHandler        SMSConnect      HIGH    |
Received a connection from SMS 1.
|none      MESSAGE     2 28 Sep 2001 14:28:17 SMS_SMHandler        perform_request HIGH    |
Request from SMS_SM
|none      MESSAGE     2 28 Sep 2001 14:28:17 SMSCoordinatorFactor newCoordinator  HIGH    |
Schedule submit for SMS command #1002
|EVENT: /ITM/ITM                   28 Sep 2001 14:28:17 HIGH    | ClassName:
'NoCommandSubmitCoordinator' - MethodName: 'process' - Reason: 'No Commit for this Submit:
processing finished successfully'
|none      MESSAGE     2 28 Sep 2001 14:28:17 SMS_SMHandler        perform_request HIGH    |
Submit Request succeeded
```

6.3.4. Generating an ISD callback

An ISD callback is now realized in order to launch the feedback commands generation for the calling ISD (immediate generation mode has been set for the ITM). Six feedback commands are generated by the ITM component and the following traces appear in the JSim window:



Command Number: 000000003
Command Type: OPERATION Command
Source ID: 0001
Destination ID: 0002
MOP PPID: 00257
Creation Date: SEP 28, 2001

Command ID: 1002 No Command

```
000000002050001000200257200109281002 [SENT: No Command]
6000021770500020001002572001092810000000000002000000000000000000000000 [RECV: Operation Command ACK]
60000217804000100011254520010928000000000002021120010928143715 [RECV: Feedback Command Start of report]
000000003050001000200257200109281000600002178000000000000100000000000l[Send: ACK command]
600002179040001000112545200109280000000000020201000000000000001002100000l9935 [RECV: Feedback Command Current Debit and Credit]
000000003050001000200257200109281000600002179000000000000100000000000l[Send: ACK command]
6000021800400010001125452001092800000000000202020000000000000001000001674l64200l0928Y [RECV: Feedback Command Purchase List]
000000003050001000200257200109281000600002180000000000000100000000000l[Send: ACK command]
6000021810400010001125452001092800000000000202020000000000000001000001739700200l0928Y [RECV: Feedback Command Purchase List]
000000003050001000200257200109281000600002181000000000000100000000000l[Send: ACK command]
60000218204000100011254520010928000000000002020600000000000000lN [RECV: Feedback Command STU Responding Status]
000000003050001000200257200109281000600002182000000000000100000000000l[Send: ACK command]
600002183040001000112545200109280000000000021202 [RECV: Feedback Command End of report]
000000003050001000200257200109281000600002183000000000000100000000000l[Send: ACK command]
```

To get a short help, click on one of the parameters.
Connected

Fig. 6.5 – JSim display after feedback commands are received

At the same time, many traces are generated by the ITM component:

```
|none      MESSAGE     11 28 Sep 2001 14:37:13 SSMHandler          perform_request HIGH    |
Request from SSM
|none      MESSAGE     11 28 Sep 2001 14:37:13 CoordinatorFactory  newCoordinator  HIGH    |
SSM command is a CCM
|none      MESSAGE     11 28 Sep 2001 14:37:13 SSMHandler          perform_request HIGH    |
SSM Submit Coordinator scheduled
FOUND ?
ANI PRESENT
ANI PRESENT : 0000000000000000
```

```
 Composer::EZU>xField=IPPV1
 Composer::EZU>fieldName=FID
 Composer::EZU>fieldName=Flags
 Composer::EZU>fieldName=RID
 Composer::EZU>fieldName=PDate
 Composer::EZU>fieldName=EDate
 Composer::EZU>xField=IPPV2
 Composer::EZU>fieldName=FID
 Composer::EZU>fieldName=Flags
 Composer::EZU>fieldName=RID
 Composer::EZU>fieldName=PDate
 Composer::EZU>fieldName=EDate
 Composer::EZU>xField=IPPV3
 Composer::EZU>fieldName=FID
Composer::EZU>fieldName=Flags
 Composer::EZU>fieldName=RID
 Composer::EZU>fieldName=PDate
 Composer::EZU>fieldName=EDate
*** getISD : ***
SELECT    T0.ua,    T0.managementstatus,    T0.interactivepurchase,    T0.automaticcallback,
T0.callfrequency,     T0.nextcb,     T0.lateresponding,     T0.initialsms,     T0.pairedird,
T0.cb_ipaddress, T0.auth_phone1, T0.auth_phone2, T0.auth_phone3, T0.cb_phone, T0.last_phone
FROM ISD T0 WHERE T0.UA=2
Create a new DB connection for thread : 0x14138b940
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:13  LOW       |  ClassName:
'InformationService'  -  MethodName:  'getConnection'  -  Reason:  'New  DB  connection  creation
time : 0s'
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:14  LOW       |  ClassName:
'InformationService' - MethodName: 'getISD' - Reason: 'DB Select time : 1s'
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:14  LOW       |  ClassName:
'InformationService' - MethodName: 'isAlreadyExistingIRD' - Reason: 'DB Select time : 0s'
|EVENT: /ITM/ITM                  28 Sep 2001 14:37:14 LOW    | ClassName: 'CallBackHistory'
- MethodName: 'insert' - Reason: 'DB Insert time : 0s'
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:14  LOW       |  ClassName:
'InformationService' - MethodName: 'getIRD' - Reason: 'DB Select time : 0s'
|EVENT:  /ITM/ITM                               28  Sep  2001  14:37:14  HIGH      |  ClassName:
'CCMSubmitCoordinator' - MethodName: 'process' - Reason: 'Detected an CRDReport in CCM'
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:14  LOW       |  ClassName:
'InformationService' - MethodName: 'getAccount' - Reason: 'DB Select time : 0s'
*** updateAccount : ***
update                                      Account                                        set
MonetaryUnit='N/A',CreditLimit=200,CB_Threshold=20,Credit=210,Debit=199.35,AccountType=0
where ISD_UA=0000000002 and AccountID=1
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:14  LOW       |  ClassName:
'InformationService' - MethodName: 'updateAccount' - Reason: 'DB Update time : 0s'
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:14  LOW       |  ClassName:
'CCMSubmitCoordinator' - MethodName: 'process' - Reason: 'IRDSNRecord indicated by FID'
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:14  LOW       |  ClassName:
'CCMSubmitCoordinator' - MethodName: 'process' - Reason: 'EEPROM indicated by FID'
|EVENT:  /ITM/ITM                               28  Sep  2001  14:37:14  HIGH      |  ClassName:
'CCMSubmitCoordinator' - MethodName: 'process' - Reason: 'Detected an IPPV in CCM'
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:14  LOW       |  ClassName:
'InformationService' - MethodName: 'InformationService' - Reason: 'DB Select time : 0s'
|none      WARNING     12 28 Sep 2001 14:37:14 ITMComponent         createIPPV      HIGH   |
Attempt to create a duplicate IPPV for UA=2, Product=000007238501, PurchaseDate=20010928
|EVENT:  /ITM/ITM                               28  Sep  2001  14:37:14  HIGH      |  ClassName:
'CCMSubmitCoordinator' - MethodName: 'process' - Reason: 'Detected an IPPV in CCM'
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:14  LOW       |  ClassName:
'InformationService' - MethodName: 'InformationService' - Reason: 'DB Select time : 0s'
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:14  LOW       |  ClassName:
'InformationService' - MethodName: 'createIPPV' - Reason: 'DB Insert time : 0s'
|EVENT:  /ITM/ITM                               28  Sep  2001  14:37:14  HIGH      |  ClassName:
'CCMSubmitCoordinator' - MethodName: 'process' - Reason: 'Detected an IPPV in CCM'
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:14  LOW       |  ClassName:
'InformationService' - MethodName: 'InformationService' - Reason: 'DB Select time : 0s'
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:14  LOW       |  ClassName:
'InformationService' - MethodName: 'createIPPV' - Reason: 'DB Insert time : 0s'
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:14  LOW       |  ClassName:
'InformationService' - MethodName: 'updateIRD' - Reason: 'DB Update time : 0s'
|EVENT:  /ITM/ITM                               28  Sep  2001  14:37:14  HIGH      |  ClassName:
'CCMSubmitCoordinator' - MethodName: 'process' - Reason: 'Launch commit coordinator'
|EVENT:  /ITM/ITM                               28  Sep  2001  14:37:14  HIGH      |  ClassName:
'CCMCommitCoordinator' - MethodName: 'process' - Reason: 'Started'
```

```
|none      MESSAGE     11 28 Sep 2001 14:37:14 SSMHandler           perform_request HIGH   |
SSM Submit Coordinator succeeded
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:15  LOW       |  ClassName:
'InformationService' - MethodName: 'getAccount' - Reason: 'DB Select time : 1s'
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:15  LOW       |  ClassName:
'InformationService' - MethodName: 'getAccount' - Reason: 'DB Select time : 0s'
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:15  LOW       |  ClassName:
'InformationService' - MethodName: 'getAccount' - Reason: 'DB Select time : 0s'
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:15  LOW       |  ClassName:
'InformationService' - MethodName: 'getAccount' - Reason: 'DB Select time : 0s'
*** updateISD : ***
update                                        ISD                                          set
ManagementStatus=1,InteractivePurchase=1,AutomaticCallBack=0,CallFrequency=0,NextCB=to_date(
'20010928123715',
'yyyymmddhh24miss'),LateResponding=1,InitialSMS=1,PairedIRD=00000000000000000001,CB_IPAddres
s='255.255.255.255:5002',Auth_Phone1='9999999999',Auth_Phone2='9999999997',Auth_Phone3='9999
999996',CB_Phone='8888888881',Last_Phone='0000000000000000' where UA=0000000002
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:15  LOW       |  ClassName:
'InformationService' - MethodName: 'UpdateISD' - Reason: 'DB Update time : 0s'
|EVENT:  /ITM/ITM                               28  Sep  2001  14:37:15  HIGH      |  ClassName:
'CCMCommitCoordinator' – Met
hodName: 'process' - Reason: 'Send EMMs back to SSM'
|EVENT:  /ITM/ITM                                28  Sep  2001  14:37:15  MEDIUM  |  ClassName:
'CCMCommitCoordinator' - MethodName: 'process' - Reason: 'EMM reply is sent to SSM'
*** getISD : ***
SELECT    T0.ua,    T0.managementstatus,    T0.interactivepurchase,    T0.automaticcallback,
T0.callfrequency,     T0.nextcb,     T0.lateresponding,     T0.initialsms,     T0.pairedird,
T0.cb_ipaddress, T0.auth_phone1, T0.auth_phone2, T0.auth_phone3, T0.cb_phone, T0.last_phone
FROM ISD T0 WHERE T0.UA=0000000002
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:15  LOW       |  ClassName:
'InformationService' - MethodName: 'getISD' - Reason: 'DB Select time : 0s'
|none      MESSAGE     12 28 Sep 2001 14:37:15 SMSFeedbackCommands  generateFeedbac HIGH   |
Feedback current credit & debit
|none      MESSAGE     12 28 Sep 2001 14:37:15 SMSFeedbackCommands  generateFeedbac HIGH   |
Feedback start of report


|INFO      EVENTTYPE   F TIMESTAMP            CLASS                 METHOD          LEVEL  |
MESSAGE
--------------------------------------------------------------------------------------------
-----------
|none      MESSAGE     12 28 Sep 2001 14:37:15 SMSFeedbackCommands  generateFeedbac HIGH   |
Generate Current DEBIT & CREDIT
|none      MESSAGE     11 28 Sep 2001 14:37:15 SMS_SMHandler        perform_request HIGH   |
Request from SMS_SM
|none      MESSAGE     11 28 Sep 2001 14:37:15 SMSCoordinatorFactor newCoordinator  HIGH   |
Schedule submit for SMS command #1000
|EVENT:  /ITM/ITM                                28  Sep  2001  14:37:15  MEDIUM  |  ClassName:
'AckSubmitCoordinator' - MethodName: 'process' - Reason: 'Started submit'
|EVENT:  /ITM/ITM                               28  Sep  2001  14:37:15  HIGH      |  ClassName:
'DoNothingAckCoordinator'  -  MethodName:  'ack'  -  Reason:  'An  ack  for  a  do-nothing  cmd  has
been received.'
|none      MESSAGE     11 28 Sep 2001 14:37:15 SMS_SMHandler        perform_request HIGH   |
Submit Request succeeded
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:15  LOW       |  ClassName:
'InformationService' - MethodName: 'getIRD' - Reason: 'DB Select time : 0s'
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:15  LOW       |  ClassName:
'InformationService' - MethodName: 'getAccount' - Reason: 'DB Select time : 0s'
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:15  LOW       |  ClassName:
'InformationService' - MethodName: 'getAccount' - Reason: 'DB Select time : 0s'
|none      MESSAGE     6 28 Sep 2001 14:37:15 SMS_SMHandler        perform_request HIGH   |
Request from SMS_SM
|none      MESSAGE     6 28 Sep 2001 14:37:15 SMSCoordinatorFactor newCoordinator  HIGH   |
Schedule submit for SMS command #1000
|EVENT:  /ITM/ITM                                28  Sep  2001  14:37:15  MEDIUM  |  ClassName:
'AckSubmitCoordinator' - MethodName: 'process' - Reason: 'Started submit'
|EVENT:  /ITM/ITM                               28  Sep  2001  14:37:15  HIGH      |  ClassName:
'DoNothingAckCoordinator'  -  MethodName:  'ack'  -  Reason:  'An  ack  for  a  do-nothing  cmd  has
been received.'
|none      MESSAGE     6 28 Sep 2001 14:37:15 SMS_SMHandler        perform_request HIGH   |
Submit Request succeeded
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:15  LOW       |  ClassName:
'InformationService' - MethodName: 'getIPPVsToReport' - Reason: 'DB Select time : 0s'
```

```
|none      MESSAGE     12 28 Sep 2001 14:37:15 SMSFeedbackCommands  generateFeedbac HIGH   |
Feedback IPPV
|EVENT:  /ITM/ITM                             28  Sep  2001  14:37:15  LOW       |  ClassName:
'InformationService' - MethodName: 'getIRD' - Reason: 'DB Select time : 0s'
|none      MESSAGE     12 28 Sep 2001 14:37:15 SMSFeedbackCommands  generateFeedbac HIGH   |
Feedback IPPV
|none      MESSAGE     11 28 Sep 2001 14:37:15 SMS_SMHandler        perform_request HIGH   |
Request from SMS_SM
|none      MESSAGE     11 28 Sep 2001 14:37:15 SMSCoordinatorFactor newCoordinator  HIGH   |
Schedule submit for SMS command #1000
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:15  MEDIUM  |  ClassName:
'AckSubmitCoordinator' - MethodName: 'process' - Reason: 'Started submit'
|EVENT:  /ITM/ITM                             28  Sep  2001  14:37:15  HIGH     |  ClassName:
'IPPVAckCoordinator' - MethodName: 'ack' - Reason: 'An ack for a 202 cmd has been received.'
|EVENT:  /ITM/ITM                             28  Sep  2001  14:37:15  LOW      |  ClassName:
'InformationService' - MethodName: 'getIRD' - Reason: 'DB Select time : 0s'
|EVENT:  /ITM/ITM                             28  Sep  2001  14:37:15  LOW      |  ClassName:
'InformationService' - MethodName: 'updateIPPV' - Reason: 'DB update time : 0s'
|none      MESSAGE     11 28 Sep 2001 14:37:15 SMS_SMHandler        perform_request HIGH   |
Submit Request succeeded
|none      MESSAGE     12 28 Sep 2001 14:37:15 SMSFeedbackCommands  generateFeedbac HIGH   |
Feedback STU Responding Status
|none      MESSAGE      6 28 Sep 2001 14:37:15 SMS_SMHandler        perform_request HIGH   |
Request from SMS_SM
|none      MESSAGE      6 28 Sep 2001 14:37:15 SMSCoordinatorFactor newCoordinator  HIGH   |
Schedule submit for SMS command #1000
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:15  MEDIUM  |  ClassName:
'AckSubmitCoordinator' – Met
hodName: 'process' - Reason: 'Started submit'
|EVENT:  /ITM/ITM                             28  Sep  2001  14:37:15  HIGH     |  ClassName:
'IPPVAckCoordinator' - MethodName: 'ack' - Reason: 'An ack for a 202 cmd has been received.'
|EVENT:  /ITM/ITM                             28  Sep  2001  14:37:15  LOW      |  ClassName:
'InformationService' - MethodName: 'getIRD' - Reason: 'DB Select time : 0s'
|EVENT:  /ITM/ITM                             28  Sep  2001  14:37:15  LOW      |  ClassName:
'InformationService' - MethodName: 'updateIPPV' - Reason: 'DB update time : 0s'
|none      MESSAGE      6 28 Sep 2001 14:37:15 SMS_SMHandler        perform_request HIGH   |
Submit Request succeeded
|none      MESSAGE     12 28 Sep 2001 14:37:15 SMSFeedbackCommands  generateFeedbac HIGH   |
Feedback end of report
|none      MESSAGE     11 28 Sep 2001 14:37:15 SMS_SMHandler        perform_request HIGH   |
Request from SMS_SM
|none      MESSAGE     11 28 Sep 2001 14:37:15 SMSCoordinatorFactor newCoordinator  HIGH   |
Schedule submit for SMS command #1000
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:15  MEDIUM  |  ClassName:
'AckSubmitCoordinator' - MethodName: 'process' - Reason: 'Started submit'
|EVENT:  /ITM/ITM                             28  Sep  2001  14:37:15  HIGH     |  ClassName:
'DoNothingAckCoordinator'  -  MethodName:  'ack'  -  Reason:  'An  ack  for  a  do-nothing  cmd  has
been received.'
|none      MESSAGE     11 28 Sep 2001 14:37:15 SMS_SMHandler        perform_request HIGH   |
Submit Request succeeded
|EVENT:  /ITM/ITM                             28  Sep  2001  14:37:15  HIGH     |  ClassName:
'CCMCommitCoordinator' - MethodName: 'process' - Reason: 'Feedback command sent'
|EVENT:  /ITM/ITM                             28  Sep  2001  14:37:15  HIGH     |  ClassName:
'CCMCommitCoordinator' - MethodName: 'process' - Reason: 'Commit succeeded'
|none      MESSAGE      6 28 Sep 2001 14:37:15 SMS_SMHandler        perform_request HIGH   |
Request from SMS_SM
|none      MESSAGE      6 28 Sep 2001 14:37:15 SMSCoordinatorFactor newCoordinator  HIGH   |
Schedule submit for SMS command #1000
|EVENT:  /ITM/ITM                              28  Sep  2001  14:37:15  MEDIUM  |  ClassName:
'AckSubmitCoordinator' - MethodName: 'process' - Reason: 'Started submit'
|EVENT:  /ITM/ITM                             28  Sep  2001  14:37:15  HIGH     |  ClassName:
'DoNothingAckCoordinator'  -  MethodName:  'ack'  -  Reason:  'An  ack  for  a  do-nothing  cmd  has
been received.'
|none      MESSAGE      6 28 Sep 2001 14:37:15 SMS_SMHandler        perform_request HIGH   |
Submit Request succeeded
```

7. ITM Deployment

7.1. Initial installation/upgrade

The `nagra_install` utility will be used to perform an initial installation or an upgrade of the ITM application. Please refer to [6] to have a full description of the ITM installation.

7.2. Data migration from Call Collector to ITM

Whenever the ITM must replace an existing Call Collector (CC) application, a migration phase is necessary to export the data from the legacy database to the new ITM database. Because this migration may be customer specific, a separate document will describe in details such a migration. Here are the main operations to be followed for such a migration:

- Backup the CC database and stop the CC application,
- Install the ITM application,
- Load the data from the CC backup to the ITM database,
- Start the ITM application.

Document [7] should be consulted for more information.

8. Troubleshooting

8.1. Operating system problems

Because the ITM is installed under an ASE service (either imssoft, acssoft or itmsoft), all the ITM processes will automatically be restarted using the ASE redundancy. If a disk crash occurs on the Oracle server (i.e. the machine where the database server runs), the unique way to recover the data will be to restore the database backup. See § 9 to have a description of the maintenance of the ITM application.

8.2. SMS related problems

8.2.1. The SMS connection cannot be established

The following table gives the checklist to follow when the connection between the SMS and the SMS-SM cannot be established.

Check point	Action to perform	Action to recover
Is the SMS-SM started?	Use the command `sys_smm` to see whether the component is running.	`start_smm` to start the component.
Is the communication port ready?	Use the command `netstat -a | grep <port_number>` to see whether the port is in the `LISTEN` state.	If the port is in a “TIME_WAIT” state, stop the component using `stop_smm` and wait for 60 seconds before restarting it.
Is the SMS Gateway ready?	Use the command `sysacs` to check that the process is running.	`start_smsgwy` to start the process.

Table 8-1 – Troubleshooting : SMS commection

8.2.2. Negative acknowledgements are returned to the SMS

The next table deals with the situation where the SMS command 1001 (i.e. negative acknowledgement) is systematically returned upon a business command issue.

Check point	Action to perform	Action to recover
What type of error is it?	Look at the “error code” and “error code extension” fields of the command 1001 to have a better idea of the problem (document [3] gives a whole description of the error codes and their extension).	Send the command again once the correction of every field of the command has been done.
Where does the negative acknowledgement come from?	Look at the value of the transaction_number field: • If it’s like “’0xxxxxxxx”, it means that the negative acknowledgement has been generated by the SMS Gateway. • If it’s like “5xxxxxxxx”, it means that the negative acknowledgement has been generated by the SMS-SM. • If it’s like “6xxxxxxxx”, it means that the negative acknowledgement has been generated by the ITM.	• If the error is generated by the SMS Gateway, use the command `gshow` or `cshow` to get more information about the problem. • If the error comes from the SMS-SM, it’s probably one of the command fields that is incorrect. • If the error is generated by the ITM, look at the traces (or events) to have a better idea of

Check point	Action to perform	Action to recover
		the problem.
What caused the error?	Look at the ITM traces. Depending on the displayed messages, the main problems are: • Syntax error, • Semantic errors (ex : date out of range), • ITM not running, • Oracle database error.	• First, try to restart the ITM. • If an Oracle error is always raised, check on the Oracle server that the listener component is up and running. The listener can be stopped with `lsnrctl stop` and restarted with `lsnrctl start` • Check the environment variables `TWO_TASK`, `ORACLE_USER` and `ORACLE_PASS`. • Check that the file `tnsnames.ora` in the `$TNS_ADMIN` directory contains the right section for the ITM instance.

Table 8-2 – Troubleshooting : negative acknowledgement returned to the SMS

8.2.3. No acknowledgement is returned to the SMS

When the SMS-SM does not respond to a command sent by the SMS, it is necessary to stop the SMS-SM and the ITM processes, then to restart both of them.

8.2.4. No feedback command are generated

If the SMS doesn't receive any feedback command, execute the following check list to solve the problem.

Check point	Action to perform	Action to recover
Is the SMS connected?	Use the command `netstat –a | grep <port_number>` and check that the port is in the `ESTABLISHED` state.	Restart the SMS once its configuration parameters have been verified.
Did the SMS send command 1002?	See the log files of the SMS-SM to know whether command 1002 has been sent with the right source id.	Once connected, the first thing the SMS-SM has to do is to send the command 1002 with the right source id.
Is there any impulsive purchase to be reported to the SMS?	Check the presence of records to be retrieved in the CAS database : • Log on the Oracle server as Oracle user, • Check the value of the TWO_TASK variable that must correspond to the database name (typically `itm3net.world`) • Open a SQL*Plus session (the typical command is `sqlplus itmdba/<password>`) • Issue the following SQL request: `select count(*) from ippv where hasbeenreported=0 and ua in (select ua from isd`	Issue a callback with an ISD after one or several impulsive purchases have been done.

Check point	Action to perform	Action to recover
	`where initialsms = '0123');` where `'0123'` may be replaced with the right SMS source id. The returned number must be greater than zero. • Quit the SQL*Plus session by entering `quit;`	

Table 8-3 – Troubleshooting : no feedback command generated

8.3. IRD related problems

8.3.1. The IRD cannot call back

If the SSM does not receive any IRD connection, execute the following checklist to solve the problem.

Check point	Action to perform	Action to recover
Is the IP address of the SSM well defined?	Try to telnet to the SSM port. If the connection is refused, there is a problem in the SSM parameters.	If ASE is installed, there is a 'logical' IP address overriding the 'physical' IP address of the server. This 'logical' IP address must be specified in the 'SSM_Address' parameter.
Is the SSM listening on the IRD port?	Use the command: `netstat –a | grep <port_number>` and check that the port is in the `LISTENING` state.	Restart the SSM once its configuration parameters have been checked.
Must the IRD login be activated?	Check whether the IRD configuration enables a login authentication or not.	Adjust the 'SSM_Login' parameter according the IRD setting and restart the SSM component.

Table 8-4 – Troubleshooting : the IRD cannot call back

8.3.2. The IRD connects to the SSM but there are ciphering problems

If some errors regarding the (de)ciphering of CCM and EMM appear in the SSM and Cipher log files, execute the following checklist to solve the problem.

Check point	Action to perform	Action to recover
Is the following event sent by the Cipher component: Timeout while processing an EMM?	Check the 'Cipher_EmeList' parameter in the global configuration file. Be especially cautious about the EME name, it usually is '$9879/tcp:9879' and one must make sure the $ is not interpreted by the shell as a variable. Single quotes are usually enough, sometimes you need to specify '\$9879/tcp:9879'.	Restart the Cipher once its configuration parameters have been checked.
	Check that there is a line for the Cipher in the EME's *tcpparams* file.	Restart the EME once its configuration parameters have been checked.

Table 8-5 – Troubleshooting : ciphering problems

9. Maintenance

9.1. Product support

Troubleshooting support and maintenance assistance is bound to the contractual terms and conditions agreed with Nagravision SA.

9.2. Application maintenance

Apart from the backup of the Oracle database, the ITM needs no regular maintenance to maintain its performance.

9.3. ITM database maintenance

9.3.1. Database backup

The ITM database backup is an operation that must be manually and daily executed by the operator in charge of maintaining the system. Note that this operation does not require to stop the ITM application. The following commands must be entered in sequence:

Command	Description
`rlogin –l oracle itmora`	Logs on to the oracle account on the Oracle server.
`export ORACLE_SID=<ITM_SID>`	Exports the ORACLE_SID environment variable to point to the <ITM_SID> of the ITM database.
`cd ~/upgrades/itm<ITM_SIDNb>/ v<ItmDbVersion >/bin`	Change to the directory containing the database scripts.
`./db_backup ITM`	Launch the database backup. This operation generates a file which location is given at the backup's end. Note that the only parameter of this script is the type of the database, i.e. 'ITM'.
. . .	Copy the database file on tape device. This operation is not described in the present document because it depends on the site configuration

Table 9-1 – ITM database backup

9.3.2. Database restore

Restoring the ITM database must be done in exceptional situations where a system crash has brought data into an unrecoverable state. This operation has to be manually done and consists of the following operations:

command	Description
. . .	This operation is not described in the present document because it depends on the site configuration.
`rlogin –l oracle itmora`	Logs on to the oracle account on the Oracle server.
`export ORACLE_SID=<ITM_SID>`	Exports the ORACLE_SID environment variable to point to the <ITM_SID> of the ITM database.
`cd ~/upgrades/itm<ITM_SIDNb>/`	Change to the directory containing the database scripts.

`v<ItmDbVersion >/bin`	
`> ./db_restore ITM <ItmDbVersion> N Y`	Launch the database restore. Note that the various parameters of this script are: • DBType : type of the database, for instance 'ITM', • DBVersion : database version, like '1.04' for instance, • Structure : specify whether the database structure and data dictionary must be created [Y/N], • Data: specify if the content of the database must be imported from an export file [Y/N].

Table 9-2 – ITM database restore

9.3.3. Database purge

The ITM database purge is also an operation that must be done manually and periodically by the operator in charge of maintaining the system.

Note It is the intention of Nagravision to provide a scheduling mechanism for the periodic execution of purge procedures but this is not available in the present release.

Note that this operation does not require to stop the ITM application. The following commands must be entered in sequence:

command	Description
`rlogin –l oracle itmora`	Logs on to the oracle account on the Oracle server.
`export ORACLE_SID=<ITM_SID>`	Exports the ORACLE_SID environment variable to point to the <ITM_SID> of the ITM database.
`cd ~/upgrades/itm<ITM_SIDNb>/ v<ItmDbVersion >/bin`	Change to the directory containing the database scripts.
`./ItmdbPurge`	Database purge. This operation deletes old data stored in the database.

Table 9-3 – ITM database purge

Criteria applied for the purge of ITM data are described in the following table:

Db table	Criteria	Customizable parameter	Default value
CALLBACK	DATTIME < today – N	N : history period for callbacks	N : 31 days
IPPV	(PURCHASEDATE < today – N) and (HAS_BEEN_REPORTED = 1)	N : history period for reported IPPV	N : 31 days[1]

Table 9-4 – Purge criteria for ITM data

As indicated in the previous table, several parameters can be modified to meet the specific needs of the customer. These parameters must be edited as follows:

Command	Description
`rlogin –l oracle itmora`	Logs on to the oracle account on the Oracle server.
`cd $RTM_HOME/sql`	Moves to directory where definition files of SQL packages are located.
`cd ~/upgrades/itm<ITM_SIDNb>/ v<ItmDbVersion >/bin`	Change to the directory containing the database scripts.
`vi ItmdbPurge`	Modify parameters and save the file.

Table 9-5 – Modification of the purge parmeters

[1] To be noticed that the SMS command 111 'Get History From Call Collector' allows the SMS to retrieve a two-month history of IPPV information. This is why this parameter should not be less than 60 days (therefore, the default value should be modified accordingly…)

9.4. Reporting anomalies to NagraVision

Before contacting Nagravision with the address "support@nagra.com", please write following information down:

- ITM version,
- ITM database version,
- EIS protocol version used.

Additionally, the full content of the configuration file `init_itm.dat` should be provided.

Appendix A Glossary of terms

Here is the list of technical terms used throughout this document:

Term	Definition
Circular log file	Logs are reported according a circular position in the file. Circular means that when the end of the file is reached, the current log supersedes the 1st line in the file, the next log supersedes the 2nd line, etc.
Conditional Access System (CAS)	System granting access rights to the subscribers.
Fields	In this document, the term “field” stands for the data contained in a SMS command or a CCM (Call Collector Message).
Jsim	SMS simulator written in Java
Product	Regular subscription or pay per view (or set of them) a subscriber can watch on its TV.
Session	There are two types of sessions mentioned in this document : • IRD sessions, which is the link between an IRD and the SSM. • Oracle sessions, which is the links established with an Oracle database in order to issue SQL request against it.
Subscriber	A subscriber owns an ISD and an IRD.
Subscriber Management System (SMS)	External front-end system that sends commands to the CAS system and receives feedback commands from it.

Table 9-6 – Glossary of terms

Appendix B Sequence diagrams

This appendix contains several sequence diagrams explaining the underlying mechanisms involved in the ITM application when dealing with external requests.

B.1 Basic mechanisms

Several common basic mechanisms are used in different parts of the ITM application that are described below.

B.1.1 Two phases commit transaction

The “Two phases commit transaction” mechanism is used to execute functions without any present operator employee or subscriber. The functions that meet this requirement are typically:

- Adding rights,
- Suspend STB,
- Manage credit.

The following figure shows the mechanism involved when processing a SMS command:

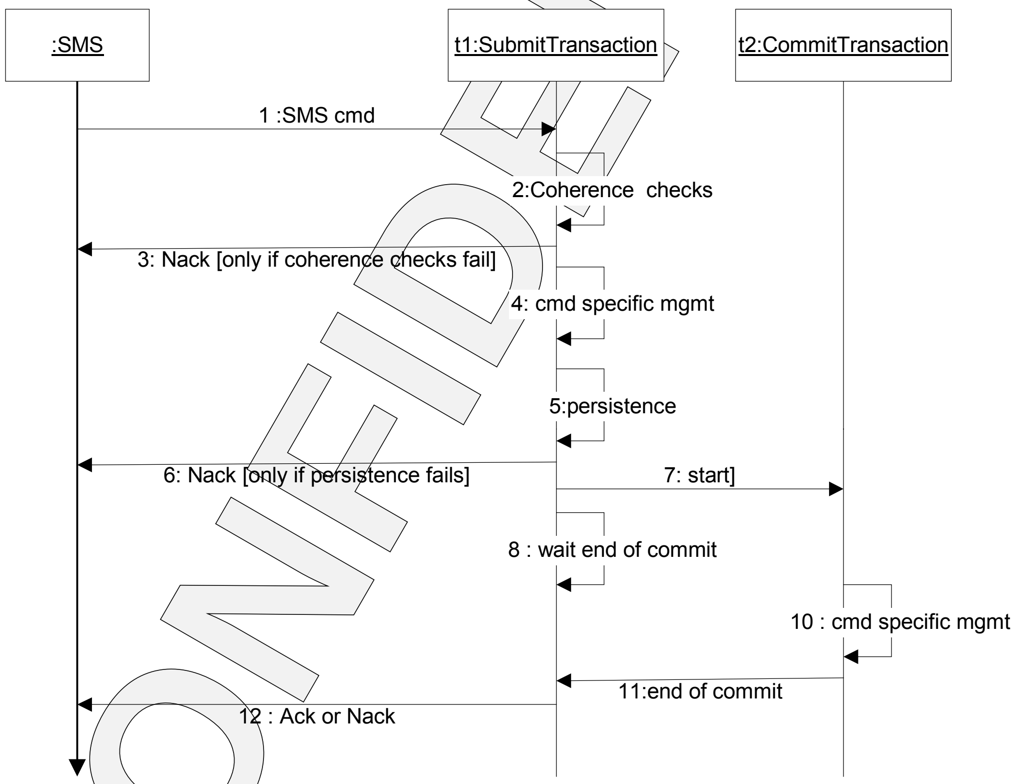


Fig. 9.1 – Two phases commit transaction mechanism

B.1.2 Request coordinator

On an incoming SMS command (as well as for an IRD callback), the client creates a request and transmits it to an interface which is in charge of scheduling the execution of the request. Once the result of the request is available, the client may have access to it by calling the GetResult method.

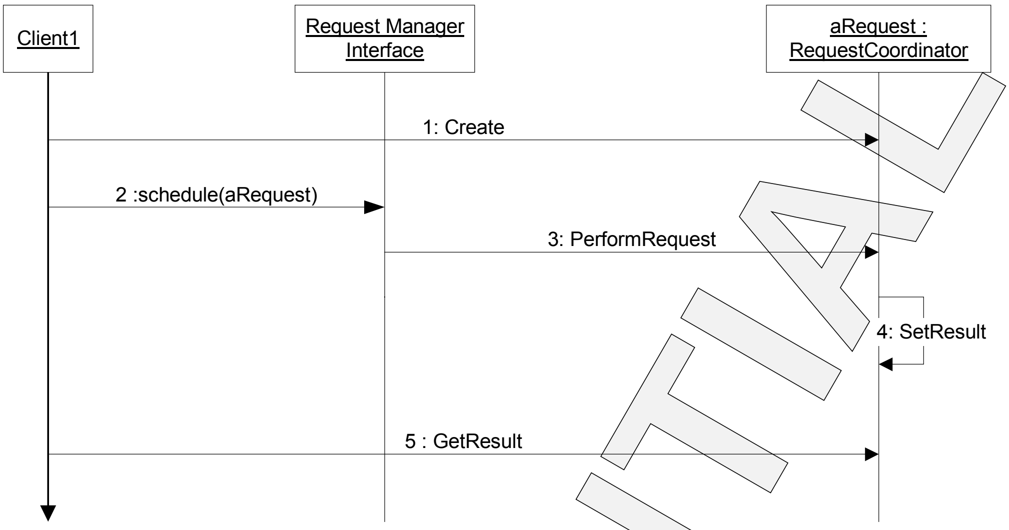


Fig. 9.2 – Request coordinator

B.1.3 Acknowledgement strategy

Two layers are used to manage acknowledgements. A session manager performs the syntax check while a request manager deals with the semantic check.

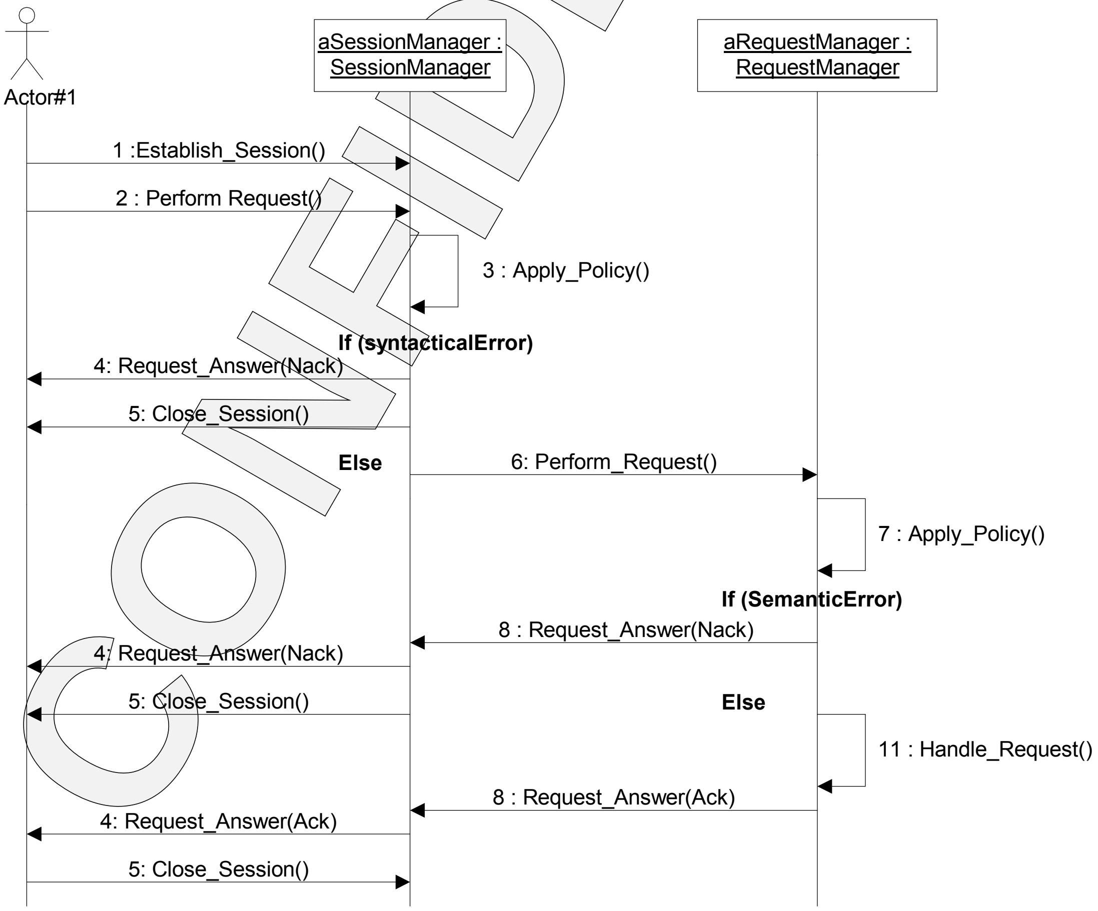


Fig. 9.3 – Acknowledgement strategy

B.2 Feedback command generation

This paragraph gives an overview of the management of feedback commands.

B.2.1 Acknowledgement manager

When a feedback command is forwarded by the ITM to the SMS, a module called ‘acknowledge manager’ registers an object called ‘acknowledge coordinator’ until the command has been acknowledged.

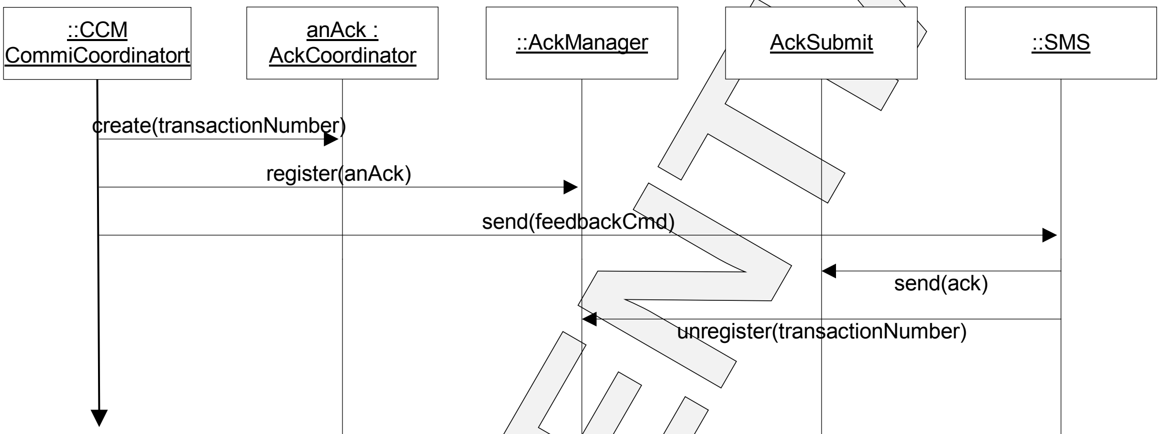


Fig. 9.4 – Acknowledgement manager

B.2.2 Immediate mode

In this mode, feedback commands are generated at the callback time. If the SMS does not acknowledge the command, the feedback commands are built during the next callback.

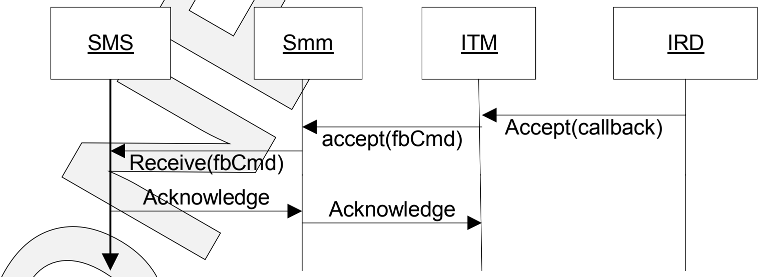


Fig. 9.5 – Immediate mode

B.2.3 Batch mode

In this mode, every 10 minutes (by default), a process wakes up and generates all the feedback commands for a subset of subscribers. On the next go, the process takes another subset and so forth.

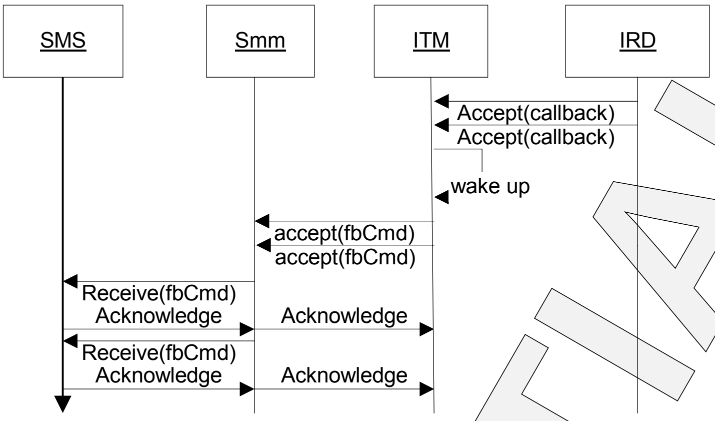


Fig. 9.6 – Batch mode

Appendix C Differences between Call Collector and ITM

This appendix contains the list of differences between the legacy 'Call Collector' application (also called CCO) and the new ITM DNASP-2 application, which aim is to replace and enhance functionality of the former one. These differences are presented is the next table:

Call Collector	ITM DNASP-2	Difference
The CCO database is 'home made' and its access is restricted to Nagravision specific tools.	The ITM DNASP-2 database is an Oracle database.	The whole set of Oracle tools can be used to access, monitor and manage the ITM database.
The CCO application is multi-threaded but monolithic.	The ITM application is multi-threaded and made of several independent components.	• The ITM application can benefit from the power of a multi-processor computer. • The ITM application is scalable and its components can be spread onto several computers.
The CCO application is not multi-threaded.	Every component of the ITM application is multi-threaded.	The ITM application can benefit from the power of a multi-processor computer.
The CCO architecture is not up-to-date.	The ITM architecture integrates several state-of-the-art technologies like CORBA, XML and the multi-threaded communication library ACE.	• The CCO can handle up to 30 simultaneous callbacks whereas the ITM can manage up to 200 simultaneous callbacks. • The ITM offers an open architecture, which enables the ability to easily add new features.
Only one instance of the CCO application can be installed in a CAS system.	Thanks to an open architecture relying on CORBA and due to the presence of the SMS-SM component, several instances of the ITM can be installed in the same CAS system.	The ITM application is scalable and can follow the customer expansion. Several ITM can be spread through different geographical areas.
The CCO can manage both a PSTN and a cable return path at the same time.	The ITM can also manage both a PSTN and a cable return path at the same time.	The ITM can manage a hybrid configuration with both kinds of return path. However, in such a configuration, the login authentication must be disabled on IRDs using the cable return path.
The CCO is not integrated with the Nagravision System Management (NSM).	The ITM is integrated with the NSM.	System management of the ITM can be done through the NSM console, which centralizes all events of every application. Monitoring the application activity is much easier.
The SMS feedback command 206 'STU Responding Status' is only generated during the IRD callback.	The ITM offers a special function enabling the periodic generation of the SMS feedback command 206 'STU Responding Status' for every IRD that is in 'late' status in the ITM database.	A new function is available in the ITM to let the operator periodically know about the IRDs being in a late responding status.
The SMS feedback commands generation is done only during the IRD callback but an external CCO tool named 'f202send' can launch a standalone generation of SMS feedback commands 202 'PPV Purchase List'.	The ITM offers a periodic generation mode where SMS feedback commands are generated every 10 minutes for a subset of ISDs (based on a modulo on the UA value).	This periodic generation mode can be used alone or it can be combined with the immediate generation mode depending on a startup parameter. To be noticed that SMS feedback commands are generated (in the periodic mode) only for ISDs owning unreported IPPV in the database.

Call Collector	ITM DNASP-2	Difference
SMS downtime is managed when only one SMS is connected.	The SMS downtime management policy can handle several SMS at the same time.	The SMS downtime management policy of the ITM observes the following rules : • There is a limit in the number (1,000 by default) of SMS feedback commands sent to the SMS and still not acknowledged by the SMS. When this limit is reached, no more SMS feedback command is generated by the ITM. • When a SMS feedback command is still not acknowledged after a given delay (default is one hour), this command is considered as negatively acknowledged by the SMS.

Table 9-7– Differences between CallCollector and ITM DNASP-2

—— END OF DOCUMENT ——