NAGRAVISION®
DIGITAL PAY-TV SYSTEMS

Company Confidential

NAGRAVISION SMS GATEWAY

Interface Specification

CAS Aladin 1.4 STD

ISSUE 1.0.0

SAS

Issued on June 30, 2008 for CNS Antigua.

CH–1033 Cheseaux, Vaud, Switzerland.

First published, April 2007.
Revised, -.
Part number: SmsGwySpe. CAS-Aladin1.4-STD.1.0.0

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

Contents–Summary

Contents–Full listing

List of figures

List of tables

Conventions used in this guide

Pull-quotes

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

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

Convention for Windows

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

Acknowledgements

Printing or viewing online

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

Also, refer to the notice at the bottom 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

Term	Definition	Description
ANI	Automatic Number Identification	Also known as caller id
CAS	Conditional Access System	A generic term for a system used in pay television.
CC	Call-Collector	This is the sub-system of the CAS managing the callback coming from the STB
DVB	Digital Video Broadcasting	DVB is a family of international standards for all program delivery media: satellite, cable, terrestrial, microwave, MDS, CATV, and SMATV.
EBNF	Extended Backus-Naur Form	A formal mathematical way to describe a language
EMM	Entitlement Management Message	Carries data from the system to one or many smart cards.
ICC	Integrated Circuit Card	Smart card
IPPV	Impulse Pay Per View	A PPV event product, which can be impulsively purchased through the STB and results in the acquisition of the related entitlement needed in the SMART CARD.
ITM	Interactive Transaction Manager	This is the new name of the call-collector
MOP	Management Operator	The operator who manages end-users and entitlements.
NVOD	Near Video On Demand.	
PPV	Pay Per View	
Product	Product	A single or group of services or events that may be purchased as a single entity.
SAS	Subscriber Authorization System	This is the sub-system of the CAS that converts the SMS command to EMM
SMS	Subscriber Management System	
SMSgw	SMS gateway	SMSgw is an application and an interface described in this document.
STB	Set-Top Box	The decoder installed at the end-user home
STU	See STB	
UA	Unique Address	This is the id or number of the smart card
UTC	Coordinated Universal Time	Formerly known as GMT (Greenwich mean time).

Table 0-1 Acronyms, abbreviations and other terms

Data formats

Format	Description	Samples	Data
hex	Raw hexadecimal value	19 or 13(hex) in 2 bytes 88564006 or 05476126(hex) in 4 bytes	0x00:0x13 0x05:0x47:0x61:0x26
HHMMSS	time hour-min-sec represented in ASCII	102500 (10h25 and 00 sec)	0x31:0x30:0x32:0x35:0x30:0x30
ip_num	human representation of an IP address. The length is fixed to 15 bytes (4 x 3-digit num value separated with dot characters).	As an example the IP address 1.112.25.2 must be formatted as: 001.112.025.002	0x30:0x30:0x31:0x2E:0x31:0x31:0x32:0x2E 0x30:0x32:0x35:0x2E:0x30:0x30:0x32
num	numerical value represented in ASCII.	206	0x32:0x30:0x36
num_x	hexadecimal numerical value represented in ASCII.	6A10F9	0x36:0x3A:0x31:0x30:0x3F:0x39
r_num	numerical value represented in ASCII. The range is restricted.	see num samples	
r_num_x	hexadecimal value represented in ASCII. The range is restricted.	see num_x samples	
r_p_num	numerical value represented in ASCII and padded with “space” char. The range is restricted.	see p_num samples	
p_num	numerical value represented in ASCII and padded with “space” char	206__ (the value is a string of 5 digit, 3 significant digit and padded with 2 space)	0x32:0x30:0x36:0x20:0x20
r_text	any text represented in ASCII characters. The range is restricted.	N as No Y as Yes	0x4E 0x59
text	any text represented in ASCII characters.	SMS_GWY	0x53:0x4D:0x53:0x5F:0x47:0x57:0x59
YYYYMMDD	date year-month-day represented in ASCII.	20030518 (18 May 2003)	0x32:0x30:0x30:0x33:0x30:0x35:0x31:0x38

Table 0-2 Data formats

Release Notes

Release	Description
1.4.0 – Issue 1.0.0 April 2007	• The list of new commands supported by Aladin 1.4 is: 1. cmd 19 (Patch Smart Card) 2. cmd 25 (Suspend all ICC Features with Delay 3. cmd 122 (Set Network) • The list of modified commands is: 1. cmd 6 – add note for multi instance right 2. cmd 20 – "Important note" cmd 15 is no longer needed. The "Important note" details that the smart card is suspended at MOP level. 3. cmd 21 – "Important Note" cmd 15 is no longer needed 4. cmd 52, 901, 902 – add new error code "DATABASE_ERROR" 5. cmd 10, 901, 902, 903 – filler of field "event_name" must be "SPACE" instead of value 0x00. See section 4.6.1. • Add new section "Command handling" (see chapter 4.7) • Add new error code – 0051 to 0061 • Add new error code extension – 0096 to 0106 • Change IMS_product_id max value = 004294967295
1.3.1 March 2003	• Command 111 is supported by release Aladin 1.3
1.3.0 Dec 2005	• The list of new commands supported by Aladin 1.3 is: 1) cmd 53 (clear PIN code) is re-introduced to be backward compatible with SMSGWY spec. 2.6.x 2) cmd 79 (Force Tune) is re-introduced to be backward compatible with SMSGWY spec. 2.6.x 3) cmd 23-Suspend all ICC Features 4) cmd 24-Reactivate all ICC Features 5) cmd 35 – Create Operator 6) cmd 36 - Cancel Operator 7) cmd 57-Bind the CableCard with the Host 8) cmd 217 – Impulse Purchase List • The list of commands modified is: 1) cmd 60 (immediate callback) - the fields CbDate and CbTime are now optional to be backward compatible with SMSGWY spec 2.6.x 2) cmd 61 (Enable automatic callback) - the field CbTime is now optional to be backward compatible with SMSGWY spec 2.6.x 3) cmd 69 (IRD generic command): add note about pairing 4) cmd 69 (IRD generic command): an example shows how to format data field

Release	Description
	5) cmd 79-Force Tune support two data formats • add error code 48, 49, 50 • add error code extension 93, 94, 95 • add note in cmd 96 (purge PPV and IPPV) • add note about pairing for cmd 69 (IRD generic command) • add note about pairing for cmd 53 (Clear PIN code) • add note about pairing for cmd 79 (Force Tune) • Root Header structure – creation date is not used as a constraint any more • Root Header structure - MOP_PPID range is 0 to 65536 • Address Header structure – add accuracy on UA range (0 to 4294967295) • Command 79-Force Tune support two data formats

1. Introduction

This document presents the interface that interconnects an SMS and the Nagravision conditional access system (CAS). It provides the connection specifications, the communication protocol and the commands utilized in the interface. This issue illustrates only the set of commands fully supported and tested by the CAS Aladin standard delivery.

1.1 Audience

This guide is directed at the following persons:

1. The customer's personnel involved in the management of the SMS – CAS interface
2. SMS vendor's personnel involved in the development of the interface between the SMS and the CAS.
3. Nagravision's personnel involved in the CAS (marketing, customer support, developers, test team)

1.2 How to use this specification

You do not need to read this specification from cover to cover as some information is reference material only. Therefore, depending on your needs chose from the following:

– Chapter 2: System overview and communication protocols

– Chapter 3: SMS-SMSgw connection

– Chapter 4: The command format

– Chapter 5: The error codes

1.3 License notice

This document lists and describes all the commands supported by the Nagravision CAS interface. However, the usage of a given command depends on the business licenses acquired by the customer (i.e. the site operator). Make sure you have the appropriate license before using a command.

1.4 Related documents

[1] "Information technology – Syntactic meta-language – Extended BNF" (ISO/IEC 14977:1996)

[2] "Conditional Access Kernel - IRD Command Specification" from Nagravision

2. System overview

This chapter presents different aspects of the system related with the SMS gateway interface.

2.1 SMSgw purpose

The figure below illustrates the location of the SMS gateway (SMSgw). The SMSgw is an internal component of the Nagravision CAS system. From the outside world, it can be seen as a gate to the CAS. Through this gate, one or several SMS entities send instructions or commands towards the CAS. On the reverse way, the CAS sends information related to IPPV to a given SMS.

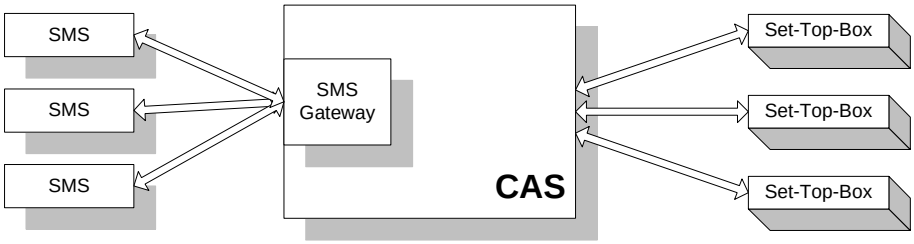


Fig. 2-1 CAS overview

2.2 Communication protocols

2.2.1 Layers

There are three communication layers. The lower level is the TCP/IP and the most abstract level is the SMSgw protocol. Between them, there is the Device IO layer whose role is to gather commands from the TCP/IP stream.

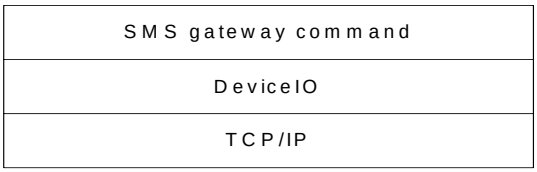


Fig. 2-2 SMSgw protocol – communication layers

2.2.2 TCP/IP protocol

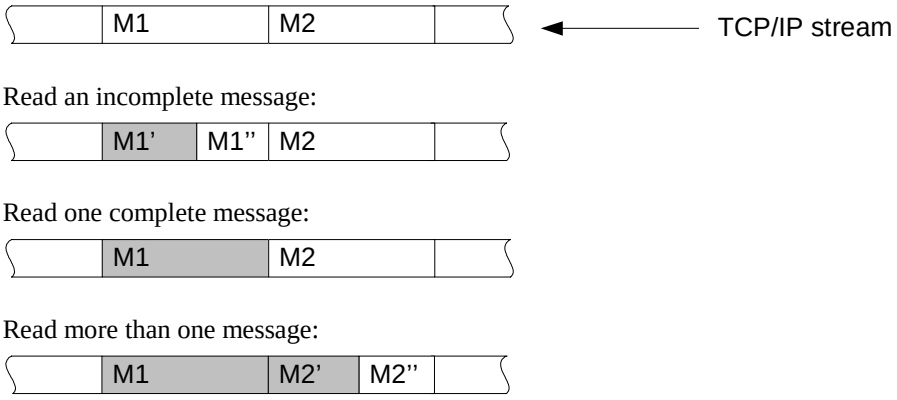
TCP/IP is a stream-based protocol. The application-oriented messages are joined together and there is no separator between these messages. When reading an application-oriented message from a stream, like TCP/IP, using OS primitives, we may have the following three possibilities:

1. The message is incomplete (only n bytes of the data have been read)

2. The message is complete (only the expected message has been read)

3. More than the expected message has been read.

The diagram below shows these three possibilities: Let's assume that the TCP/IP stream contains two application-oriented messages M1 and M2.



2.2.3 Device_IO protocol

The Device_IO is a protocol above TCP/IP used to exchange data as a buffer of known size. It consists of a fixed size header followed by the application data. The header size is two bytes, and the application data is a stream of n bytes.

Header section	Payload section
Length (2 bytes)	Application data (n bytes)

The header is a 2-byte hexadecimal value that corresponds to the number of bytes of the payload section. Warning: only the payload section length is indicated in the header. The first transmitted byte is the most significant byte of the "Length value".

The payload section content is described in the following chapters.

2.2.4 SMSgw protocol

The SMSgw protocol defines the application-oriented message. It is a high level protocol for which the user data consists of human readable ASCII characters (from ASCII code 32 to 127) used to describe the value of the command attributes.

2.3 Device_IO communication

2.3.1 Overview

Communications are established through entry points called services. An application establishes a communication with another application by specifying the service name of the target.

For Device_IO communications between applications running on different machines, a communication must first be established at the TCP/IP transport level.

To manage a communication between two systems, the following points must be addressed:

- How to establish the communication with a Device_IO server and how to indicate for which internal client the communication is intended.
- How to transmit and receive data to/from a Device_IO server.

In the following description, the commands used (open, send, receive, listen, close) are the commands of the underlying transport protocol used (TCP/IP). Device_IO does not redefine these commands. The names used below (open, send, receive, listen and close) are generic names representing the corresponding available system calls. Consequently, the calls described below only show their Device_IO parameters. Transport protocol parameters (like socket pointers for instance) are not shown in the generic description of the calls.

2.3.2 Establishing a connection with a Device_IO server

To open a communication with a Device_IO server, an application must call the Device_IO server. Two parameters must be provided:

1. The machine name on which the server is running
2. The port number (TCP-IP) corresponding to the Device_IO server.

Once the TCP communication with the server has been established, the name of the target service shall be communicated (message_1) to the Device_IO server in order to establish a link between the calling client and the requested service.

As a reply, the server shall send one or two messages. The first message (message_2) contains a connection status and, if the communication attempt has been successful, a second message (message_3) specifies whether the call has been accepted or rejected.

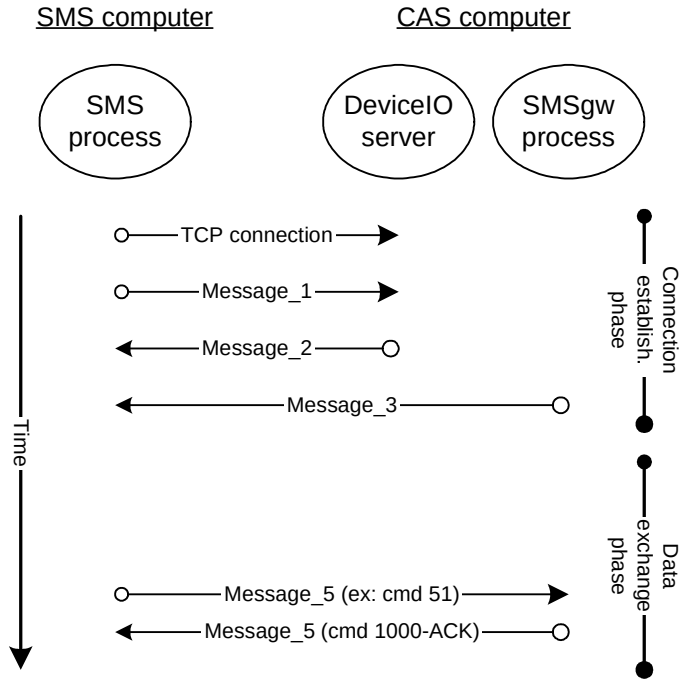


Fig. 2-3: Device_IO connection establishment and data exchange phases

An example of a Device_IO connection can be found in the chapter 6: Examples (on page 113).

2.3.3 Data exchange between client and server

After the communication link is established, messages can be exchanged (message_5) between the client (here, the SMS) and the Device_IO server.

2.3.4 Closing a connection with a Device_IO server

The connection is closed when the communication at the TCP level is closed.

2.3.5 Rules of use

If, for any reason, either the message_2 or the message_3 is not received within a specified time out (e.g. 30 sec) by the SMS when the communication is established, the SMS should close the communication and retry later. The same rule applies if the connection status returned by message_2 is different than "SUCCESS" or if the answer code returned by message_3 indicates that the call is rejected.

2.3.6 Messages format

There are four different message types in the Device_IO protocol. The next sections present the different messages and how they should be utilized.

2.3.7 Message_1 (connect to DeviceIO server)

This is the first message that is sent by the client (the SMS) to the Device_IO server.

Syntax	Size (byte)	Format	Description
len	2	hex	Length in byte of the message (see 2.2.3).
op_mode	1	r_hex	Data transfer operation mode 0 = Normal data transfer
ob_name_len	1	hex	Length of the object name attribute (in bytes).
ob_name	ob_name_len	text	Name of the applicative service to which the connection should be established. Ob_name is a string of bytes at least one byte long and at most 32 bytes long: 1 <= ob_name_len <= 32. This Name is compulsory but its content is up to the client (ex: "SMS_GWY")

2.3.8 Message_2 (answer from DeviceIO server)

This message is a response from the Device_IO server to the client (in this case the SMS); this validates the connection.

MESSAGE_2			
Syntax	**Size (byte)**	**Format**	**Description**
Len	2	hex	Message length in bytes. In this case, the length is always 1 byte.
connect_status	1	r_hex	Connection status. Refer to the table below for applicable value.

CONNECT_STATUS		
Value	**Identifier**	**Description**
0	CONNECT_FAILURE	The connection has failed for any unexpected reason
6	SUCCESS	The operation has been successfully completed.

2.3.9 Message_3 (answer from SMSgw)

This message is a response from the Device_IO server, which validates the connection with the other process (in our case: the SMSgw).

MESSAGE_3			
Syntax	**Size (byte)**	**Format**	**Description**
len	2	hex	Length in bytes of the message. In this case, the length is always 1 byte.
answer_code	1	r_num	0: call accepted 1: call rejected

2.3.10 Message_5 (message from SMS or message from SMSgw)

This message shall contain only one SMSgw command.

MESSAGE_5			
Syntax	**Size (byte)**	**Format**	**Description**
len	2	hex	Message length in bytes (see 2.2.3).
data	len	see note	SMSgw command data

The data section of the message_5 follows the structure shown below:

root header	common to all commands - see chapter 0
address header	depends on command type (EMM, CTRL, Feedback, Operation) - see chapters 4.5.2 to 4.5.5
command body	depends on command type - see chapters 4.8 to 4.11

Note The content of this section is command specific, refer to the format described in chapters 4.7 to 4.11.

3. SMS-SMSgw connections

3.1 Overview

The figure below illustrates the connections utilized by the SMS and the Nagravision SMSgw. As depicted in the figure a connection can be seen as a data channel. On the same channel, the commands are on one way and on the opposite way there are the associated responses (“ack” or “Nack”). An “ack” response means that the incoming command is correctly formatted and it has been successfully processed. A “Nack” response means that either the format or the data structure of the command is not appropriate or the command has encountered some problems during the processing in the system. The SMS is responsible for opening both channels. The commands flow from the SMS to the CAS in the channel “EMM&Control” whereas the commands flow from the CAS to the SMS in the channel “Feedback command”.

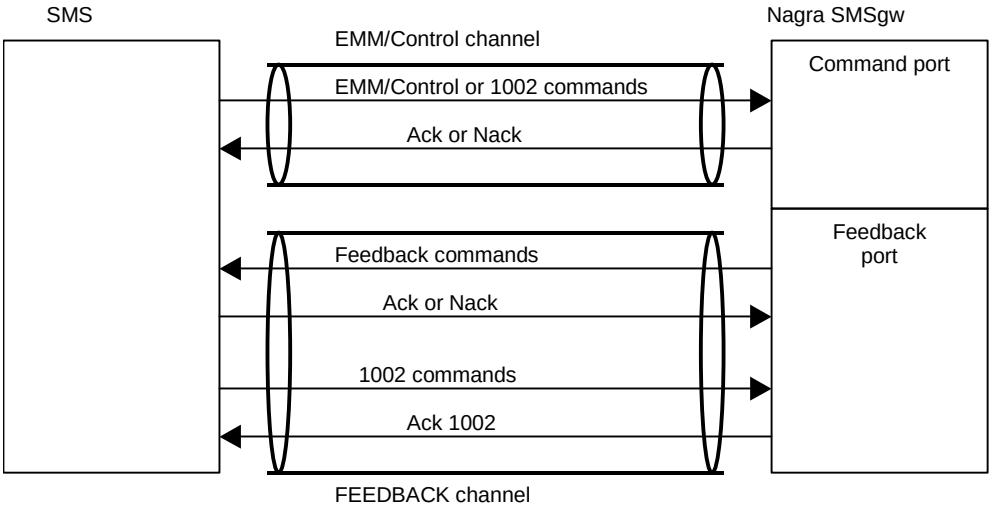


Fig. 3-1: SMS – SMSgw connections

Several SMS may be connected to the same port as shown in the following figure.

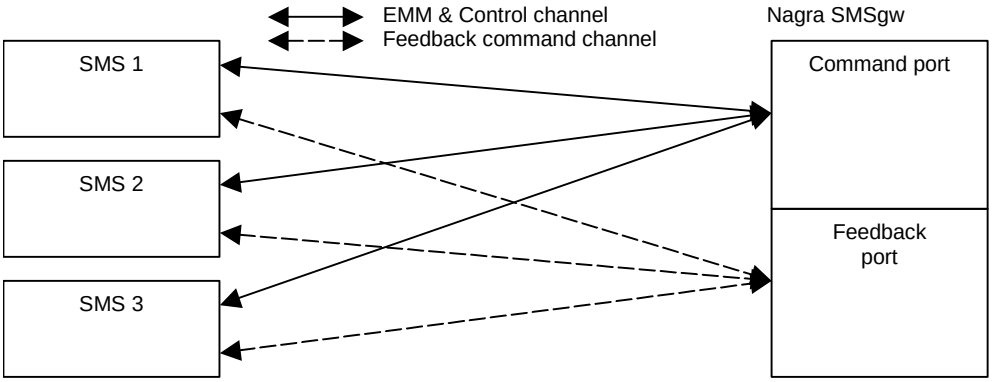


Fig. 3-2: Multi SMS connections

3.2 EMM and Control command flow

The *EMM and Control* commands are the commands sent by the SMS to the CAS. Some examples of such commands are “add product to a smart card”, “create credit”, “set authorized phone number“.

3.3 Feedback commands flow

The *feedback* commands are the commands sent by the CAS to the SMS. The kind of information issued by the CAS is one of the following:

- List of impulsive purchases that have been made by the end-users
- Error condition in the smart card (memory full, low credit)
- List of subscription products (reply of command 71: Get product)

3.4 Feedback commands routing

The routing of a *feedback* command means to which SMS the CAS should send a message. The type of message is described in the previous section.

The routing of a *feedback* command depends on the SMS source_id associated for a given SMART CARD, i.e. for a given end-user. The source_id, available in the header of any command, is extracted by the CAS during the initialization phase of an end-user. Please refer to the Important note below for more details.

Any time an SMS opens a connection on a feedback channel and after the initial Device_IO handshake has been successful, the SMS must also issue a command 1002 in order to associate an SMS_id to the connection. The CAS uses that SMS_id to enable the routing of feedback commands only if appropriate SMS is up and running.

Important

- The source_id identifies an SMS; its value shall not change over time, for the system’s lifetime. The source_id and dest_id MUST be distributed by Nagravision.
- ITM-DNASP2 system: command 52 is used to catch SMS source_id
- ITM-Aladin system: command 13 is used to catch SMS source_id

3.5 Rules of use

3.5.1 Source identifier

A source identifier is associated to each connection established between the SMS and the Nagravision CAS. Each connection must have its own source identifier and two connections may not share the same source identifier at any given time. The source identifier associated to a connection is given by the field "source_id" present in the header of each message sent from the SMS to the CAS.

3.5.2 Transaction number

Each transaction number used must be unique during the life of a given connection. The transaction number is a field located in the header of each message sent from the SMS to the CAS. This is necessary to determine which commands have been processed successfully (ACK, command 1000) or not (NACK, command 1001).

3.5.3 SMS connection is alive

During periods of SMS-CAS interface inactivity, the SMS must send periodically a command 1002 on both ports (EMM/Control and feedback) to the CAS. The main reason for such a recommendation is that idle TCP connections may not be kept open forever by intelligent network devices (such as switches or firewalls). Nagravision recommends that the command 1002 be sent every 5 minutes on both ports.

3.5.4 SMS connection establishment

Any time the SMS opens a connection on either the EMM/Control port or on the Feedback port, it must first generate a command 1002.

4. SMSgw commands

4.1 Specification

4.1.1 Command-response

The SMS sends a command to the CAS. As a response, the CAS sends an acknowledgment message back to the SMS. The same occurs for feedback commands: the CAS sends a feedback command to the SMS. As a response, the SMS sends an acknowledgement back to the CAS. The acknowledgment messages are, of course, reported on the same communication channel through which the corresponding commands have been sent.

Note

The system will not behave as expected if the following values are not handled correctly:

- Transaction_id
- Source_id
- Dest_id

4.1.2 Asynchronous by nature

The protocol between the SMS and the CAS is an asynchronous protocol. Several SMS commands may be sent before acknowledgement messages are received. In other words, there is no need to wait for the acknowledgement of a given command before sending the next commands to the CAS.

Important

The acknowledgement messages may be received (by the SMS) in a different order than the order in which the corresponding commands have been sent. However for a given SMART CARD, the sequence of EMM will follow the sequence of SMS commands.

Please note also that at the end of the transmission chain, i.e. the SMART CARD, the sequence of commands is not guaranteed.

4.1.3 Feedback

If there is no return path, the SMS does not need to open a connection on the feedback port, unless command 71 (Get Products) is sent on the control port to trigger the generations of commands 215 (Product List) on the feedback port.

4.2 Metrics

Metric	Value range (typical)
Connection	The *EMM&control* port and *feedback* ports may accept up to 10 connections each.
SMS commands	The CAS can process from 4 up to 20 SMS *EMM&control* commands per second depending on performance of the CAS hardware.
Feedback commands	The CAS can generate and send up to 500 *feedback* commands per second. The flow of data is not necessarily smooth.

Note Those metrics depend on factors such as system architecture and processing power; they are provided here to give an order of magnitude.

4.3 Time and date

All dates and times must be in UTC.

4.4 Identifiers

This section contains descriptions of all identifiers used throughout this document:

ID	Definition
circuit_id	A number uniquely identifying a video/audio/data stream.
command_id	The identifier of an SMS command, this field is part of the command. It is documented by the SMS for all commands except the FEEDBACK commands that are documented by the IMS or the CC.
dest_id	Identifier of the addressed SMS command. It is entered at the time of system configuration.
IMS_event_id	An IMS generated identifier for each event. Entered through the IMS editor or generated when processing the EPG data feed.
IMS_product_id	The identifier of a product in the IMS. It is the only product identifier known to the IMS.
MOP_PPID	The identifier of the management operator. This identifier is provided by Nagravision at system configuration time.
SMS_product_id	A product identifier for the SMS. It is provided and managed by the SMS. It is entered in the system when the SMS Gateway through commands: 300, 303, 305 or 307.
source_id	An identifier for the source of SMS commands. This identifier is entered in the system at the time of system configuration.
STB serial number	The set-top box hardware serial number.
STU_number	This Nagravision STB number is used to identify the set-top box in the CA system for pairing purposes.

Table 4-1 Identifiers

4.5 Headers

The structure of any message is as follows:

root header	common to all commands (chapter Root header described below)			
address header	EMM 4.5.2	CTRL 4.5.3	Feedback 4.5.4	Operation 4.5.5
command body	cmd 0nn 4.8	cmd 1nn 0	cmd 2nn 0	cmd 1000 4.11

The chapter below details the internal structure of the 3 modules (root header, address header and command body).

4.5.1 Root header

Field	Size	Format	Description
transaction_number	9	num	Number used to uniquely identify a transaction across the interface for each source. range: 000000000 to 999999999
command_type	2	r_num	01: EMM 02: CONTROL 04: FEEDBACK 05: OPERATION.
source_id	4	num	A number that identifies a source such as the SMS or IMS (this number is provided by Nagravision) range: 0000 to 9999
dest_id	4	num	Identifier of the addressed SMS. This number is defined at system configuration. range: 0000 to 9999
MOP_PPID	5	num	Identifier of the technical management operator. This number is provided at system configuration by: Nagravision. range: 00000 to 65535
creation_date	8	YYYYMMDD	Creation date of the command (UTC). Note: this date is not used by the CAS to apply any rule or constraint.

List of error codes (NACK messages) applicable to this part of the command are listed below:

Field	Error codes	Error codes extension
any	BAD_ROOT_HEADER_SYNTAX	NO_EXTENTED_ERROR
transaction_number	BAD_HEADER_SYNTAX	BAD_TRANSACTION_NUMBER_FORMAT
transaction_number	BAD_USAGE	TRANS_NR_ALREADY_IN_USE
command_type	BAD_HEADER_SYNTAX	BAD_COMMAND_TYPE
source_id	BAD_HEADER_SYNTAX	BAD_SOURCE_ID
source_id	BAD_USAGE	SOURCE_NOT_AUTHORIZED
source_id	BAD_USAGE	SOURCE_ALREADY_IN_USE
dest_id	BAD_HEADER_SYNTAX	BAD_DEST_ID
dest_id	BAD_USAGE	DEST_NOT_AUTHORIZED
MOP_PPID	BAD_HEADER_SYNTAX	BAD_MOP_PPID
MOP_PPID	INVALID_PPID	MOP_NOT_AUTHORIZED
creation_date	BAD_HEADER_SYNTAX	BAD_DATE_FORMAT
creation_date	BAD_USAGE	DATE_IN_THE_FUTURE

4.5.2 Address header – EMM cmd

Field	Size	Format	Description
broadcast_mode	1	r_text	N = Normal: standard broadcasting mode.
broadcast_start_date	8	YYYYMMDD	Broadcast start date (UTC). The command must be sent to the SMART CARD from this date. See Note below
broadcast_end_date	8	YYYYMMDD	Broadcast end date (UTC). The command must be sent to the smart card until this date. See Note below
address_type	1	r_text	EMM addressing mode for EMM commands U = Unique G = Global (all smart card of the MOP are addressed).
UA	10 or 0	num	UA is the Unique Address of the smart card for which the command is intended. If the "address_type" field = G, then this field is void.

Note The broadcast_start_date is always used by the SAS. If it is in the past then it is set to the current date and time, and otherwise it is used as is. The broadcast_end_date is always ignored and rather computed using the broadcast_start_date and adding the duration specified in the broadcasting profile associated to the SMS command (association between SMS command and broadcasting profile is configurable at SAS level). The recommended usage is to set the broadcast_start_date in the past or to the current date and time. This allows having a better control on the EMM bandwidth. Please do not confuse the broadcast period that specifies the period when the EMM are broadcasted, with the right validity period that specified when a right is valid.

List of error codes (NACK messages) applicable to this part of the command are listed below

Field	Error codes	Error codes extension
any	BAD_HEADER_SYNTAX	BAD_DATA_FORMAT
broadcast_mode	BAD_HEADER_SYNTAX	BAD_BROADCAST_MODE
broadcast_start_date	BAD_HEADER_SYNTAX	BAD_DATE_FORMAT
broadcast_end_date	BAD_HEADER_SYNTAX	BAD_DATE_FORMAT
broadcast_end_date	BAD_HEADER_SYNTAX	BAD_DATE_SEQUENCE
broadcast_end_date	BAD_HEADER_SYNTAX	DATE_IN_THE_PAST
address_type	BAD_HEADER_SYNTAX	BAD_ADDRESS_TYPE
address_type	BAD_USAGE	ADDRESS_TYPE_NOT_AUTHORIZED
UA	BAD_HEADER_SYNTAX	BAD_UA_FORMAT

4.5.3 Address header – Control cmd

Field	Size	Format	Description
broadcast_mode	1	r_text	N = Normal: standard broadcasting mode.
broadcast_start_date	8	YYYYMMDD	Current date (today, UTC)
broadcast_end_date	8	YYYYMMDD	Current date (today, UTC)
address_type	1	r_text	EMM addressing mode for EMM commands U = Unique
UA	10	num	UA is the Unique Address of the smart card for which the command is intended. Range is 0 to 4294967295

List of error codes (NACK messages) applicable to this part of the command are listed below

Field	Error codes	Error codes extension
any	BAD_HEADER_SYNTAX	BAD_DATA_FORMAT
broadcast_mode	BAD_HEADER_SYNTAX	BAD_BROADCAST_MODE
broadcast_start_date	BAD_HEADER_SYNTAX	BAD_DATE_FORMAT
broadcast_end_date	BAD_HEADER_SYNTAX	BAD_DATE_FORMAT
broadcast_end_date	BAD_HEADER_SYNTAX	BAD_DATE_SEQUENCE
broadcast_end_date	BAD_HEADER_SYNTAX	DATE_IN_THE_PAST
address_type	BAD_HEADER_SYNTAX	BAD_ADDRESS_TYPE
address_type	BAD_USAGE	ADDRESS_TYPE_NOT_AUTHORIZED
UA	BAD_HEADER_SYNTAX	BAD_UA_FORMAT

4.5.4 Address header – Feedback cmd

Field	Size	Format	Description
UA	10	num	UA is the Unique Address of the smart card for which the command is intended. Range is 0 to 4294967295

List of error codes (NACK messages) applicable to this part of the command are listed below

Field	Error codes	Error codes extension
any	BAD_HEADER_SYNTAX	BAD_DATA_FORMAT
UA	BAD_HEADER_SYNTAX	BAD_UA_FORMAT

4.5.5 Address header – Operation cmd

Field	Size	Format	Description
none			There is no header for this command type.

List of error codes (NACK messages) applicable to this part of the command are listed below

Field	Error codes	Error codes extension
any	BAD_HEADER_SYNTAX	BAD_DATA_FORMAT

4.5.6 List of errors due to abnormal operational conditions

Field	Error codes	Error codes extension
not applicable	BAD_ROOT_HEADER_SYNTAX	NO_EXTENTED_ERROR
not applicable	FATAL_ERROR	EXTERNAL_SYSTEM_ERROR
not applicable	NO_ITM_PRESENT	NO_EXTENDED_ERROR_CODE
not applicable	NO_RTM_PRESENT	NO_EXTENDED_ERROR_CODE
not applicable	NO_SERVER_AVAILABLE	NO_EXTENDED_ERROR_CODE
not applicable	SMS_NOT_IDENTIFIED	NO_EXTENDED_ERROR_CODE
not applicable	TM_SERVER_ERROR	CORBA_EXCEPTION
MOP_PPID	NOT_AUTHORIZED	NO_EXTENDED_ERROR_CODE
source_id	NOT_DEFAULT_FEEDBACK_SMS	NO_EXTENDED_ERROR_CODE
source_id	SOURCE_ID_ALREADY_USED	NO_EXTENDED_ERROR_CODE
source_id	SMS_NOT_AUTHORIZED	NO_EXTENDED_ERROR_CODE
source_id	NOT_AUTHORIZED	NO_EXTENDED_ERROR_CODE

4.6 Special notes

4.6.1 Event_name filler rule

The *event_name* field is characterized by its fixed size length. The useful section of the field is a string of characters (ASCII printable char). The remaining part of the field must be padded with SPACE characters (value 0x20).

As the previous version of the SMS gateway specification indicated that the padding must be done with 0x00 byte, the CAS will continue to accept incoming SMS commands that follow this format. However any new SMS development shall follow the rule of the SPACE characters.

The drawback for the CAS system when the SMS uses filler 0x00 is that the traces of the incoming SMS command saved in a log file are corrupted. This is because the value 0x00 is considered as an end of string indicator.

However, the different CAS applications processing the SMS commands are not effected by this 0x00 byte.

4.6.2 Event_name/Product_name overwriting rule

The product name or event name as it will be displayed in the STB comes from one of the following sources:

Option 1: SMS controls the product name

> Purchase list handling (SMS commands 10, 901, 902, 903, 905 and 923): If the event name length is equal to 0 then the event name used is the one provided by the IMS and stored in the CAS database.
>
> Note that for SMS command 2 and subscription product for command 901 and 902, the event name cannot be specified and is therefore always the one from IMS.

Option 2: IMS controls the product name

> Product name provided by the SMS is ignored and unconditionally replaced by value set in IMS database.

The operational mode is configurable in the CAS.

4.6.3 Price overwriting rule

The price value of a product is usually provided by the SMS (like in cmd 10 for instance). As a remainder, this price is a value that will be displayed by the STB to inform the end-user of the cost of a given product, i.e. it is for information purpose only. However, if the SMS cannot provide this price value, the CAS can replace it with a value set in the IMS database. The operational mode is configurable in the CAS.

Note: this is not applicable for subscription products.

4.7 Command handling

4.7.1 Products

The table below shows the different kinds of products and which commands handle them.

Cmd	Label	SUB	PPV		Rental DVR	Rental PPV	PPFP	Coins	TVODSub
		B	A	C	A	B	B	C	B
002	Add Product	X		X				X	X
004	Product Suspension	X	X						
005	Product Reactivation	X	X						
006	Product Cancellation	X	X	X	X	X	X		X
010	Add Event Product		X	X	X	X	X	X	
901	Activate Card	P	V	V	V	V	V		P
902	Activate Card Without Return Path	P	V	V	V	V	V		P
903	Add List of Products	P	V	V	V	V	V		P

P ➔ applicable for loop product V ➔ applicable for loop ppv

A ➔ Aladin C ➔ Cameleon B ➔ Both Aladin and Cameleon ICC

4.7.2 Credit and price

In the HE and smart cards, the credit and price are handled in CAS currency and not in local currency. The SMS must convert the credit / price data from local currency into the CAS currency. Please note that generally the coefficient is equal to one, and thus the CAS currency and local currency are the same.

CAS currency = local currency / coefficient

This applies to all types of commands (EMM, CONTROL, MACRO EMM and FEEDBACK).

4.8 EMM commands (0nn)

In this section, we present the structure of all commands related to the alteration of the smart card. The SMS will generate these commands. For each command, there are two tables. The first table illustrates the data structure of the command. The second table provides the different types of error message that could be returned by the CAS.

4.8.1 Command 2: Add Product

SAS replaces original mop by sop	no

The SMS uses this command to add, in a smart card, an entitlement of services associated to a product.

Important Command 2 shall not be used to authorize a PPV. The command 10 should be used instead.

COMMAND 2: ADD PRODUCT			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0002
IMS_product_id	12	num	IMS product ID range: 000000000000 to 004294967295
begin_date	8	YYYYMMDD	Subscription begin date (UTC). Subscription is not valid before this date.
end_date	8	YYYYMMDD	Subscription end date (UTC). Subscription is not valid after this date.

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
IMS_product_id	BAD_COMMAND_SYNTAX	BAD_IMS_PRODUCT_ID_FORMAT
IMS_product_id	PRODUCT_NOT_FOUND	NO_EXTENDED_ERROR_CODE
IMS_product_id	BAD_PRODUCT_TYPE	PPV_PRODUCT
IMS_product_id	BAD_PRODUCT_STATUS	CANCELLED_PRODUCT
IMS_product_id	BAD_PRODUCT_STATUS	SUSPENDED_PRODUCT
IMS_product_id	BAD_PRODUCT_STATUS	INVALID_PURCHASE_DATE
IMS_product_id	BAD_PRODUCT_TYPE	DRAFT_PRODUCT
begin_date	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT
begin_date	BAD_COMMAND_SYNTAX	BAD_DATE_SEQUENCE
end_date	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT
end_date	BAD_COMMAND_SYNTAX	BAD_DATE_SEQUENCE
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
MOP_PPID	INVALID_PPID	ISD_MOP_NOT_FOUND
MOP_PPID	DATABASE_ERROR	DATA_ERROR

4.8.2 Command 4: Product Suspension

SAS replaces original mop by sop	no

The SMS uses this command to suspend, in a smart card, an entitlement of services or PPV associated to a product.

The end-user will not be able to watch the corresponding services or PPV until a *product reactivation* command is sent by the SMS. This command may be used when there is a payment problem with the end-user. This command does not impact callbacks.

COMMAND 4: PRODUCT SUSPENSION			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0004
IMS_product_id	12	num	IMS product ID range: 000000000000 to 004294967295

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
IMS_product_id	BAD_COMMAND_SYNTAX	BAD_IMS_PRODUCT_ID_FORMAT
IMS_product_id	PRODUCT_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE

4.8.3 Command 5: Product Reactivation

SAS replaces original mop by sop	no

The SMS uses this command to reactivate, in a smart card, an entitlement of services or PPV that have been previously suspended (with a command 4*: Product Suspension)*. The end-user will be able to watch the corresponding services or PPV again.

COMMAND 5: PRODUCT REACTIVATION			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0005
IMS_product_id	12	num	IMS Product ID range: 000000000000 to 004294967295

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
IMS_product_id	BAD_COMMAND_SYNTAX	BAD_IMS_PRODUCT_ID_FORMAT
IMS_product_id	PRODUCT_NOT_FOUND	NO_EXTENDED_ERROR_CODE
IMS_product_id	RIGHT_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE

4.8.4 Command 6: Product Cancellation

SAS replaces original mop by sop	no

The SMS uses this command to remove, from a smart card, an entitlement of services or PPV associated to a product.
This command is applicable to PPV entitlements that have been purchased through the SMS only. In other words, not all IPPV entitlements can be cancelled with this command.

The cancellation of PPV entitlements has no impact on the end-user's credit in the smart card.

Note When several instances of a given product are created on the smart cards, this command cancels all the instances of this product.

COMMAND 6: PRODUCT CANCELLATION			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0006
IMS_product_id	12	num	IMS Product ID range: 000000000000 to 004294967295

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
IMS_product_id	BAD_COMMAND_SYNTAX	BAD_IMS_PRODUCT_ID_FORMAT
IMS_product_id	PRODUCT_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE

4.8.5 Command 7: All products cancellation

SAS replaces original mop by sop	no

The SMS uses this command to remove, from a smart card, all types of entitlements (services and PPV). This command will also remove all non-watched IPPV. However, it does not affect non-call collected IPPV that have been watched. Such IPPV will be call-collected when the next callback occurs.

Important

This command will also **suspend the IPPV purchases** (equivalent to command 14, "Suspend Impulse Purchase")

COMMAND 7: ALL PRODUCTS CANCELLATION			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0007

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
address_type	BAD_USAGE	ADDRESS_TYPE_NOT_AUTHORIZED
MOP_PPID	INVALID_PPID	ISD_MOP_NOT_FOUND

4.8.6 Command 8: Credit management

SAS replaces original mop by sop	no

The SMS uses this command to modify, in a smart card, the data of an existing credit record. The credit record in a smart card is mandatory to allow end-users purchasing impulse-pay-per-view (IPPV).

Important

- The credit information in a smart card can be updated every 4 seconds.
- Only the mode SET CREDIT is authorized.
- The initial credit value in the smart card is set with the *command 13: Create credit for impulse purchase.*

The *credit* data field in the command shall not exceed the maximum value of 65535.99. This is taken care of by the Call-Collector.

COMMAND 8: CREDIT MANAGEMENT			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0008
credit_mode	2	r_num	03 = SET CREDIT. Set the new credit value.
credit	7	r_num	Credit amount (in the local currency) representing the Range: 00000.00 to 65535.99

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
credit_mode	BAD_COMMAND_SYNTAX	BAD_CREDIT_MODE
credit	BAD_COMMAND_SYNTAX	BAD_CREDIT_FORMAT
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
MOP_PPID	ACCOUNT_NOT_FOUND	NO_EXTENDED_ERROR_CODE

4.8.7 Command 9: Update Credit Threshold

SAS replaces original mop by sop	no

The SMS uses this command to update, in a smart card, the impulse *credit threshold* value of the end-user smart card. The *credit threshold* value allows the smart card to trigger callbacks depending on the credit status.

This command **should not be used** to initialize the Credit Threshold value. The initialization is done implicitly with *command 13: Create Credit For Impulse Purchase*.

If the *credit threshold* value is set to 0 (zero), then no callback will be issued by the smart card.

COMMAND 9: UPDATE CREDIT THRESHOLD			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0009
threshold_credit	7	r_num	Lower limit under which the smart card must do a low credit call back representing the range 00000.00 to 65535.99

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
threshold_credit	BAD_COMMAND_SYNTAX	BAD_THRESHOLD_CREDIT_FORMAT
threshold_credit	BAD_COMMAND_SYNTAX	CREDIT_THRESHOLD_TOO_HIGH
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
MOP_PPID	ACCOUNT_NOT_FOUND	NO_EXTENDED_ERROR_CODE

4.8.8 Command 10: Add Event Product

SAS replaces original mop by sop	no

The SMS uses this command to add, in a smart card, an entitlement to an *event product*. An *event product* is typically a movie showing on a given channel at a given date/time. This command is typically used to add PPV entitlements for end-users who cannot impulsively purchase PPV products.

When an event product is purchased through the SMS (by calling customer service), the smart card credit and debit values are not impacted.

Important

- The event product is pre-flagged as "call collected" in the smart card. Consequently, the purge mechanism will delete an event product once the purge date condition matches the event product date.
- For systems without return path, the deletion of PPV entitlements recorded in the end-user's smart cards should be done either by background EMM cleanup command or by SMS *command 96: Purge PPV and IPPV Records*. For systems with return path, the deletion of PPV entitlements should be done either by background EMM cleanup command or by CAS CCM command generated during callbacks.
- The data in the *event_name* field of the command are not altered by the CAS. The data is transmitted as it is in the smart card. Then the STB will display the data as it does with the EIT (Event Information Table) data. Consequently, the SMS must format this string in accordance with the specification of the char set used by the STB.
- The *event_name* string must also include control character. This means that the number of displayable characters is reduced consequently.

COMMAND 10: ADD EVENT PRODUCT

Field	Size	Format	Description
command_id	4	r_num	command_id = 0010
IMS_product_id	12	num	IMS Product ID range: 000000000000 to 004294967295
length_event_name	2	r_num	Length of valid data in *event_name* field. The number of bytes should not exceed 30. This is due to the smart card storage limitation.
event_name	32	text	Event name as displayed in the PPV purchase history in the STB user interface. The number of characters must match the length *length_event_name.* . The remaining bytes should be filled-up with "SPACE" characters (value = 0x20). See special note in section 4.6.1.
price	5	num	Price of the product, representing 000.00 to 999.99

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
IMS_product_id	BAD_COMMAND_SYNTAX	BAD_IMS_PRODUCT_ID_FORMAT
IMS_product_id	PRODUCT_NOT_FOUND	NO_EXTENDED_ERROR_CODE
IMS_product_id	PPV_IN_THE_PAST	NO_EXTENDED_ERROR_CODE
IMS_product_id	BAD_PRODUCT_TYPE	REGULAR_PRODUCT
IMS_product_id	BAD_PRODUCT_STATUS	CANCELLED_PRODUCT
IMS_product_id	BAD_PRODUCT_STATUS	SUSPENDED_PRODUCT
IMS_product_id	BAD_PRODUCT_STATUS	INVALID_PURCHASE_DATE
IMS_product_id	BAD_PRODUCT_TYPE	DRAFT_PRODUCT
length_event_name	BAD_COMMAND_SYNTAX	LENGTH_TOO_LONG
length_event_name	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
price	BAD_COMMAND_SYNTAX	BAD_PRICE_FORMAT
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
MOP_PPID	INVALID_PPID	ISD_MOP_NOT_FOUND
MOP_PPID	DATABASE_ERROR	DATA_ERROR

4.8.9 Command 13: Create Credit for Impulse Purchase

SAS replaces original mop by sop	no

The SMS uses this command to add a credit data structure in a smart card. This will allow the end-user to perform impulse PPV purchases.

The SMS should send this command only once at the initialization phase of the impulsive purchase capability of the smart card. Any other credit management should be done with command 8: *credit management*.

Important

- There is no command to remove the credit data structure in a given smart card.
- If threshold_credit is equal to 0 (zero), then no callback will be issued by the smart card.

COMMAND 13: CREATE CREDIT FOR IMPULSE PURCHASE			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0013
credit	7	r_num	Credit amount set in the smart card. Warning: the range is 00000.00 to 65535.99.
threshold_credit	7	r_num	Lower limit under which the smart card must initiate a low credit call back. It represents 00000.00 to 65535.99. Note: The smart card will truncate this value, i.e. it will only consider the integer part.

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
credit	BAD_COMMAND_SYNTAX	BAD_CREDIT_FORMAT
threshold_credit	BAD_COMMAND_SYNTAX	BAD_THRESHOLD_CREDIT_FORMAT
threshold_credit	BAD_COMMAND_SYNTAX	CREDIT_THRESHOLD_TOO_HIGH
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
MOP_PPID	INVALID_PPID	ISD_MOP_NOT_FOUND
MOP_PPID	DATABASE_ERROR	DATA_ERROR

4.8.10 Command 14: Suspend impulse purchase

SAS replaces original mop by sop	no

The SMS uses this command to suspend, in a smart card, the ability of an end-user to perform impulse purchases.

The reactivation of impulse purchases may be completed using *command 15: Reactivate impulse purchase*. This command has no impact on callback operations.

COMMAND 14: SUSPEND IMPULSE PURCHASE			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0014

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE

4.8.11 Command 15: Reactivate impulse purchase

SAS replaces original mop by sop	no

The SMS uses this command to reactivate, in a smart card, the impulse purchase capability of an end-user. This command is used to activate the impulse purchase capability of an end-user smart card that has previously received a command 14: *Suspend impulse purchase* or a command 20: *Suspend end-user smart card*.

COMMAND 15: REACTIVATE IMPULSE PURCHASE			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0015

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
MOP_PPID	ACCOUNT_NOT_FOUND	NO_EXTENDED_ERROR_CODE

4.8.12 Command 19: Patch Smart card

SAS replaces original mop by sop	yes

The SMS uses this command to generate and broadcast the patch EMM currently applicable for a smart card.

Important The patch EMM currently applicable is configured at Head End level.

COMMAND 19: PATCH SMART CARD			
Field	**size**	**Format**	**Description**
command_id	4	r_num	command_id = 0019

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
UA	EXPIRED_CARD	NO_EXTENDED_ERROR_CODE

4.8.13 Command 20: Suspend Subscriber ICC

SAS replaces original mop by sop	no

The SMS uses this command to suspend, in a smart card, all entitlements of services and PPV as well as the impulse purchase.

The entitlements of impulsive PPV already purchased are also suspended.

Important This command suspends the MOP. It suspends all the entitlements for the subscriber, including those purchased impulsively, and suspends the impulsive capability. The MOP value is provided in the root header (MOP_PPID).

The SMS can still continue to send EMM command to a deactivated smart card (no error returned). This command does not impact callback operations.

COMMAND 20: SUSPEND SUBSCRIBER ICC			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0020

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
MOP_PPID	INVALID_PPID	ISD_MOP_NOT_FOUND

4.8.14 Command 21: Reactivate Subscriber ICC

SAS replaces original mop by sop	no

The SMS uses this command to reactivate all entitlements of services and PPV in a smart card. The impulsive capability is also reactivated.

Important This command reactivates the MOP section of the smart card. The MOP value is provided in the root header (MOP_PPID).

COMMAND 21: REACTIVATE SUBSCRIBER ICC			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0021

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
MOP_PPID	INVALID_PPID	ISD_MOP_NOT_FOUND

4.8.15 Command 23: Suspend all ICC features

SAS replaces original mop by sop	yes

The SMS uses this command to suspend all features in a smartcard, i.e. all entitlements of services and PPV, the impulse purchase capability as well as entitlements of the "*free acces*s" services.

The entitlements of impulsive PPV already purchased are also suspended.

Important In order to reactivate smartcard features, the SMS should use the *command 24.*

The SMS can still continue to send EMM command to a deactivated smartcard (no error returned). This command does not impact callback operations.

COMMAND 23: SUSPEND ALL ICC FEATURES			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 23

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
UA	EXPIRED_CARD	NO_EXTENDED_ERROR_CODE

4.8.16 Command 24: Reactivate all ICC features

SAS replaces original mop by sop	yes

The SMS uses this command to reactivate all entitlements of services and PPV in a smartcard as well as entitlements of the "*free acces*s" services.

COMMAND 24: REACTIVATE ALL ICC FEATURES			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 24

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
UA	EXPIRED_CARD	NO_EXTENDED_ERROR_CODE

4.8.17 Command 25: Suspend all ICC Features with Delay

SAS replaces original mop by sop	yes

The SMS uses this command to schedule a smart card suspension. This command suspends, at the date and time specified, all features in a smart card, i.e. all entitlements of services and PPV, the impulse purchase capability as well as entitlements of the "*free acces*s" services.

The entitlements of impulsive PPV already purchased are also suspended.

Important In order to reactivate smart card features, the SMS should use the *command 24.*

The SMS can still continue to send EMM command to a deactivated smart card (no error returned). This command does not impact callback operations.

COMMAND 25: SUSPEND ALL ICC FEATURES WITH DELAY			
Field	**size**	**Format**	**Description**
command_id	4	r_num	command_id = 0025
suspension_date	8	YYYYMMDD	Suspension date (generally in UTC).
suspension_time	6	HHMMSS	Suspension time (generally in UTC). For Aladin smart cards the time is ignored and set to 23:59:56

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
UA	EXPIRED_CARD	NO_EXTENDED_ERROR_CODE
suspension_date	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT
suspension_date	BAD_COMMAND_SYNTAX	BAD_DATE_SEQUENCE
suspension_time	BAD_COMMAND_SYNTAX	BAD_TIME_FORMAT

4.8.18 Command 35: Create Operator

SAS replaces original mop by sop	yes

The SMS uses this command to add a new *Management Operator* (MOP) zone in a smartcard.

This command is used ONLY if there are several operators sharing smartcards to do Pay-Tv business independently.

This command must precede any other command (like *cmd 2: add product, cmd 13: create credit*) for a given operator.

The SMS can send this command several times without damaging the entitlements already stored in the smartcard.

Important

- In a single MOP environment, this command should not be used because any new smartcard already contains the pre-defined MOP zone for the unique operator.
- The purpose of this command is not to "virgin" a smartcard, i.e. this command does not perform any cleanup in the smartcard.

COMMAND 35: CREATE OPERATOR			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0035
MOP_PPID	5	num	Identifier of the management operator to be created. range: 00000 to 99999

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
MOP_PPID	DATABASE_ERROR	DATA_ERROR
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE

4.8.19 Command 36: Cancel Operator

SAS replaces original mop by sop	yes

The SMS uses this command to remove, from a smartcard, a Management Operator (MOP) zone and all its associated data.

Important This command will indirectly remove all entitlements and credit record currently stored in the smartcard for a given operator.

COMMAND 36: CANCEL OPERATOR			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0036
MOP_PPID	5	num	Identifier of the management operator to be created. range: 00000 to 99999

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
MOP_PPID	DATABASE_ERROR	DATA_ERROR
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE

4.8.20 Command 48: Set Zip code

SAS replaces original mop by sop	yes

The SMS uses this command to set or update the Zip code of the smart card. The Zip code information is primarily used for blackout and time zone management.

Important

For countries that use a different zip code scheme (i.e. more digits or alpha/num zip code), the operator should create a lookup map table that translates local zip code format to this zip code format.

COMMAND 48: SET ZIP CODE			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0048
zip_code	5	num	end-user's zip code. Range: 00000…99999

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
zip_code	BAD_COMMAND_SYNTAX	BAD_ZIP_CODE_FORMAT
zip_code	ZIP_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE

4.8.21 Command 49: Set Callback phone number

SAS replaces original mop by sop	yes

The SMS uses this command to set or update the parameter "phone-number" in the smart card. This parameter allows the smart card/STB to open a connection with the Call-Collector.

The parameter can be a phone number or a MAC address. The STB will determine how to interpret this number. Here are two examples:

- Phone number 1-800-555-1212, cc_number_1 is 18005551212 (with 5 trailing spaces).
- MAC address 00-06-5B-BC-8F-92, cc_number_1 is in decimal as 27308887954 (with 5 trailing spaces).

To reset the phone number in the smart card, the two following ways are accepted:

- cc_number_1 is 16 space characters (ASCII 0x20)
- cc_number_1 is 16 F characters (ASCII 0x46)

Important

- No character (ex: "-" or "/") must be present in the field cc_number_1
- To set an IP address in the smart card, the SMS must send a command 54 (Set Callback IP address).

COMMAND 49: SET CALLBACK PHONE NUMBER			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0049
cc_number_1	16	p_num	Call-Collector phone number. This field must be right padded with trailing ASCII space characters.

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
cc_number_1	BAD_COMMAND_SYNTAX	BAD_PHONE_NUMBER_FORMAT
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
MOP_PPID	CB_PROFILE_NOT_FOUND	NO_EXTENDED_ERROR_CODE

4.8.22 Command 50: Cancel ICC

SAS replaces original mop by sop	yes

The SMS uses this command to permanently disable a smart card. Such command is performed when a smart card is removed from operation (for instance the smart card is reported to be stolen, lost, or damaged).

The SMS is not allowed to use a “cancelled” smart card again because the smart card *is not recoverable* after such operation.

The smart card is flagged as cancelled and deactivated in the CAS database. Consequently, any incoming commands addressed to this smart card will be non-acknowleded by the CAS system.

Important

The effect of this command in the smart card is NOT reversible. In other words, the smart card will be permanently out-of-order.

COMMAND 50: CANCEL ICC			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0050

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
address_type	BAD_USAGE	ADDRESS_TYPE_NOT_AUTHORIZED

4.8.23 Command 51: Initialize Smart card

SAS replaces original mop by sop	yes

The SMS uses this command to initialize the Management Operator (MOP) zone of a new smart card.

The SMS must issue this command prior to any other command. The CAS will reject any other command while this *command 51* is not successfully processed.

The initialization of a smart card can occur several times and has no effect on the entitlements stored in the smart card.

Important

- This command must be the first command issued by the SMS for a new smart card
- The purpose of this command is not to "reset" the smart card content, i.e. this command does not perform any cleanup of the smart card.
- This command does not perform any refurbishing process of a smart card.

COMMAND 51: INITIALIZE SMART CARD			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0051

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
UA	UA_OUT_OF_RANGE	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
MOP_PPID	CB_PROFILE_NOT_FOUND	NO_EXTENDED_ERROR_CODE

4.8.24 Command 52: Pair the ICC with the STB

SAS replaces original mop by sop	yes

The SMS uses this command to pair a smart card with the STB. This command is mandatory before any services may be authorized. The smart card must have previously been initialized with *command 51* before pairing can take place.

This command may be issued several times to pair a smart card with a different STB.

Important

- The STU_number is a 14 digits string representing a decimal value
- The STU_number can be one of the following formats:
  a) 10 digit string followed by four space characters in the range
  "0000000000 " to "4294967295 "
  b) 14 digit string in the range 00000000000000 to 00004294967295
- The "un-pairing" action is performed with the same command 52, but with STU_number "0000000000 " or 00000000000000
- The STB_number provided in the command by the SMS must not include any checksum.

COMMAND 52: PAIR THE ICC WITH THE STB			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0052
STU_number	14	r_num	CA STB serial number in decimal. range: see "important" note above

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
STU_number	BAD_COMMAND_SYNTAX	BAD_STU_NUMBER_FORMAT
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
None	DATABASE_ERROR	BAD_PARAM_IN_CASDB

4.8.25 Command 53: Clear PIN Code

SAS replaces original mop by sop	no

The SMS uses this command to reset to the default factory value the PIN code of the STB. In other words this means that the new PIN code value forced by this command is not always 0000. Note also that the functionalities or the menu protected by the PIN code depends on the STB design.

Important

- Unlike the other commands described in this document, the cmd 53 is sensitive to the pairing status of the smartcard and the STB. Thus, the result of a cmd 53 sent to a given STB will work ONLY if the STB is paired with its smartcard. The pairing function is achieved with cmd 52.

COMMAND 53: CLEAR PIN CODE			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0053

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE

4.8.26 Command 54: Set Callback IP address

SAS replaces original mop by sop	no

The SMS uses this command to set or update the smart card field containing the IP-address of the Call-Collector.

COMMAND 54: SET CALLBACK IP ADDRESS			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0054
CC_IP_address	15	ip_num	Call-Collector IP address. Field format is 000.000.000.000 to 255.255.255.255.
CC_IP_port	5	num	Call-Collector TCP/IP port.

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
CC_IP_address	BAD_COMMAND_SYNTAX	BAD_IP_ADDRESS_FORMAT
CC_IP_port	BAD_COMMAND_SYNTAX	BAD_CC_PORT_FORMAT
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
MOP_PPID	CB_PROFILE_NOT_FOUND	NO_EXTENDED_ERROR_CODE

4.8.27 Command 56: Set PIN code

SAS replaces original mop by sop	yes

The SMS uses this command to set or update one of the 16 PIN code values that can be stored in the smart card.

The PIN code is a 4 digits numeric value. The CAS performs no checks other than boundaries on the PIN code, i.e. the same PIN code can be updated many times. The CAS does not hold the PIN code in its database.

Important

- The usage of this command is customer specific. Thus, a document illustrating how this feature works end-to-end should be prepared.
- This command is not available for any type of smart card.

COMMAND 56: SET PIN CODE			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0056
Index	2	r_num	PIN code number. Range: 01-16
PIN	4	num	PIN code value

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
Index	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
Index	VALUE_OUT_OF_RANGE	NO_EXTENDED_ERROR_CODE
PIN	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE

4.8.28 Command 57: Bind the CableCard with the Host

SAS replaces original mop by sop	yes

The SMS uses this command to bind a CableCard (also known as a POD) with the Host (i.e. the STB). This command is mandatory before any services may be authorized. The smartcard must have been previously paired with *command 52* before binding can take place.

This command may be issued several times to bind a CableCard with a different Host (i.e. the last binding replaces the previous one).

This command may be used to unbind a CableCard by providing the field 'host_id' with a null value.

Important In prevision of a future evolution of CableCard specifications, this command already anticipates a multi-host binding capability. But the CAS currently imposes the value of the field 'nb_of_hosts' to be equal to one. In case another value is provided, the command will be negatively acknowledged with a TOO_MANY_ITEMS error code…

COMMAND 57: BIND THE CABLECARD WITH THE HOST			
Field	**Size**	**format**	**Description**
command_id	4	r_num	command_id = 0057
POD_id	20	r_num	Identifier of the POD (i.e. CableCard) in decimal. Range: 0 to 18446744073709551615
nb_of_hosts	2	r_num	Number of hosts provided in this command. Range: 01 to 99. **Current value must be one (see the previous important remark)!**
for(i=0;i<nb_of_ hosts;i++) {			Counter
host_id	13	r_num	Identifier of the Host (i.e. STB) in decimal. Range: 0 to 1099511627775
}			

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
POD_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
POD_id	CARD_NOT_PAIRED	NO_EXTENDED_ERROR_CODE
nb_of_hosts	TOO_MANY_ITEMS	NO_EXTENDED_ERROR_CODE
nb_of_hosts	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
host_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE

4.8.29 Command 60: Immediate Call Back

SAS replaces original mop by sop	no

The SMS uses this command to force a smart card to call back "immediately" the CAS. Due to the architecture of the system, an immediate call means that the CAS expects to receive a call back from a given smart card in a time frame from a couple of seconds to several hours. The best case depends on the setting of the CAS and on the STB behavior regarding such call back. The worst case is several hours; this is due to the behavior of the STB. Indeed, when a STB performs a call back, and if the call fails for any reason (e.g. the line is busy), the STB will try again within a period of a couple of minutes to several hours.

Important

- CbDate and CbTime fields are optional.

COMMAND 60: IMMEDIATE CALL BACK				
Field		**Size**	**Format**	**Description**
command_id		4	r_num	command_id = 0060
optional	CbDate	8	YYYYMMDD	The field *CbDate* is recalculated by the CAS. Consequently, this parameter can be any valid date. It is recommended to set this date as the date of "today".
	CbTime	6	HHMMSS	The field *CbTime* is recalculated by the CAS. Consequently, this parameter can be any valid time. It is recommended to set this time as "000000".

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
MOP_PPID	CB_PROFILE_NOT_FOUND	NO_EXTENDED_ERROR_CODE

4.8.30 Command 61: Enable Automatic Call Back

SAS replaces original mop by sop	no

The SMS uses this command to activate the automatic feedback functionality of a smart card. Please refer to section 4.10 (Feedback commands) for more details related to the information received by the SMS when a smart card/STB performs a call back.

Important

- If the SMS should, for any reason, issue this command a second time for a given smart card, the *date_first_call* parameter cannot be set with a date that is earlier than the date sent the first time.
- CbTime field is optional.

COMMAND 61: ENABLE AUTOMATIC CALL BACK				
Field		**Size**	**Format**	**Description**
command_id		4	r_num	command_id = 0061
call_freq		2	r_num_x	01 = annual 02 = semi-annual 03 = quarterly 04 = monthly 05 = semimonthly 1m = every m days (1 should be considered as a flag) m range is 1 .. F (F meaning 15 days)
date_first_call		8	YYYYMMDD	First date (UTC) on which the smart card should call back.
optional	CbTime	6	HHMMSS	The field *CbTime* is recalculated by the CAS. Consequently, this parameter can be any valid time. It is recommended to set this time as "000000".

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
call_freq	BAD_COMMAND_SYNTAX	BAD_FREQUENCY_FORMAT
date first call	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT
date_first_call	BAD_USAGE	BAD_DATE_SEQUENCE
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
MOP_PPID	CB_PROFILE_NOT_FOUND	NO_EXTENDED_ERROR_CODE

4.8.31 Command 62: Disable Automatic Call Back

SAS replaces original mop by sop	no

The SMS uses this command to deactivate the automatic call back functionality of a smart card.

The call back triggered by memory full, credit threshold limit reached, special events, or as a consequence of an immediate callback command will still occur.

COMMAND 62: DISABLE AUTOMATIC CALL BACK			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0062

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
MOP_PPID	CB_PROFILE_NOT_FOUND	NO_EXTENDED_ERROR_CODE

4.8.32 Command 69: Send Generic IRD Command

SAS replaces original mop by sop	no

The SMS uses this command to send an "IRD command" to a decoder. The decoder will execute an operation based on the "IRD command" instruction (ex: reset PIN code, set network ID, force tune, etc.). In other words, this command is not intended to modify the data of the smart card, but it is intended to start an action executed by the decoder. The CAS provides with this command a secure transport mechanism between the head-end and the STB.

The data structure that is sent to the STB is described below. The SMS Gateway calculates the fields "sequence_number" and "checksum" for the convenience of SMS. It means that the SMS does not provide these two parameters.

```
command_body{
   sequence_number      32 uimsbf
   command_id           8  uimsbf
   operation            8  uimsbf
   for (i=0;i<N;i++){
      data              8  uimsbf
   }
   checksum             8  uimsbf
}
```

As explained above, the purpose of an "IRD command" is to provide the decoder with a message that will conduct the decoder to execute an action (ex: reset PIN code). The "command_id" and "operation" parameters provided to the decoder represent two indexes that will indicate which subroutine should be executed by the decoder.

Nagravision provides the STB manufacturers with a list of "command_id" and "operation" that map a list of given actions. This document is referenced as "Conditional Access Kernel – IRD Command Specification".

Important

- Unlike the other commands described in this document, the cmd 69 is sensitive to the pairing status of the smartcard and the STB. Thus, the result of a cmd 69 sent to a given STB will work ONLY if the STB is paired with its smartcard. The pairing function is achieved with cmd 52.

COMMAND 69: SEND GENERIC IRD COMMAND			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0069
IRD_command_id	3	r_num	command_id field of IRD command_body. Range: 000 to 255
IRD_operation	3	r_num	Operation field of IRD command_body. Range: 000 to 255
IRD_data_length	2	r_num	Length in bytes of useful part of IRD_data field Range: 00 to 48
IRD_data	96	r_text	Hexadecimal data coded in ASCII format. The complete string must be transferred (96 chars representing 48 bytes). However, only the first left IRD_data_length bytes will be included in the data field of the IRD command_body.

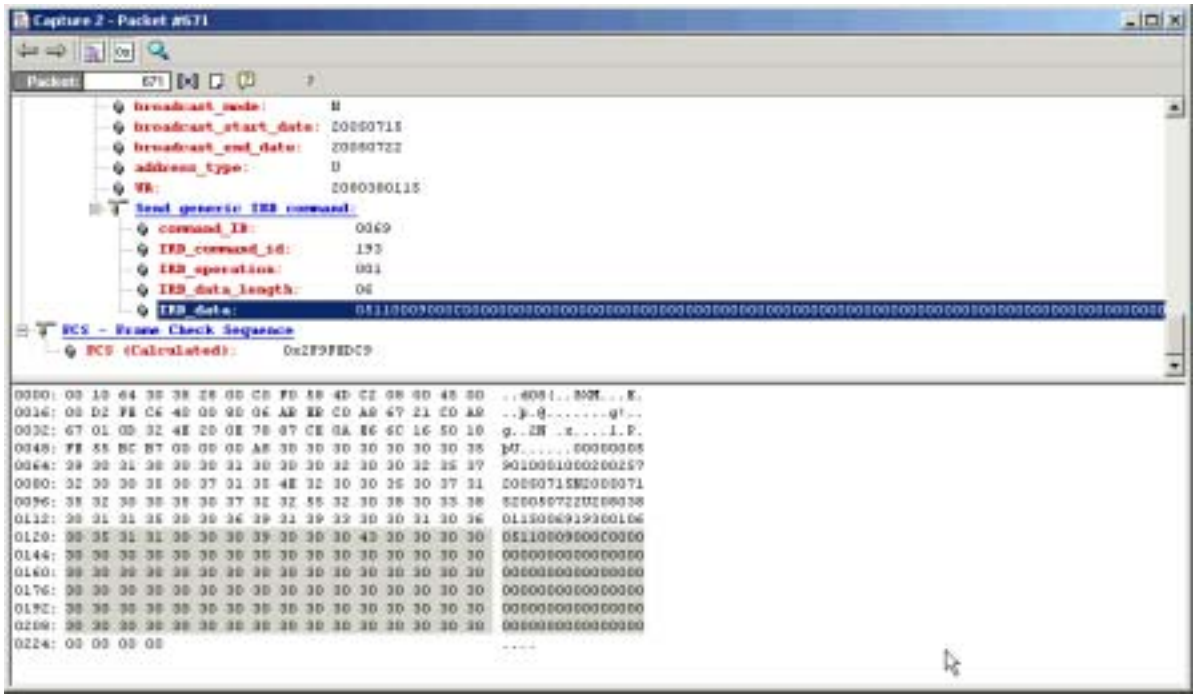
List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
IRD_command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
IRD_operation	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
IRD_data_length	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
IRD_data	BAD_COMMAND_SYNTAX	BAD_DATA_FORMAT
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE

Here is an example on how IRD_data should be formatted. In this example, the IRD-cmd “Force Tune” is sent to the STB. In the “IRD command specification” document [2], the data structure is as follows:

Parameter	Value en decimal	Value in hexadecimal	Size in byte
command_id	193	-	3
operation	001	-	3
network_id	-	05 11	2
transport_id	-	00 09	2
service_id	-	00 0C	2

In this example the “IRD_data_length” is 6. The following figure shows the detailed string of bytes of the buffer that should be sent by the SMS. Note that the remaining byte of the field “IRD_data” must be filled with char ‘0’ (in hex format 0x30).



4.8.33 Command 71: Get Products

SAS replaces original mop by sop	no

The SMS uses this command to retrieve all the non-impulsive products (subscriptions or PPV) known by the CAS for a given smart card (i.e. a given end-user).

The information comes from the CAS database. The products are returned asynchronously in command 215: Products List on the feedback port.

Impulsively purchased products can be retrieved in the same way using command 111: Get History From Call-Collector.

COMMAND 71: GET PRODUCTS			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0071

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE

4.8.34 Command 79: Force Tune

SAS replaces original mop by sop	no

The SMS uses this command to force the STB to tune to a given TV program (i.e. service). This service is identified by three parameters. Please refer to DVB specifications for detailed description.

- network: the DVB network to filter
- transport: the transport id to filter inside the network
- service: the program nb to tune inside the transport

Important

For backward compatibility, the CAS support both data format for the fields

- Network_id
- Transport_id
- Service_id

The SMS provides data in decimal format (mode 1 as illustrated below). The SMS provides data in hexadecimal format (mode 2). The default CAS package is compliant with mode 1. For mode 2 a specific CAS package should be installed.

Important

Unlike the other commands described in this document, the cmd 79 is sensitive to the pairing status of the smartcard and the STB. Thus, the result of a cmd 79 sent to a given STB will work ONLY if the STB is paired with its smartcard. The pairing function is achieved with cmd 52.

COMMAND 79: FORCE TUNE - MODE 1 (DECIMAL REPRESENTATION)			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0079
network_id	5	r_num	range is 00000 to 65535
transport_id	5	r_num	range is 00000 to 65535
service_id	5	r_num	range is 00000 to 65535

COMMAND 79: FORCE TUNE - MODE 2 (HEXADECIMAL REPRESENTATION)			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0079
network_id	4	r_num	range is 0000 to FFFF
transport_id	4	r_num	range is 0000 to FFFF
service_id	54	r_num	range is 0000 to FFFF

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE

4.8.35 Command 96: Purge PPV and IPPV Records

SAS replaces original mop by sop	no

The SMS uses this command to purge PPV, IPPV, SUB, EPR or IND records in individual smartcards that match with a deletion condition. There are two use cases to illustrate how this command works:

Example 1: to purge some old PPV and IPPV records stored in a smartcard when an end-user complains that its smartcard is full. As a result, the end-user will be able to proceed to future IPPV purchases.

Example 2: to avoid the smart card to be filled up with useless PPV and IPPV records. This can be seen as a preventive cleanup task. In this example, any time an end-user calls the SMS center to ask for additional credit, the SMS should issue this command 96.

Important

- This command will purge only IPPV records set as "call collected".
- The value (price) of PPV or IPPV that are non-watched can be refunded in the credit of the smartcard. This feature is activated with the appropriate setting of the SAS.

The SMS provides the following two parameters:

- *cleanup_date* is the date by which PPV or IPPV records older than this date will be deleted.
- *condition_date* is a parameter with a contextual meaning. It allows differentiating STB with or without a return path. For any STB with return path capability, the *condition date* = 19920101. For other STB, the *condition date = cleanup_date*

The SMS is responsible for correctly managing these parameters because the CAS system cannot verify the coherence of these parameters.

COMMAND 96: PURGE PPV AND IPPV RECORDS			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0096
cleanup_date	8	YYYYMMDD	Delete any PPV-IPPV record stored in a smart card if the PPV-IPPV expiration date is older than the cleanup_date.
condition_date	8	YYYYMMDD	- return path exists, date = 19920101 - return path does not exist, date = cleanup_date

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
cleanup_date	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT
cleanup_date	BAD_COMMAND_SYNTAX	DATE_NOT_IN_THE_PAST
condition_date	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
MOP_PPID	INVALID_PPID	ISD_MOP_NOT_FOUND

4.8.36 Command 97: Set IPPV Records as Reported

SAS replaces original mop by sop	no

The SMS uses this command to set some IPPV records on a given smart card, as reported. This command should be used to address ONLY decoders working in *prepaid without return path* mode. A decoder without return path means a decoder that is not connectable to a Nagravision Call-Collector.

Important This command should only be used in pre-paid mode environment

This command should be used by the SMS to pre-cleanup IPPV in the smart card. There are two typical examples illustrating how to use this command. The objective is the same; to avoid the smart card being filled up with useless IPPV records. This can be seen as a preventive cleanup task. What differentiates the two examples is the condition that triggers the operation:

Example 1: In this scenario, the SMS periodically and automatically sends the command 97 to specific smart cards.

Example 2: In this scenario, any time an end-user asks for additional credit (prepaid mode), the SMS issues this command 97.

COMMAND 97: SET IPPV RECORDS AS REPORTED			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0097
collect_date	8	YYYYMMDD	Set any IPPV record stored in a smart card as reported (flag “call-collected” = 1) if the IPPV expiration date is older than or equal to the parameter “collect_date”.

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
collect_date	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT
collect_date	BAD_COMMAND_SYNTAX	DATE_NOT_IN_THE_PAST
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
MOP_PPID	INVALID_PPID	ISD_MOP_NOT_FOUND

4.9 CONTROL commands (1nn)

The series of commands 1nn are utilized to set in the Call-Collector module data related with callback generated by the STB. The data provided in the command 1nn is stored in the Call-Collector database. This means that the data is persistent.

4.9.1 Command 100: Redefine Credit Limit

The SMS uses this command to set or update for a given end-user, the amount of credit that will be added in the smart card during any callback, in the Call-Collector database.

The Call-Collector will update automatically the new credit in the smart card ONLY at the next callback of the smart card-STB.

The new credit in the smart card will be the addition of the credit_limit and the current smart card debit.

Important

- The credit_limit field of the command is a 7-digit value. However, the smart card reads this value as a 5-digit integer unit, followed by 2-digit representing “cents”. Furthermore, the greatest value applicable in the smart card is 65535.99. As a consequence, for countries where the currency contains several 0 (ex: Yen), the credit value should be considered as a divided currency value, i.e. divided by 100 or 1000.
- In a Pay-TV site with pre-paid mode and with STB with return path, the credit_limit should be set to 00000.00. This means that the smart card credit will not be updated during any callback.

COMMAND 100: REDEFINE CREDIT LIMIT			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0100
credit_limit	7	r_num	credit limit range: 00000.00 to 65535.99

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
credit_limit	BAD_COMMAND_SYNTAX	BAD_CREDIT_FORMAT
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
MOP_PPID	ACCOUNT_NOT_FOUND	NO_EXTENDED_ERROR_CODE

4.9.2 Command 101: Set Authorized Phone Number

The SMS uses this command to set or update the phone numbers of a given end-user, in the Call-Collector database. Those numbers are the authorized phone numbers from where a smart card is allowed to perform any callback.

The caller id (i.e. the phone number from where the smart card-STB is calling) will be checked with those authorized phone numbers upon each callback. The CAS will generate the warning *command 205* if the phone number used during the callback is not authorized.

COMMAND 101: SET AUTHORIZED PHONE NUMBER			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0101
phone_number_1	16	p_num	Primary phone number. This field must be padded with trailing ASCII space characters for numbers not requiring 16 digits.
phone_number_2	16	p_num	Alternate phone number. This field must be padded with trailing ASCII space characters for numbers not requiring 16 digits.
phone_number_3	16	p_num	Alternate phone number. This field must be padded with trailing ASCII space characters for numbers not requiring 16 digits.

List of error codes:

Field	**Error codes**	**Error codes extension**
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
phone_number_1	BAD_COMMAND_SYNTAX	BAD_PHONE_NUMBER_FORMAT
phone_number_2	BAD_COMMAND_SYNTAX	BAD_PHONE_NUMBER_FORMAT
phone_number_3	BAD_COMMAND_SYNTAX	BAD_PHONE_NUMBER_FORMAT
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
MOP_PPID	CB_PROFILE_NOT_FOUND	NO_EXTENDED_ERROR_CODE

4.9.3 Command 104: Create ICC On Call Collector

The SMS uses this command to create a new ICC record in the Call-collector database.

Important

This command is redundant with the "initialize command (51)". The only reason to support this command is to be backward compatible with SMS operating DN2 systems. Consequently for new SMS – CAS interfaces, this command should not be used.

COMMAND 104: CREATE ICC ON CALL COLLECTOR			
Field	**Size**	**Format**	**Description**
command_id	4	0104	Command ID = 104
STU_number	14	Num	Nagravision STB serial number in decimal: 10 digits, with 4 trailing spaces. Ex: "1234567890 "

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
STU_number	BAD_COMMAND_SYNTAX	BAD_STU_NUMBER_FORMAT
STU_number	STU_ALREADY_EXISTS	NO_EXTENDED_ERROR_CODE

4.9.4 Command 105: Cancel ICC On Call-Collector

The SMS uses this command to cancel an existing smart card record in the Call-Collector database. This command should be issued when the smart card is deleted from the SMS database.

Important

The different records for a given end-user in the CAS database are set as cancelled. As a result of this command, the CAS will not accept any callback from STB. Furthermore, a cancelled smart card on the Call-Collector will not report a command 206 (STB responding status).

COMMAND 105: CANCEL ICC ON CALL-COLLECTOR			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0105

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
MOP_PPID	CB_PROFILE_NOT_FOUND	NO_EXTENDED_ERROR_CODE
MOP_PPID	ACCOUNT_NOT_FOUND	NO_EXTENDED_ERROR_CODE

4.9.5 Command 110: EMM cleanup

The SMS uses this command to "delete" all EMM in the CAS and EMM broadcaster for a given smart card. This command should be used if a discrepancy is suspected between SMS customer data and EMM for this customer.

COMMAND 110: EMM CLEANUP			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0110

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE

4.9.6 Command 111: Get History From Call-Collector

The SMS uses this command to retrieve a list of IPPV information stored in the CAS database. The list of IPPV represents the history of IPPV items that have been call-collected during a given period of time. The period is set in the CAS once and for all. The CAS response will be generated in the *Feedback* channel. The IPPV records will be generated in one of the following format:

- cmd 202 (multiple)
- cmd 216
- cmd 217

The command type is set once and for all in the CAS system.

COMMAND 111: GET HISTORY FROM CALL-COLLECTOR			
Field	**byte**	**format**	**Description**
command_id	4	r_num	command_id = 0111

List of error codes:

Field	**Error codes**	**Error codes extension**
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE

4.9.7 Command 120: Enable Callback Rules

The SMS uses this command to activate, in the Call-Collector, the processing of the callback rules for a given smart card.

During the next callback from the STB, the CALL-COLLECTOR will check the PPV purchase history of the corresponding end-user against the various callback rules defined in the CAS database. The first matching rule will (possibly) define a new credit limit and a new callback interval for the smart card.

Please refer to the document 'SAS ITM DNASP-3 User Guide' for a complete description of this marketing facility.

COMMAND 120: ENABLE CALLBACK RULES			
Field	**Size**	**Format**	**Description**
command_id	4	0120	Command ID = 120

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
MOP_PPID	CB_PROFILE_NOT_FOUND	NO_EXTENDED_ERROR_CODE

4.9.8 Command 121: Disable Callback Rules

The SMS uses this command to deactivate, in the Call-Collector, the processing of the callback rules for a given smart card and to use a new credit limit as well as a new callback interval for the smart card.

During the next callback from the STB, the CALL-COLLECTOR will no longer check the purchase history of the corresponding end-user against the various callback rules defined in the CAS database. Instead, it will use the new credit limit to renew the credit on the smart card and will compute the next regular callback date using the new callback interval specified in the command.

Please refer to the document 'SAS ITM DNASP-3 User Guide' for a complete description of this marketing facility.

COMMAND 121: DISABLE CALLBACK RULES			
Field	**Size**	**Format**	**Description**
command_id	4	0121	Command ID = 121
credit_limit	7	0000000 to 6553599	New credit limit within the range 0.00 to 65535.99
call_freq	2	01	annual
		02	semiannual
		03	quarterly
		04	monthly
		05	semimonthly
		1m	every m days (1 should be considered as a flag)

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
credit_limit	BAD_COMMAND_SYNTAX	BAD_CREDIT_FORMAT
call_freq	BAD_COMMAND_SYNTAX	BAD_FREQUENCY_FORMAT
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
MOP_PPID	ACCOUNT_NOT_FOUND	NO_EXTENDED_ERROR_CODE
MOP_PPID	CB_PROFILE_NOT_FOUND	NO_EXTENDED_ERROR_CODE

4.9.9 Command 122: Set Network

The SMS uses this command to set or update the CAS network ID and STB context associated to the

ICC.

In an advanced broadcast system, there are several data paths for the EMM. The CAS network ID is one of the parameters of the CAS to set this data path.

In non Nagravision native STB, the EMM external format is different. The STB context value allows the CAS system to set the appropriate EMM external format.

Note

- If the smart card does not yes exist in the SAS database, then this command creates it.
- When several CAS networks or STB contexts are applicable for a given Head End, then this command must be the first command issued by the SMS for a new smart card. Otherwise the EMM generated by the SMS commands are either not routed to the appropriate CAS network, or not processed by the STB (wrong EMM external format).
- If the CAS network is changed, then the EMM that are currently broadcast on the previous CAS network are not reported to the new CAS network.

COMMAND 122: SET NETWORK			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0122
CAS_network_id	3	r_num	CAS network identifier. The range is 000 to 999
STB_context	2	r_num	STB context. Combination of STB kind (e.g. Nagravision native STB or MediaGuard) and CAK kind (e.g. Merlin or MediaGuard). The range is 00 to 99

List of error codes:

Field	**Error codes**	**Error codes extension**
command_id	BAD_COMMAND_SYNTAX	NO_EXTENDED_ERROR_CODE
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
UA	BAD_COMMAND_SYNTAX	BAD_UA_FORMAT
CAS_network_id	BAD_COMMAND_SYNTAX	BAD_NETWORK_FORMAT
CAS_network_id	NETWORK_NOT_FOUND	NO_EXTENDED_ERROR_CODE
STB_context	BAD_COMMAND_SYNTAX	BAD_STB_CONTEXT_FORMAT
STB_context	STB_CONTEXT_NOT_FOUND	NO_EXTENDED_ERROR_CODE

FEEDBACK commands (2nn)

The commands described in this section are commands sent by the Nagravision CAS system to the SMS. The “FEEDBACK” commands are used to provide to the SMS information and data related to the IPPV functionality available to a customer. For instance, the “FEEDBACK” command will report to the SMS the list of events (movies) that have been impulsively purchased by the customer. In this context, an impulsive purchase means that the customer purchased a movie with its remote control.

Depending on the CAS configuration up to six kinds of FEEDBACK commands will be generated and sent to the SMS:

Command 211: Start of report

Command 201: Current credit and debit

for (I=0; I<nb_new_PPV; I++)

{

Command 202:PPV purchase list

}

Command 205: Calling phone discrepancies (if ANI is enabled)

Command 206:STU Responding Status (responding = Y)

Command 212: End of report

Important

The different types of command 2nn that are in the report generated by the CAS can be selected. As a result, the typical recommended report structure based on the Nagravision experience should be

Command 211: Start of report

for (I=0; I<nb_new_PPV; I++)

{

Command 202:PPV purchase list

}

Command 212: End of report

In case a STB does not callback within the designated time frame defined in the Call-Collector database, the following FEEDBACK command will be generated:

Command 206: STU Responding Status (responding = N)

Important

In the scenario described above, the CAS will only issue the command 206. The command 211 and 212 will not be sent.

4.9.10 Command 200: Low credit alarm

The CAS uses this command to report to the SMS that the origin of the last callback (generated by the smart card) was due to a low credit condition.

The low credit condition is reached when the credit and debit in the smart card is smaller than the threshold credit.

COMMAND 200: LOW CREDIT ALARM			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0200
STU_number	14	num	Nagravision STB serial number
credit	7	r_num	Credit amount representing 00000.00 to 65535.99
debit	7	r_num	Debit amount representing 00000.00 to 65535.99

List of error codes:

Field	**Error codes**	**Error codes extension**
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
STU_number	BAD_COMMAND_SYNTAX	BAD_STU_NUMBER_FORMAT
credit	BAD_COMAMND_SYNTAX	BAD_CREDIT_FORMAT
debit	BAD_COMMAND_SYNTAX	BAD_DEBIT_FORMAT

4.9.11 Command 201: Current Debit and Credit

The CAS uses this command to report to the SMS the credit and debit (of the smart card) reported by the smart card during the last callback.

COMMAND 201: CURRENT DEBIT AND CREDIT			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0201
STU_number	14	num	STB CA serial number ex: 72664281 will be formatted as “00000072664281”
credit	7	r_num	Credit amount representing 00000.00 to 65535.99
debit	7	r_num	Debit amount representing 00000.00 to 65535.99

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
STU_number	BAD_COMMAND_SYNTAX	BAD_STU_NUMBER_FORMAT
credit	BAD_COMAMND_SYNTAX	BAD_CREDIT_FORMAT
debit	BAD_COMMAND_SYNTAX	BAD_DEBIT_FORMAT

4.9.12 Command 202: PPV Purchase List

The CAS uses this command to report to the SMS each new IPPV item that has been communicated by the smart card during the last callback. Consequently, this command is sent as many times as there are new IPPV items provided during the last callback.

COMMAND 202: PPV PURCHASE LIST			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0202
STU_number	14	num	STB CA serial number ex: 72664281 will be formatted as “00000072664281”
IMS_product_id	12	num	IMS_product_id of the event product purchased impulsively through the EPG
purchase_date	8	YYYYMMDD	Date of IPPV purchase (UTC)
watched_status	1	r_text	Indicates if the IPPV has been watched (Y) or (N). The IPPV is watched if the accumulated viewing time exceeds the limit defined for the event or the limit predefined for the service carrying the event.

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
STU_number	BAD_COMMAND_SYNTAX	BAD_STU_NUMBER_FORMAT
IMS_product_id	BAD_COMAMND_SYNTAX	BAD_IMS_PRODUCT_ID_FORMAT
purchase_date	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT
watched_status	BAD_COMMAND_SYNTAX	BAD_FLAG_FORMAT

4.9.13 Command 203: PPV Purchase List Report

The CAS uses this command to report to the SMS the full list of new IPPV items that have been communicated by the smart card during the last callback. Consequently, this command is sent only once with the complete list of new IPPV items provided during the last callback.

COMMAND 203: PPV PURCHASE LIST REPORT			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	Command ID = 203
STU_number	14	num	STB CA serial number ex: 72664281 will be formatted as “00000072664281”
nb_of_ppv	2	num	Number of PPV provided in this command
for(i=0;i<nb_of_ ppv;i++) {			Counter
IMS_product_id	12	num	IMS product ID of the event product purchased impulsively through the EPG
purchase_date	8	YYYYMMDD	Date of IPPV purchase (UTC)
watched_status	1	r_text	Indicates if the IPPV has been watched (Y) or (N). The IPPV is watched if the accumulated viewing time exceeds the limit defined for the event or the limit predefined for the service carrying the event.
}			

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
STU_number	BAD_COMMAND_SYNTAX	BAD_STU_NUMBER_FORMAT
IMS_product_id	BAD_COMAMND_SYNTAX	BAD_IMS_PRODUCT_ID_FORMAT
purchase_date	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT
watched_status	BAD_COMMAND_SYNTAX	BAD_FLAG_FORMAT

4.9.14 Command 205: Phone Discrepancies

The CAS uses this command to report to the SMS that there was a phone discrepancy during the last callback (generated by the smart card).

A phone discrepancy warning will occur if the customer's phone number (caller_ID) does not match any authorized phone numbers, stored in the CAS database, for that given customer.

The number can be a phone number, an IP address or a MAC address. Please refer to command 49, 54 or 55 definition.

COMMAND 205: PHONE DISCREPANCIES			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0205
STU_number	14	num	STB CA serial number Ex: 72664281 will be formatted as "00000072664281"
phone_number_1	16	num	First phone number stored in the CC
phone_number_2	16	num	Second phone number stored in the CC
phone_number_3	16	num	Third phone number stored in the CC
abnormal_phone	16	num	Number used by STB to call CC

List of error codes:

Field	**Error codes**	**Error codes extension**
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
STU_number	BAD_COMMAND_SYNTAX	BAD_STU_NUMBER_FORMAT
Phone_number_1	BAD_COMAMND_SYNTAX	BAD_PHONE_NUMBER_FORMAT
Phone_number_2	BAD_COMMAND_SYNTAX	BAD_PHONE_NUMBER_FORMAT
Phone_number_3	BAD_COMMAND_SYNTAX	BAD_PHONE_NUMBER_FORMAT
abnormal_phone	BAD_COMMAND_SYNTAX	BAD_PHONE_NUMBER_FORMAT

4.9.15 Command 206: STU Responding Status

The CAS uses this command to report to the SMS that a given end-user smart card did not make a callback as expected. Only *automatic* and *immediate* callback types are concerned.

A non-responding STB is defined by the Call-Collector's callback expiration parameter.

COMMAND 206: STU RESPONDING STATUS			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0206
STU_number	14	num	STB CA serial number Ex: 72664281 will be formatted as "00000072664281"
responding	1	r_text	A STB may be responding: Y or N

List of error codes:

Field	**Error codes**	**Error codes extension**
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
STU_number	BAD_COMMAND_SYNTAX	BAD_STU_NUMBER_FORMAT
responding	BAD_COMAMND_SYNTAX	BAD_FLAG_FORMAT

4.9.16 Command 207: ICC Memory Full Alarm

The CAS uses this command to report to the SMS that the origin of the last callback (generated by the smart card) was due to a smart card memory full condition.

This information is retrieved by the CallCollector during the last callback. The obsolete data (i.e. expired subscriptions or IPPV) in the smart card will be deleted. Consequently, the smart card memory is released for subsequent usage.

Important

If the smart card is loaded with IPPV items that are entitlements for movies broadcast in the future, then those items won't be deleted.

COMMAND 207: ICC MEMORY FULL ALARM			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0207
STU_number	14	num	STB CA serial number Ex: 72664281 will be formatted as "00000072664281"

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
STU_number	BAD_COMMAND_SYNTAX	BAD_STU_NUMBER_FORMAT

4.9.17 Command 211: Start of Report

The CAS uses this command to warn the SMS that a set of *FEEDBACK* messages (i.e. 2nn commands) for a given end-user will follow. The set of messages are bound or terminated with command 212.

COMMAND 211: START OF REPORT			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0211
STU_callback_date	8	YYYYMMDD	Date (UTC) of the callback being reported
STU_callback_time	6	HHMMSS	Time (UTC) of the callback being reported

List of error codes:

Field	**Error codes**	**Error codes extension**
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
STU_callback_date	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT
STU_callback_time	BAD_COMMAND_SYNTAX	BAD_TIME_FORMAT

4.9.18 Command 212: End of Report

The CAS uses this command to warn the SMS that a set of *FEEDBACK* messages (i.e. 2nn commands) for a given end-user is terminated.

COMMAND 212: END OF REPORT			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0212
number_of_IPPV	2	num	The number of IPPV reports (occurrences of command 202) that should have been sent in this report.

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
number_of_IPPV	BAD_COMMAND_SYNTAX	BAD_NUMBER_OF_IPPV_FORMAT

4.9.19 Command 215: Products List

The CAS uses this command to provide to the SMS the list of *subscription type* products known in the CAS database for a given end-user.

The SMS first sends, in the EMM/CTRL channel, a *command 71: Get Products.* Then, the CAS answers with *command 215: Product List* in the FEEDBACK channel.

COMMAND 215: PRODUCTS LIST			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0215
original_transaction_number	9	num	Transaction number of the command 71 that triggered the generation of this command
STU_number	14	r_num	STB CA serial number ex: 72664281 will be formatted as "00000072664281"
ICC_suspended	1	r_text	Smart card suspend state: Y or N
nb_of_products	2	num	Number of products returned in this command
for(i=0;i<nb_of_products; i++)			
{			
IMS_product_id	12	num	IMS product ID Range: 000000000000 to 004294967295
product_suspended	1	r_text	Product suspend state: Y or N
}			

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
original_transaction _number	BAD_COMMAND_SYNTAX	BAD_TRANSACTION_NUMBER_FORMAT
STU_number	BAD_COMMAND_SYNTAX	BAD_STU_NUMBER_FORMAT
ICC_suspended	BAD_COMMAND_SYNTAX	BAD_FLAG_FORMAT
nb_of_products	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
IMS_product_id	BAD_COMMAND_SYNTAX	BAD_IMS_PRODUCT_ID_FORMAT
product_suspended	BAD_COMMAND_SYNTAX	BAD_FLAG_FORMAT

4.9.20 Command 216: PPV Purchase List Extended

The CAS uses this command to report to the SMS each new IPPV item that has been communicated by the smart card during the last callback.

This command is sent as many times as there are new IPPV items provided during the last callback.

Important

This command is similar to *command 202: PPV Purchase List*. It provides an additional parameter: the purchase_time.

COMMAND 216: PPV PURCHASE LIST EXTENDED			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 0216
STU_number	14	num	STB CA serial number Ex: 72664281 will be formatted as "00000072664281"
IMS_product_id	12	num	IMS_product_id of the event product purchased impulsively through the EPG
purchase_date	8	YYYYMMDD	Date of IPPV purchase (UTC)
purchase_time	6	HHMMSS	Timestamp of IPPV purchase (UTC)
watched_status	1	r_text	Indicates if the IPPV has been watched (Y) or (N). The IPPV is watched if the accumulated viewing time exceeds the limit defined for the event or the limit predefined for the service carrying the event.

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
STU_number	BAD_COMMAND_SYNTAX	BAD_STU_NUMBER_FORMAT
IMS_product_id	BAD_COMAMND_SYNTAX	BAD_IMS_PRODUCT_ID_FORMAT
purchase_date	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT
purchase_time	BAD_COMMAND_SYNTAX	BAD_TIME_FORMAT
watched_status	BAD_COMMAND_SYNTAX	BAD_FLAG_FORMAT

4.9.21 Command 217: Impulse Purchase List

The CAS uses this command to report to the SMS the full list of new products (e.g. IPPV, ISUB) that have been communicated by the smartcard during the last callback. Consequently, this command is sent only once with the complete list of purchased products provided during the last callback.

COMMAND 217: IMPULSE PURCHASE LIST			
Field	**Size**	**format**	**Description**
command_id	4	r_num	command_id = 0217
STU_number	14	num	STB CA serial number Example: 72664281 will be formatted as "00000072664281"
nb_of_products	2	num	Number of products provided in this command
for(i=0;i<nb_of_products;i++) {			Counter
IMS_product_id	12	num	IMS product ID of the event product purchased impulsively through the EPG
product_type	2	num	Type of product: 00: Unknown product type 01: The product contains subscription rights 02: The product contains pay-per-view rights 03: Rental DVR product 04: Free DVR product 05: VOD rental PPV 06: The product contains pay-per-time (PPT) rights 07: The product contains pay-per-floating-period (PPFP) rights
purchase_date	8	YYYYMMDD	Date of product purchase (UTC)
purchase_time	6	HHMMSS	Timestamp of product purchase (UTC)
watched_status	1	r_text	Indicates if the product has been watched (Y) or (N). The product is watched if the accumulated viewing time exceeds the limit defined for the event or the limit predefined for the service carrying the event.
payment_status	1	r_text	Indicates if the product has been already paid (Y) or not (N). The product is paid if the purchase was made impulsively using a prepaid credit (i.e. the credit limit of the account used for the purchase is null).
} // for loop			

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
command_id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
STU_number	BAD_COMMAND_SYNTAX	BAD_STU_NUMBER_FORMAT
IMS_product_id	BAD_COMAMND_SYNTAX	BAD_IMS_PRODUCT_ID_FORMAT
product_type	BAD_COMAMND_SYNTAX	BAD_NUMBER_FORMAT
product_type	BAD_PRODUCT_TYPE	NO_EXTENDED_ERROR_CODE
product_type	VALUE_OUT_OF_RANGE	NO_EXTENDED_ERROR_CODE
purchase_date	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT
purchase_time	BAD_COMMAND_SYNTAX	BAD_TIME_FORMAT
watched_status	BAD_COMMAND_SYNTAX	BAD_FLAG_FORMAT
payment_status	BAD_COMMAND_SYNTAX	BAD_FLAG_FORMAT

4.10 MACRO EMM commands (9nn)

This chapter describes the data of the command section for the 9nn SMS command series. Those commands are the same kind as the 0nn SMS command series (see previous chapter). The aim of this series is to optimize the overall command message bandwidth of the SMS - CAS interface. Indeed, for a large volume of end-users, the amount of transactions (EMM command messages) is very high. Thus the utilization of the interface should be optimized. The solution adopted here is to concatenate several commands of the 0nn series in one *macro* command.

4.10.1 Command 901: Activate Smart card

SAS replaces original mop by sop	As this Macro command contains CAS instructions that affect both sop and mop data, the replacement of the mop by sop follows the rule of unitary commands (see command 002 to 099)

This command is a 'one-shot' initialization command designed to improve the performances of the CAS when initializing a new smart card in the CAS database.

Important

- Please read the **Important** note of command 2, 10, 13, 48, 49, 52, 61, 100, 101
- If the CAS must generate a negative acknowledgement (i.e. NACK) upon processing this SMS command, the body of the command will be truncated to 999 characters long before being inserted in the field 'command_section' of the negative acknowledgement.

COMMAND 901: ACTIVATE SMART CARD			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	Command ID = 901
zip_code	5	num	end-user's zip code. Range: 00000…99999
STU_number	14	r_num	CA STB serial number in decimal. range: see important note in section 4.8.24
credit	7	r_num	Credit amount (in the local currency) representing the range: 00000.00 to 65535.99
threshold_credit	7	r_num	Lower limit under which the smart card must do a low credit call back representing the range 0.00 to 65535.99
credit_limit	7	r_num	Credit limit within the range 00000.00 to 65535.99

call_freq	2	r_num_x	01 = annual 02 = semi-annual 03 = quarterly 04 = monthly 05 = semimonthly 1m = every m days (1 should be considered as a flag) m range is 1 .. F (F meaning 15 days)
date_first_call	8	YYYYMMDD	First date (UTC) on which the smart card should call back.
cc_number_1	16	p_num	Call-Collector phone number. This field must be right padded with trailing ASCII space characters.
phone_number_1	16	p_num	Primary phone number for the call-collector. This field must be padded with trailing ASCII space characters for numbers not requiring 16 digits.
phone_number_2	16	p_num	Alternate phone number for the call-collector.
phone_number_3	16	p_num	Alternate phone number for the call-collector.
nb_of_products	2	r_num	Number of products provided in this command. The range is 00 to 99
for(i=0;i<nb_of_products;i++) {			Counter
IMS_product_id	12	num	IMS product ID Range: 000000000000 to 004294967295
begin_date	8	YYYYMMDD	Subscription begin date (UTC). Subscription is not valid before this date.
end_date	8	YYYYMMDD	Subscription end date (UTC). Subscription is not valid after this date.
}			
nb_of_ppv	2	r_num	Number of PPV product provided in this command. The range is 00 to 99
for(i=0;i<nb_of_ppv;i++) {			Counter
IMS_product_id	12	num	IMS product ID range: 000000000000 to 004294967295
length_event_name	2	r_num	Length of valid data in *event_name* field. The number of bytes should not exceed 30. This is due to the smart card storage limitation.
event_name	32	text	Event name as displayed in the PPV purchase history in the STB user interface. The number of characters must match the length *length_event_name.* The remaining bytes should be filled-up with “SPACE” characters (value = 0x20). See special note in section 4.6.1.
price	5	num	Price of the product, representing 0.00 to 999.99
}			

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
UA	UA_OUT_OF_RANGE	NO_EXTENDED_ERROR_CODE
MOP_PPID	DATABASE_ERROR	DATA_ERROR (invalid parent MOP)
MOP_PPID	CB_PROFILE_NOT_FOUND	NO_EXTENDED_ERROR_CODE
zip_code	BAD_COMMAND_SYNTAX	BAD_ZIP_CODE_FORMAT
zip_code	ZIP_NOT_FOUND	NO_EXTENDED_ERROR_CODE
STU_number	BAD_COMMAND_SYNTAX	BAD_STU_NUMBER_FORMAT
credit	BAD_COMMAND_SYNTAX	BAD_CREDIT_FORMAT
threshold credit	BAD_COMMAND_SYNTAX	BAD_THRESHOLD_CREDIT_FORMAT
threshold credit	BAD_COMMAND_SYNTAX	CREDIT_THRESHOLD_TOO_HIGH
credit_limit	BAD_COMMAND_SYNTAX	BAD_CREDIT_FORMAT
call_freq	BAD_COMMAND_SYNTAX	BAD_FREQUENCY_FORMAT
date_first_call	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT
date_first_call	BAD_USAGE	BAD_DATE_SEQUENCE
cc_number_1	BAD_COMMAND_SYNTAX	BAD_PHONE_NUMBER_FORMAT
phone_number_1	BAD_COMMAND_SYNTAX	BAD_PHONE_NUMBER_FORMAT
phone_number_2	BAD_COMMAND_SYNTAX	BAD_PHONE_NUMBER_FORMAT
phone_number_3	BAD_COMMAND_SYNTAX	BAD_PHONE_NUMBER_FORMAT
nb_of_products	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
IMS_product_id	BAD_COMMAND_SYNTAX	BAD_IMS_PRODUCT_ID_FORMAT
IMS_product_id	PRODUCT_NOT_FOUND	NO_EXTENDED_ERROR_CODE
IMS_product_id	BAD_PRODUCT_TYPE	PPV_PRODUCT
IMS_product_id	BAD_PRODUCT_TYPE	RECULAR_PRODUCT
IMS_product_id	BAD_PRODUCT_TYPE	DRAFT_PRODUCT
IMS_product_id	BAD_PRODUCT_STATUS	CANCELLED_PRODUCT
IMS_product_id	BAD_PRODUCT_STATUS	SUSPENDED_PRODUCT
IMS_product_id	BAD_PRODUCT_STATUS	INVALID_PURCHASE_DATE
IMS_product_id	PPV_IN_THE_PAST	NO_EXTENDED_ERROR_CODE
begin_date	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT
begin_date	BAD_COMMAND_SYNTAX	BAD_DATE_SEQUENCE
end_date	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT
end_date	BAD_COMMAND_SYNTAX	BAD_DATE_SEQUENCE
length_event_name	BAD_COMMAND_SYNTAX	LENGTH_TOO_LONG
length_event_name	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT

price	BAD_COMMAND_SYNTAX	BAD_PRICE_FORMAT
None	DATABASE_ERROR	BAD_PARAM_IN_CASDB

4.10.2 Command 902: Activate Smart card Without Return Path

SAS replaces original mop by sop	As this Macro command contains CAS instructions that affect both sop and mop data, the replacement of the mop by sop follows the rule of unitary commands (see command 002 to 099)

This command is a 'one-shot' initialization command designed to improve the performances of the CAS when initializing a new smart card in the CAS database.

Important

- Please read the **Important** note of command 2, 10, 13, 48, 49, 52, 61, 100, 101
- If the CAS must generate a negative acknowledgement (i.e. NACK) upon processing this SMS command, the body of the command will be truncated to 999 characters long before being inserted in the field 'command_section' of the negative acknowledgement.

COMMAND 902: ACTIVATE SMART CARD WITHOUT RETURN PATH			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	Command ID = 902
zip_code	5	num	End-user's zip code. Range: 00000…99999
STU_number	14	r_num	CA STB serial number in decimal. Range: see important note in section 4.8.24
nb_of_products	2	r_num	Number of products provided in this command. The range is 00 to 99
for(i=0;i<nb_of_products;i++) {			Counter
IMS_product_id	12	num	IMS product ID range: 000000000000 to 004294967295
begin_date	8	YYYYMMDD	Subscription begin date (UTC). Subscription is not valid before this date.
end_date	8	YYYYMMDD	Subscription end date (UTC). Subscription is not valid after this date.
}			
nb_of_ppv	2	num	Number of PPV products provided in this command. The range is 00 to 99
for(i=0;i<nb_of_ppv;i++) {			Counter
IMS_product_id	12	num	IMS product ID range: 000000000000 to 004294967295
length_event_name	2	r_num	Length of valid data in *event_name* field. The number of bytes should not exceed 30. This is due to the smart card storage limitation.

event_name	32	text	Event name as displayed in the PPV purchase history in the STB user interface. The number of characters must match the length *length_event_name.* The remaining bytes should be filled-up with “SPACE” characters (value = 0x20). See special note in section 4.6.1.
price	5	num	Price of the product, representing 0.00 to 999.99
}			

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
UA	UA_OUT_OF_RANGE	NO_EXTENDED_ERROR_CODE
MOP_PPID	DATABASE_ERROR	DATA_ERROR (invalid parent MOP)
MOP_PPID	CB_PROFILE_NOT_FOUND	NO_EXTENDED_ERROR_CODE
zip_code	BAD_COMMAND_SYNTAX	BAD_ZIP_CODE_FORMAT
zip_code	ZIP_NOT_FOUND	NO_EXTENDED_ERROR_CODE
STU_number	BAD_COMMAND_SYNTAX	BAD_STU_NUMBER_FORMAT
nb_of_products	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
IMS_product_id	BAD_COMMAND_SYNTAX	BAD_IMS_PRODUCT_ID_FORMAT
IMS_product_id	PRODUCT_NOT_FOUND	NO_EXTENDED_ERROR_CODE
IMS_product_id	BAD_PRODUCT_TYPE	PPV_PRODUCT
IMS_product_id	BAD_PRODUCT_TYPE	RECULAR_PRODUCT
IMS_product_id	BAD_PRODUCT_TYPE	DRAFT_PRODUCT
IMS_product_id	BAD_PRODUCT_STATUS	CANCELLED_PRODUCT
IMS_product_id	BAD_PRODUCT_STATUS	SUSPENDED_PRODUCT
IMS_product_id	BAD_PRODUCT_STATUS	INVALID_PURCHASE_DATE
IMS_product_id	PPV_IN_THE_PAST	NO_EXTENDED_ERROR_CODE
begin_date	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT
begin_date	BAD_COMMAND_SYNTAX	BAD_DATE_SEQUENCE
end_date	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT
end_date	BAD_COMMAND_SYNTAX	BAD_DATE_SEQUENCE
length_event_name	BAD_COMMAND_SYNTAX	LENGTH_TOO_LONG
length_event_name	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
price	BAD_COMMAND_SYNTAX	BAD_PRICE_FORMAT
None	DATABASE_ERROR	BAD_PARAM_IN_CASDB

4.10.3 Command 903: Add List of Products

SAS replaces original mop by sop	no

The SMS uses this command to add a list of new products (service products, non-impulsively purchasable event products and package products) in a smart card.

Important

- Please read the **Important** note of command 2, 10, 13, 48, 49, 52, 61, 100, 101
- If the CAS must generate a negative acknowledgement (i.e. NACK) upon processing this SMS command, the body of the command will be truncated to 999 characters long before being inserted in the field 'command_section' of the negative acknowledgement.

COMMAND 903: ADD LIST OF PRODUCTS			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	Command ID = 903
nb_of_products	2	r_num	Number of products provided in this command. The range is 00 to 99
for(i=0;i<nb_of_products;i++) {			Counter
IMS_product_id	12	num	IMS product ID Range: 000000000000 to 004294967295
begin_date	8	YYYYMMDD	Subscription begin date (UTC). Subscription is not valid before this date.
end_date	8	YYYYMMDD	Subscription end date (UTC). Subscription is not valid after this date.
}			
nb_of_ppv	2	r_num	Number of PPV product provided in this command. The range is 00 to 99
for(i=0;i<nb_of_ppv;i++) {			Counter
IMS_product_id	12	num	IMS product ID range: 000000000000 to 004294967295
length_event_name	2	r_num	Length of valid data in *event_name* field. The number of bytes should not exceed 30. This is due to the smart card storage limitation.
event_name	32	text	Event name as displayed in the PPV purchase history in the STB user interface. The number of characters must match the length *length_event_name.* The remaining bytes should be filled-up with "SPACE" characters (value = 0x20). See special note in section 4.6.1.
price	5	num	Price of the product, representing 0.00 to 999.99
}			

List of error codes:

Field	Error codes	Error codes extension
command_id	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
UA	UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
UA	CANCELLED_CARD	NO_EXTENDED_ERROR_CODE
MOP_PPID	INVALID_PPID	ISD_MOP_NOT_FOUND
MOP_PPID	DATABASE_ERROR	DATA_ERROR (invalid parent MOP)
nb_of_products	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
IMS_product_id	BAD_COMMAND_SYNTAX	BAD_IMS_PRODUCT_ID_FORMAT
IMS_product_id	PRODUCT_NOT_FOUND	NO_EXTENDED_ERROR_CODE
IMS_product_id	BAD_PRODUCT_TYPE	PPV_PRODUCT
IMS_product_id	BAD_PRODUCT_TYPE	RECULAR_PRODUCT
IMS_product_id	BAD_PRODUCT_TYPE	DRAFT_PRODUCT
IMS_product_id	BAD_PRODUCT_STATUS	CANCELLED_PRODUCT
IMS_product_id	BAD_PRODUCT_STATUS	SUSPENDED_PRODUCT
IMS_product_id	BAD_PRODUCT_STATUS	INVALID_PURCHASE_DATE
IMS_product_id	PPV_IN_THE_PAST	NO_EXTENDED_ERROR_CODE
begin_date	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT
begin_date	BAD_COMMAND_SYNTAX	BAD_DATE_SEQUENCE
end_date	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT
end_date	BAD_COMMAND_SYNTAX	BAD_DATE_SEQUENCE
length_event_name	BAD_COMMAND_SYNTAX	LENGTH_TOO_LONG
length_event_name	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
price	BAD_COMMAND_SYNTAX	BAD_PRICE_FORMAT

4.11 OPERATION commands (10nn)

4.11.1 Command 1000: Acknowledge Command

The CAS and the SMS use this command to acknowledge a command issued by the other side of the interface. The field “transaction_number” identifies the corresponding command that has been processed successfully.

Important

The fields IMS_product_id and SMS_product_id are no longer applicable since the series of cmd 3nn is not supported by the interface. Consequently, these fields will be set to 000000000000 and 000000000000 respectively.

COMMAND 1000: ACKNOWLEDGE COMMAND			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 1000
transaction_ number	9	num	Transaction number acknowledged Range: 000000000 to 999999999
IMS_product_id	12	num	IMS product ID is always 000000000000
SMS_product_id	12	num	SMS product ID is always 000000000000

4.11.2 Command 1001: Non-acknowledged Command

The command specified could not be completed because it is either REJECTED or POSTPONED. A rejected command is a consequence of a mistake in the command format or in the command processing. A postponed command is a consequence of a CAS system that is busy.

Important

When a command is POSTPONED, the SMS should resubmit the command after a delay of 60 min. Note that the resubmitted command should be identified with a new transaction id.

COMMAND 1001: NON-ACKNOWLEDGE COMMAND			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	Command ID = 1001
transaction_ number	9	num	transaction_number acknowledged Range: 000000000 to 999999999
nack_status	1	r_num	1 = REJECTED means that the command has been rejected because an error has been detected. 2 = POSTPONED means that the command could not be completed because the system is busy.
error_code	4	r_num	Main error code, refers to chapter 5.1
error_code_ext	4	r_num	Extension error code, refers to chapter 5.2
length_of_command_body	3	num	Length of the following section
command_section	n		Command section of the command that caused the error. n = length_of_command_body

4.11.3 Command 1002: No Command

The SMS should send this command in the following two conditions:

1. While the SMS opens a channel.
2. While there is no activity, i.e. no command to send or receive. The period should be 5 min.

The SMS should separately manage the EMM/CONTROL channel and the FEEDBACK channel:

COMMAND 1002: NO COMMAND			
Field	**Size**	**Format**	**Description**
command_id	4	r_num	command_id = 1002

5. Error codes

If an error occurs when executing an SMS command, it is reported by the use of an error code and an error code extension. The error code indicates an error category and the error code extension gives more details regarding the error source (e.g., an incorrectly specified field or value exceeded).

An error code is always followed by an error code extension (possibly extension 0000: NO_EXTENDED_ERROR_CODE). The error codes and extensions available are indicated in the following tables.

5.1 Table of Error codes

ERROR CODE NAME	CODE	DESCRIPTION
FATAL_ERROR	0000	The Oracle database interface returns an exception as database error or protocol error due to a non-understood field in the SMS command. All SMS EMM commands might generate such FATAL_ERROR.
BAD_ROOT_HEADER_SYNTAX	0001	The syntax of the command root header is not correct. See error code extension
BAD_HEADER_SYNTAX	0002	The syntax of the command header is not correct. See error code extension
BAD_COMMAND_SYNTAX	0003	The syntax of the command is not correct. See error code extension.
DATABASE_ERROR	0004	An error occurred during an IMS database query processing.
MESSAGE_NOT_FOUND	0005	The message referenced in the command does not exist.
PRODUCT_NOT_FOUND	0006	The product_id used in the command does not exist in the IMS database.
CANCELED_CARD	0007	The smart card referenced in the command has been canceled.
UA_NOT_FOUND	0008	The smart card UA referenced in the command does not exist
PPV_IN_THE_PAST	0009	The command attempts to access a PPV whose validity is expired.
STU_ALREADY_EXISTS	0010	The command attempts to create a smart card in the CC database, but the smart card is bound to an already existing STB.
SERVICE_NOT_FOUND	0011	The service referenced in the command does not exist in the IMS database.
PRODUCT_ALREADY_EXISTS	0013	The command attempts to create an already existing product.
UA_ALREADY_EXISTS	0014	The command attempts to create in the CC database a smart card that already exists.

ERROR CODE NAME	CODE	DESCRIPTION
BAD_EPG_FORMAT	0015	The format of the EPG data feed is not correct.
DB_INCONSISTENT_TOO_MANY_ROWS	0021	Inconsistency in the IMS database
DB_INCONSISTENT_INVALID_PRODUCT	0022	Inconsistency in the IMS database
PRODUCT_INCONSISTENT	0024	There is an inconsistency between the product definition received and the internal SMS Gateway product database.
TOO_MANY_ITEMS	0025	Too many items are given in the list.
VALUE_OUT_OF_RANGE	0026	The value in the incoming command is out of the authorized range.
BAD_USAGE	0027	The usage of this parameter is not correct. See error code extension.
INVALID_PPID	0028	The MOP PPID used in this command is invalid.
SYSTEM_ERROR	0029	An error not related to the business or to the field value occurs within the system. See error code extension.
BAD_PRODUCT_TYPE	0030	The type of the product is wrong (i.e. PPV instead of SUBSCRIPTION).
BAD_PRODUCT_STATUS	0031	The status of the product does not allow the execution of this command.
ACCOUNT_NOT_FOUND	0032	The account that corresponds to this UA and to this MOP does not exist in the database.
CB_PROFILE_NOT_FOUND	0033	The callback profile that corresponds to this UA and to this MOP does not exist in the database.
ZIP_NOT_FOUND	0034	The Zip code does not exist in the database.
RIGHT_NOT_FOUND	0035	The Right does not exist in the database
NO_LICENSE	0036	No license for this type of SMS command
NOT_AUTHORIZED	0037	This command is not authorized
SMS_NOT_IDENTIFIED	0038	No cmd 1002 was received on feedback channel before this command
NO_ITM_PRESENT	0039	This error is returned if cmd 71 or 111 is issued but no ITM is configured in the CAS system
NO_RTM_PRESENT	0040	A cmd 1002 is issued by SMS on EMM/control channel, but no RTM is configured in the CAS system.
SMS_NOT_AUTHORIZED	0041	Invalid source_ID used by SMS.
NOT_DEFAULT_FEEDBACK_SMS	0042	This error is returned if the SMS issues a cmd 1002 with a wrong source_ID
NO_SERVER_AVAILABLE	0043	There is no TM application available to handle the command
TM_SERVER_ERROR	0044	Occurs when system returns a CORBA exception

ERROR CODE NAME	CODE	DESCRIPTION
SOURCE_ID_ALREADY_USED	0045	The SMS Source ID is already used by another connection. Check why two different SMS are connected with the same source_ID
UA_OUT_OF_RANGE	0046	smart card Unique Address is out of range for the operator
EXPIRED_CARD	0047	The smardcard life time has expired
COMMAND_THRESHOLD_OVERFLOW	0048	This error is generated back to the SMS if there are too many commands for a given smart card coming in a short time frame.
INVALID_VOUCHER	0049	The voucher checked is invalid
NO_CIPHER_PRESENT	0050	There is no cipher available
CARD_NOT_PAIRED	0051	Card is not paired with STB.
SEGMENT_NOT_FOUND	0052	The segment requested in not defined in CAS.
PRODUCT_CAT_NOT_FOUND	0053	The product category requested in not defined in CAS.
NETWORK_NOT_FOUND	0054	The network ID requested in not defined in CAS.
STB_CONTEXT_NOT_FOUND	0055	The STB context requested in not defined in CAS.
COND_ADD_ID_NOT_FOUND	0056	The conditional add ID requested in not defined in CAS.
ONLINE_PURCHASE_REFUSED	0057	Online purchase has been refused.
UA_LOCKED_WHILE_EXCHANGED	0058	A smartcard exchange is in progress for the given UA. No other SMS command can be accepted as long as the exchange is not finished.
NO_FREE_PAIRING_SLOT	0059	All the available pairing slots are already used.
EXCHANGED_CARD	0060	The smartcard has been exchanged with a new one, therefore it cannot be used anymore.
SC_SET_ID_NOT_FOUND	0061	There is no smartcard set ID defined in CAS for the requested UA.

Table 5-1 Error codes

5.2 Table of Error code extensions

ERROR CODE EXTENSION NAME	CODE	DESCRIPTION
NO_EXTENDED_ERROR_CODE	0000	No error code extension is available for the error code specified.
BAD_DEBIT_FORMAT	0001	The command contains a debit field whose format is incorrect.
BAD_CREDIT_FORMAT	0002	The command contains a credit field whose format is incorrect.
BAD_CREDIT_MODE	0003	The command contained a credit_mode field whose value is not one of the authorized ones.
BAD_DATE_FORMAT	0004	The command contains a date whose format is incorrect.
BAD_DATE_SEQUENCE	0005	The command contains a begin date and an end date that are out of sequence.
BAD_FREQUENCY_FORMAT	0006	The field call_freq of the command contains a value whose format is incorrect.
BAD_STU_NUMBER_FORMAT	0007	The format of the STB number specified in the command is incorrect.
BAD_IMS_PRODUCT_ID_FORMAT	0008	The format of the IMS_product_id of the command is incorrect.
BAD_MESSAGE_NUMBER_FORMAT	0010	The value of the message_number field of the command is incorrect.
BAD_PHONE_NUMBER_FORMAT	0011	The format of the value of a phone_number field of the command is incorrect.
BAD_PRICE_FORMAT	0013	The value of the price field of the command is incorrect.
BAD_THRESHOLD_CREDIT_FORMAT	0014	The value of the threshold_credit field of the command is incorrect.
BAD_UA_FORMAT	0015	The value of the UA field of the command is incorrect.
BAD_ZIP_CODE_FORMAT	0016	The value of the zip code field of the command is incorrect.
DIFFERENT_PRODUCTS	0017	The command attempts to define a product with a product_id already attributed to a different product.
BAD_BROADCAST_MODE	0019	The broadcast mode is incorrect.
BAD_ADDRESS_TYPE	0020	The format of the value of the address_type field of the command is incorrect.
BAD_MOP_PPID	0021	The MOP_PPID indicated in the command is not valid.
BAD_DEST_ID	0022	The dest_id indicated in the command is not valid.
BAD_SOURCE_ID	0023	The source_id indicated in the command is not valid.

ERROR CODE EXTENSION NAME	CODE	DESCRIPTION
BAD_COMMAND_TYPE	0024	This type of SMS command does not exist.
BAD_COMMAND_ID	0025	The command ID given in the SMS command does not exist.
BAD_NUMBER_FORMAT	0027	A non-numerical character was found in the content of a Num field.
BAD_ERROR_CODE	0032	The error code specified in the command does not exist (section 8).
BAD_ERROR_CODE_EXT	0033	The error code extension specified in the command does not exist (section 8).
CREDIT_THRESHOLD_TOO_HIGH	0034	The value of the threshold_credit field of the command is too high.
BAD_SERVICE_UID_FORMAT	0040	The format of the value of a service_UID field of the command is incorrect.
BAD_SERVICE_NUMBER_FORMAT	0041	The format of the value of the service_number field of the command is incorrect.
BAD_NUMBER_OF_IPPV_FORMAT	0044	The format of the number_of_IPPV field in the command is incorrect.
BAD_IP_ADDRESS_FORMAT	0045	The format of the IP address in the command is incorrect.
EXTERNAL_SYSTEM_NOT_RESPONDING	0048	The other components of the CAS system do not respond to the gateway process.
EXTERNEL_SYSTEM_ERROR	0049	The other components of the CAS system have not successfully processed the command.
BAD_SERVICE_ID_FORMAT	0052	The format of the service ID is incorrect.
BAD_TRANSPORT_ID_FORMAT	0053	The format of the transport ID is incorrect.
BAD_NETWORK_ID_FORMAT	0054	The format of the network ID is incorrect.
BAD_LID_FORMAT	0055	The format of the lid is incorrect.
BAD_PRIORITY_FORMAT	0056	The format of the priority is incorrect.
BAD_MODE_FORMAT	0057	The format of the mode is incorrect.
LENGTH_TOO_LONG	0058	The length is out of range.
BAD_FLAG_VALUE	0059	The flag value is not recognized.
BAD_CC_PORT_FORMAT	0060	The format of the CC port is incorrect.
BAD_TRANSACTION_NUMBER_FORMAT	0061	The format of the transaction number is incorrect.
BAD_PURGE_MODE_FORMAT	0062	The format of the purge mode is incorrect.
BAD_CALLBACK_FORMAT	0063	The format of the callback type is incorrect
BAD_TIME_FORMAT	0064	The time format is incorrect
DATE_NOT_IN_THE_PAST	0065	This date must be set in the past
ACCESS_ERROR	0066	Error when trying to access database
TRANSACTION_ERROR	0067	An error occurs during a database transaction (i.e. limitation reached, overflow…)

ERROR CODE EXTENSION NAME	CODE	DESCRIPTION
DATA_ERROR	0068	An error related to the data contained in database occurs (i.e. expected record is missing).
TRANS_NR_ALREADY_IN_USE	0069	This transaction number is already in use within the system.
COMMUNICATION_ERROR	0070	Communication error between components within the system.
INTERNAL_ERROR	0071	System internal error.
SOURCE_NOT_AUTHORIZED	0072	Use of this source identifier is not allowed.
SOURCE_ALREADY_IN_USE	0073	This source identifier is already used by another source.
DEST_NOT_AUTHORIZED	0074	Use of this destination identifier is not allowed.
MOP_NOT_AUTHORIZED	0075	Use of this MOP PPID is not allowed.
DATE_IN_THE_FUTURE	0076	This date must be set in the present or in the past.
CANCELLED_PRODUCT	0077	The product has been cancelled.
SUSPENDED_PRODUCT	0078	The product has been suspended.
INVALID_PURCHASE_DATE	0079	The purchase date is invalid.
DRAFT_PRODUCT	0080	The product is a draft.
PPV_PRODUCT	0081	The product is a PPV.
DATE_IN_THE_PAST	0082	This date must be set in the present or in the future.
ADDRESS_TYPE_NOT_AUTHORIZED	0083	This type of address is not allowed for this command.
ISD_MOP_NOT_FOUND	0084	The MOP record does not exist for this UA.
BAD_DATA_FORMAT	0085	The format of the data is invalid.
REGULAR_PRODUCT	0086	This product is a regular (subscription) product.
INVALID_CATEGORY	0087	The category requested is not defined in the CAS.
CORBA_EXCEPTION	0088	Low level system error
RENTAL_PRODUCT	0089	Operation not allowed with a rental product
FREE_RENTAL_PRODUCT	0090	Operation not allowed with a Free Rental product
VOD_RENTAL_PPV	0091	Operation not allowed with a VOD Rental product

PPT_PRODUCT	0092	Operation not allowed with a Pay Per Time product
BAD_SECRET_CODE	0093	The secret code is incorrect
BAD_VERIFICATION_CODE	0094	The verification code is incorrect
BAD_THIRD_PARTY	0095	The third party is unknown
BAD_PARAM_IN_CASDB	0096	SAS is wrongly configured and cannot properly handle the SMS command for the given UA.
BAD_CATEGORY_FORMAT	0097	The format of the category is incorrect
BAD_SUBCATEGORY_FORMAT	0098	The format of the subcategory is incorrect
BAD_FREE_PRODUCT_MODE	0099	The free product mode contains an unauthorized value
BAD_PRODUCT_CAT_FORMAT	0100	The format of the product category is incorrect
BAD_NB_FREE_PROD_FORMAT	0101	The format of the number of free product is incorrect
BAD_NETWORK_FORMAT	0102	The format of the network ID is incorrect
BAD_STB_CONTEXT_FORMAT	0103	The format of the STB context is incorrect
BAD_PURCHASE_MODE	0104	The purchase mode contains an unauthorized value
BAD_COND_ADD_ID_FORMAT	0105	The format of the conditional add ID is incorrect
BAD_ONLINE_PURCHASE_MODE	0106	The online purchase mode contains an unauthorized value

Table 5-2 Error code extensions

6. UA and CA-S/N checksum

This chapter describes the procedure to manage UA (unique address) and CA-S/N (conditional access serial number) at the customer site.

6.1 Definitions

The mandatory information to allow a CAS to fully recognize a given customer is:

UA Unique address of the smart card. This is a 12-digits long number that uniquely identifies the smart card. The first 10 digits represent the address itself; the last 2 digits are a checksum allowing the SMS to verify the number given by the customer. The preferred (but not mandatory) format is:

nn nnnn nnnn cc

This number is printed and bar-coded on the smart card and is normally accessible through a set-top box menu on the TV screen.

CA-S/N Conditional access serial number. This is a 12-digits long number that uniquely identifies the set-top box for the conditional access system. The first 10 digits represent the serial number; the last 2 digits are a checksum allowing the SMS to verify the number given by the customer. The preferred (but not mandatory) format is:

nn nnnn nnnn cc

This number may be printed (and possibly bar-coded) on a sticker on the set-top box and is normally accessible through a set-top box menu on the TV screen.

This number may differ (and usually differs) from the manufacturing S/N, which identifies the set-top box for the manufacturer himself. We had to use a common format throughout the system and different numbers for all set-top boxes whatever the format chosen by the manufacturer.

6.2 Pairing operation

The pairing operation links a smart card with a set-top box. The pairing operation is mandatory. Without the pairing operation, the end-user's STB does not work.

The end-user will communicate his UA and his CA-S/N to the SMS during the installation process. Those numbers may be filled in the contract or in any place the customer finds suitable for this purpose.

The pairing key, which is used to effectively prevent the use of a smart card in another set-top box, is provided in the Nagravision system.

6.3 Data files

Nagravision usually provides the customer with files containing the list of the smart cards produced. Those files contain a list of UA. They are given by production box (250 smart cards) and/or by production batch (indeterminate number of boxes in one file).

On the other hand, the set-top box manufacturer has the responsibility to provide files containing the list of set-top boxes effectively produced. Those files must contain a list or range of CAS/N. If needed by the customer, those files may provide a link between the CAS/N and the manufacturing S/N for each box, but this is not requested by the Nagravision system.

Nagravision may possibly provide the range of CAS/N allocated for each manufacturer but has no view on what is effectively produced.

6.4 CA-S/N – Pairing keys

The CA-S/N is chosen by Nagravision and provided to the set-top box manufacturer, each of them associated with a pairing key. This pairing key is an element of security of the system and shall not be known by the end-user, by the customer, or by any unauthorized people at the manufacturing site. Thus this pairing key shall not appear on any support (label, screen, paper, accessible memory, file…). A violation of this rule is a severe security breach and may induce Nagravision into litigation actions.

6.5 Checksum algorithm

The following C routine describes the formula used for the computation of the checksum:

```
unsigned char calcCheckSum (unsigned long SN)
{
  return ((6*(SN/100000000L)+19*(SN/10000000L%10)+
           8*(SN/10000L%1000)+(SN/100L%100))%23+
          (SN%100))%100;
}
```

7. ASCII Table

Decimal	Hex	Binary	Value
00	00	00000000	NUL (Null char.)
32	20	00100000	SP (Space)
33	21	00100001	! (exclamation mark)
34	22	00100010	" (double quote)
35	23	00100011	# (number sign)
36	24	00100100	$ (dollar sign)
37	25	00100101	% (percent)
38	26	00100110	& (ampersand)
39	27	00100111	' (single quote)
40	28	00101000	((left/opening parenthesis)
41	29	00101001	) (right/closing parenthesis)
42	2A	00101010	* (asterisk)
43	2B	00101011	+ (plus)
44	2C	00101100	, (comma)
45	2D	00101101	- (minus or dash)
46	2E	00101110	. (dot)
47	2F	00101111	/ (forward slash)
48	30	00110000	0
49	31	00110001	1
50	32	00110010	2
51	33	00110011	3
52	34	00110100	4
53	35	00110101	5
54	36	00110110	6
55	37	00110111	7
56	38	00111000	8
57	39	00111001	9
58	3A	00111010	: (colon)
59	3B	00111011	; (semi-colon)
60	3C	00111100	< (less than)
61	3D	00111101	= (equal sign)
62	3E	00111110	> (greater than)
63	3F	00111111	? (question mark)
64	40	01000000	@ (AT symbol)
65	41	01000001	A
66	42	01000010	B
67	43	01000011	C
68	44	01000100	D
69	45	01000101	E
70	46	01000110	F
71	47	01000111	G
72	48	01001000	H
73	49	01001001	I
74	4A	01001010	J

75	4B	01001011	K
76	4C	01001100	L
77	4D	01001101	M
78	4E	01001110	N
79	4F	01001111	O
80	50	01010000	P
81	51	01010001	Q
82	52	01010010	R
83	53	01010011	S
84	54	01010100	T
85	55	01010101	U
86	56	01010110	V
87	57	01010111	W
88	58	01011000	X
89	59	01011001	Y
90	5A	01011010	Z
91	5B	01011011	[(left/opening bracket)
92	5C	01011100	\ (back slash)
93	5D	01011101	] (right/closing bracket)
94	5E	01011110	^ (caret/cirumflex)
95	5F	01011111	_ (underscore)
96	60	01100000	`
97	61	01100001	a
98	62	01100010	b
99	63	01100011	c
100	64	01100100	d
101	65	01100101	e
102	66	01100110	f
103	67	01100111	g
104	68	01101000	h
105	69	01101001	i
106	6A	01101010	j
107	6B	01101011	k
108	6C	01101100	l
109	6D	01101101	m
110	6E	01101110	n
111	6F	01101111	o
112	70	01110000	p
113	71	01110001	q
114	72	01110010	r
115	73	01110011	s
116	74	01110100	t
117	75	01110101	u
118	76	01110110	v
119	77	01110111	w
120	78	01111000	x
121	79	01111001	y
122	7A	01111010	z

8. Examples

8.1 Device_IO connection establishment

The following example is a network capture of one whole Device_IO connection process.

```
Packet #1
  Flags:         0x00
  Status:        0x01
  Packet Length:64
  Timestamp:     18:04:12.473073 10/09/2001
Ethernet Header
  Destination:           08:00:2B:C5:7E:2A
  Source:                00:C0:F0:3D:7F:9D
  Protocol Type:         0x0800  IP
IP Header - Internet Protocol Datagram
  Version:               4
  Header Length:         5  (20  bytes)
  Type of Service:       %00000000
  Precedence: Routine,   Normal Delay,   Normal Throughput,   Normal Reliability
  Total Length:          44
  Identifier:            63087
  Fragmentation Flags:   %010  Do Not Fragment   Last Fragment
  Fragment Offset:       0  (0  bytes)
  Time To Live:          128
  Protocol:              6  TCP - Transmission Control Protocol
  Header Checksum:       0x82E9
  Source IP Address:     192.168.0.31
  Dest. IP Address:      192.168.0.3
  No IP Options
TCP - Transport Control Protocol
  Source Port:           1090  ff-fms
  Destination Port:      20000  sms_gateway
  Sequence Number:       55594
  Ack Number:            0
  Offset:                6  (24  bytes)
  Reserved:              %000000
  Code:                  %000010    Synch
  Window:                8192
  Checksum:              0xCB26
  Urgent Pointer:        0
TCP Options:
  Option Type:           2  Maximum Segment Size
    Length:              4
    MSS:                 1460

  TCP Data Area:         No more data.
Extra bytes (Padding):
  ..                  00 00
FCS - Frame Check Sequence
  FCS (Calculated):      0x5FAFB276
```

```
Packet #2
  Flags:         0x00
  Status:        0x01
  Packet Length:64
  Timestamp:     18:04:12.473409 10/09/2001
Ethernet Header
  Destination:           00:C0:F0:3D:7F:9D
  Source:                08:00:2B:C5:7E:2A
  Protocol Type:         0x0800  IP
IP Header - Internet Protocol Datagram
  Version:               4
  Header Length:         5  (20  bytes)
  Type of Service:       %00000000
  Precedence: Routine,   Normal Delay,   Normal Throughput,   Normal Reliability
  Total Length:          44
  Identifier:            56682
  Fragmentation Flags:   %010  Do Not Fragment   Last Fragment
  Fragment Offset:       0  (0  bytes)
  Time To Live:          60
  Protocol:              6  TCP - Transmission Control Protocol
  Header Checksum:       0xDFEE
  Source IP Address:     192.168.0.3
  Dest. IP Address:      192.168.0.31
  No IP Options
TCP - Transport Control Protocol
  Source Port:           20000  sms_gateway
  Destination Port:      1090  ff-fms
  Sequence Number:       1524200406
  Ack Number:            55595
  Offset:                6  (24  bytes)
  Reserved:              %000000
```

```
  Code:                 %010010    Ack    Synch
  Window:               33580
  Checksum:             0x9939
  Urgent Pointer:       0
TCP Options:
  Option Type:          2  Maximum Segment Size
    Length:             4
    MSS:                1460

  TCP Data Area:        No more data.
Extra bytes (Padding):
  ..                  00 00
FCS - Frame Check Sequence
  FCS (Calculated):     0x36B97026
```

```
Packet #3
  Flags:        0x00
  Status:       0x01
  Packet Length:64
  Timestamp:    18:04:12.473450 10/09/2001
Ethernet Header
  Destination:          08:00:2B:C5:7E:2A
  Source:               00:C0:F0:3D:7F:9D
  Protocol Type:        0x0800  IP
IP Header - Internet Protocol Datagram
  Version:              4
  Header Length:        5  (20  bytes)
  Type of Service:      %00000000
  Precedence: Routine,   Normal Delay,   Normal Throughput,   Normal Reliability
  Total Length:         40
  Identifier:           63343
  Fragmentation Flags:  %010  Do Not Fragment   Last Fragment
  Fragment Offset:      0  (0  bytes)
  Time To Live:         128
  Protocol:             6  TCP - Transmission Control Protocol
  Header Checksum:      0x81ED
  Source IP Address:    192.168.0.31
  Dest. IP Address:     192.168.0.3
  No IP Options
TCP - Transport Control Protocol
  Source Port:          1090  ff-fms
  Destination Port:     20000  sms_gateway
  Sequence Number:      55595
  Ack Number:           1524200407
  Offset:               5  (20  bytes)
  Reserved:             %000000
  Code:                 %010000    Ack
  Window:               8760
  Checksum:             0x11EB
  Urgent Pointer:       0
  No TCP Options
  TCP Data Area:        No more data.
Extra bytes (Padding):
  ......              00 00 00 00 00 00
FCS - Frame Check Sequence
  FCS (Calculated):     0x62217FDC
```

```
Packet #4
  Flags:        0x00
  Status:       0x01
  Packet Length:69
  Timestamp:    18:04:12.485934 10/09/2001
Ethernet Header
  Destination:          08:00:2B:C5:7E:2A
  Source:               00:C0:F0:3D:7F:9D
  Protocol Type:        0x0800  IP
IP Header - Internet Protocol Datagram
  Version:              4
  Header Length:        5  (20  bytes)
  Type of Service:      %00000000
  Precedence: Routine,   Normal Delay,   Normal Throughput,   Normal Reliability
  Total Length:         51
  Identifier:           63599
  Fragmentation Flags:  %010  Do Not Fragment   Last Fragment
  Fragment Offset:      0  (0  bytes)
  Time To Live:         128
  Protocol:             6  TCP - Transmission Control Protocol
  Header Checksum:      0x80E2
  Source IP Address:    192.168.0.31
  Dest. IP Address:     192.168.0.3
  No IP Options
TCP - Transport Control Protocol
  Source Port:          1090  ff-fms
  Destination Port:     20000  sms_gateway
  Sequence Number:      55595
  Ack Number:           1524200407
  Offset:               5  (20  bytes)
  Reserved:             %000000
  Code:                 %011000    Ack    Push
  Window:               8760
  Checksum:             0xC9C3
  Urgent Pointer:       0
  No TCP Options
SMSgateway  spec. 2.6.2 - decoder v.0.8
```

```
DeviceIO: message_1
  len:                  9
  op_mode:              1  Fast data transfer (do not allow tracing)
  ob_name_len:          7
  ob_name:              SMS_GWY
FCS - Frame Check Sequence
  FCS (Calculated):     0xC6762DDA
```

```
Packet #5
  Flags:        0x00
  Status:       0x01
  Packet Length:64
  Timestamp:    18:04:12.486303 10/09/2001
Ethernet Header
  Destination:          00:C0:F0:3D:7F:9D
  Source:               08:00:2B:C5:7E:2A
  Protocol Type:        0x0800  IP
IP Header - Internet Protocol Datagram
  Version:              4
  Header Length:        5  (20  bytes)
  Type of Service:      %00000000
  Precedence: Routine,   Normal Delay,   Normal Throughput,   Normal Reliability
  Total Length:         43
  Identifier:           56683
  Fragmentation Flags:  %010  Do Not Fragment   Last Fragment
  Fragment Offset:      0  (0  bytes)
  Time To Live:         60
  Protocol:             6  TCP - Transmission Control Protocol
  Header Checksum:      0xDFEE
  Source IP Address:    192.168.0.3
  Dest. IP Address:     192.168.0.31
  No IP Options
TCP - Transport Control Protocol
  Source Port:          20000  sms_gateway
  Destination Port:     1090  ff-fms
  Sequence Number:      1524200407
  Ack Number:           55606
  Offset:               5  (20  bytes)
  Reserved:             %000000
  Code:                 %011000    Ack    Push
  Window:               33580
  Checksum:             0xAADF
  Urgent Pointer:       0
  No TCP Options
SMSgateway  spec. 2.6.2 - decoder v.0.8
DeviceIO: message_2
  len:                  1
  value:                6
Extra bytes (Padding):
  ...               00 00 00
FCS - Frame Check Sequence
  FCS (Calculated):     0xBDB648A1
```

```
Packet #6
  Flags:        0x00
  Status:       0x01
  Packet Length:64
  Timestamp:    18:04:12.640961 10/09/2001
Ethernet Header
  Destination:          08:00:2B:C5:7E:2A
  Source:               00:C0:F0:3D:7F:9D
  Protocol Type:        0x0800  IP
IP Header - Internet Protocol Datagram
  Version:              4
  Header Length:        5  (20  bytes)
  Type of Service:      %00000000
  Precedence: Routine,   Normal Delay,   Normal Throughput,   Normal Reliability
  Total Length:         40
  Identifier:           63855
  Fragmentation Flags:  %010  Do Not Fragment   Last Fragment
  Fragment Offset:      0  (0  bytes)
  Time To Live:         128
  Protocol:             6  TCP - Transmission Control Protocol
  Header Checksum:      0x7FED
  Source IP Address:    192.168.0.31
  Dest. IP Address:     192.168.0.3
  No IP Options
TCP - Transport Control Protocol
  Source Port:          1090  ff-fms
  Destination Port:     20000  sms_gateway
  Sequence Number:      55606
  Ack Number:           1524200410
  Offset:               5  (20  bytes)
  Reserved:             %000000
  Code:                 %010000    Ack
  Window:               8757
  Checksum:             0x11E0
  Urgent Pointer:       0
  No TCP Options
  TCP Data Area:        No more data.
Extra bytes (Padding):
  ......            00 00 00 00 00 00
FCS - Frame Check Sequence
  FCS (Calculated):     0xB161CF21
```

```
Packet #7
  Flags:        0x00
  Status:       0x01
  Packet Length:64
  Timestamp:    18:04:12.641152 10/09/2001
Ethernet Header
  Destination:          00:C0:F0:3D:7F:9D
  Source:               08:00:2B:C5:7E:2A
  Protocol Type:        0x0800  IP
IP Header - Internet Protocol Datagram
  Version:              4
  Header Length:        5  (20  bytes)
  Type of Service:      %00000000
  Precedence: Routine,   Normal Delay,   Normal Throughput,   Normal Reliability
  Total Length:         43
  Identifier:           56686
  Fragmentation Flags:  %010  Do Not Fragment   Last Fragment
  Fragment Offset:      0  (0  bytes)
  Time To Live:         60
  Protocol:             6  TCP - Transmission Control Protocol
  Header Checksum:      0xDFEB
  Source IP Address:    192.168.0.3
  Dest. IP Address:     192.168.0.31
  No IP Options
TCP - Transport Control Protocol
  Source Port:          20000  sms_gateway
  Destination Port:     1090  ff-fms
  Sequence Number:      1524200410
  Ack Number:           55606
  Offset:               5  (20  bytes)
  Reserved:             %000000
  Code:                 %011000    Ack    Push
  Window:               33580
  Checksum:             0xB0DC
  Urgent Pointer:       0
  No TCP Options
SMSgateway  spec. 2.6.2 - decoder v.0.8
DeviceIO: message_3
  len:                  1
  value:                0
Extra bytes (Padding):
  ...             00 00 00
FCS - Frame Check Sequence
  FCS (Calculated):     0x77CEA603
```

```
Packet #14
  Flags:        0x00
  Status:       0x01
  Packet Length:64
  Timestamp:    18:04:12.841621 10/09/2001
Ethernet Header
  Destination:          08:00:2B:C5:7E:2A
  Source:               00:C0:F0:3D:7F:9D
  Protocol Type:        0x0800  IP
IP Header - Internet Protocol Datagram
  Version:              4
  Header Length:        5  (20  bytes)
  Type of Service:      %00000000
  Precedence: Routine,   Normal Delay,   Normal Throughput,   Normal Reliability
  Total Length:         40
  Identifier:           65135
  Fragmentation Flags:  %010  Do Not Fragment   Last Fragment
  Fragment Offset:      0  (0  bytes)
  Time To Live:         128
  Protocol:             6  TCP - Transmission Control Protocol
  Header Checksum:      0x7AED
  Source IP Address:    192.168.0.31
  Dest. IP Address:     192.168.0.3
  No IP Options
TCP - Transport Control Protocol
  Source Port:          1090  ff-fms
  Destination Port:     20000  sms_gateway
  Sequence Number:      55606
  Ack Number:           1524200413
  Offset:               5  (20  bytes)
  Reserved:             %000000
  Code:                 %010000    Ack
  Window:               8754
  Checksum:             0x11E0
  Urgent Pointer:       0
  No TCP Options
  TCP Data Area:        No more data.
Extra bytes (Padding):
  ......          00 00 00 00 00 00
FCS - Frame Check Sequence
  FCS (Calculated):     0x22659BEF
```

8.2 Example of command (52)

Raw data:

```
0000: 08 00 2B C5 7E 2A 00 C0 F0 3D 7F 9D 08 00 45 00  ..+.~*..ð=....E.
0016: 00 78 29 70 40 00 80 06 4F 9D C0 A8 00 1F C0 A8  .x)p@...O.......
0032: 00 03 04 42 4E 20 00 00 D9 78 5A D9 74 24 50 18  ...BN ...xZ.t$P.
0048: 21 EB 7B 6B 00 00 00 4E 20 30 30 30 30 30 30 30  !.{k...N 00000000
0064: 32 30 31 30 30 30 30 31 30 30 30 32 30 30 32 35  20100010002002 57
0080: 32 20 30 31 31 20 30 39 4E 32 30 30 31 31 30 30  2 011 09N2001100
0096: 39 32 20 30 31 31 30 30 39 55 30 30 30 30 30 30  92 011009U000000
0112: 30 30 30 31 20 30 35 32 31 32 33 34 35 36 37 38  0001 052123456 78
0128: 39 30 20 20 20 20 00 00 00 00                    90    ....
```

Whole network packet, interpreted:

—— END OF DOCUMENT ——