NAGRAVISION®
DIGITAL PAY-TV SYSTEMS

Nagravision S.A.

SMS Gateway interface definition

V 02.06.02

IN CONFIDENCE TO Recipient's name

CH - 1033 Cheseaux, Switzerland. +41.21.732.03.11 +41.21.732.03.00

CH - 1033 Cheseaux, Switzerland. +41.21.732.03.11 +41.21.732.03.00

CH - 1033 Cheseaux, Switzerland. +41.21.732.03.11 +41.21.732.03.00

1. Introduction

1.1 Summary

This document describes the work of the SMS gateway and in particular its role as:

- the interface in between the SMS and the SAS,
- the interface in between the SMS and the Call Collector,
- the interface in between the SMS and IMS.

1.2 Document History

See SasGwySpe020600.hst.doc

1.3 Trademarks

All company and product names used herein may be the trademarks or registered trademarks of their respective companies.

1.4 Acronyms and Definitions

Access right	Entitlement to watch a product.
ANI	**A**utomatic **N**umber **I**dentification, (Also known as caller id).
CA	**C**onditional **A**ccess— system.
CC	**C**all **C**ollector or the system, which runs on the same computer as the SAS and the SMS-Gateway is used to upload information from the ICC into the CC database, e.g., a list of expired impulsively purchased PPV event products. The uploaded information is subsequently transmitted through the SMS-Gateway to the SMS.
Cryptoperiod	The amount of time between CW changes.
CW	**C**ontrol **W**ord, used to scramble the MPEG data stream and unscramble the stream in the STB. CWs are changed periodically.
Device IO	Nagravision proprietary protocol based on TCP/IP
ECE	**EC**M **E**ncryptor. A UNIX machine that takes ECM data from the IMS and creates a properly formatted ECM stream for transmission to the demultiplexer.
ECM	**E**ntitlement **C**ontrol **M**essage. This encrypted message contains the CW and right descriptor for services.
EMB	**EM**M **B**roadcaster. A UNIX machine that takes EMM data from the SAS and creates properly formatted EMM packets for transmission to the multiplexer.
EME	**EM**M **E**ncryptor. A UNIX machine that takes EMM data and prepares encrypted messages. This encryption unit makes also decryption of messages coming from STB through modem lines.
EMM	**E**ntitlement **M**anagement **M**essage. Carries data from the system to one or many smart cards.
EPG	**E**lectronic **P**rogram **G**uide.
Event	Time slice of a service.
Free access	A broadcast mode that transmits a signal to make material available to all bearers of a valid ICC.
ICC	Integrated Circuit Card (I.e., a smart card).
IEMB	**I**mpulse **EM**M **B**roadcaster. A UNIX machine that takes IEMM data from the SAS and creates properly formatted IEMM packets for transmission to the multiplexer.
IEMM	**I**mpulsive **EMM** (Carries the data related to products that may be impulsively purchased).
IMS	**I**nformation **M**anagement **S**ystem. The computer that maintains the product and event databases and provides ECM data to the ECE.
IPPV	A PPV event product, which can be impulsively purchased through the STB and results in the acquisition of an IEMM acquired for processing in the ICC.
MOP	**M**anagement **OP**erator. The commercial program provider who manages subscribers and subscriptions.
NVOD	**N**ear **V**ideo **O**n **D**emand.
PDL	**P**arameter **D**efinition **L**anguage. A macro language used for building messages.
PPV	**P**ay **P**er **V**iew.

Preview Time	The amount of time at the beginning of an event during which the event is broadcast in free access mode.
Product	A single or group of services or events that may be purchased as a single entity.
UA	**U**nique **A**ddress for a smart card. This address identifies a particular smart card in the field.
UI	**U**ser **I**nterface.
SAS	**S**ubscriber **A**uthorization **S**ystem or the system in charge of sending the commands to the subscriber smart-card (ICC). ICC commands are sent by way of EMM and IEMM streams.
SCC	**S**ystem **C**ontrol **C**omputer used to configure the MPEG program encoders and multiplexers.
Service	TV Channel such as CNN or TNT
SMS	**S**ubscriber **M**anagement **S**ystem
Special PPV Event	An event impulsively purchased which will initiate an automatic call back once it has been watched. This event among others will be feed back to the SMS, enabling it to estimate the audience.
STB	Set-top box
STU	See STB
Watched Criterion	The number of cryptoperiods that have to be processed before an event is considered as watched.

1.5 IDs

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 a 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 for 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. This identifier is determined at the time of product creation (Command 300). It is the only product identifier known to the IMS. When a product is created, the SMS_product_ID must be given, it is returned along with the newly created product_ID to enable mapping between both IDs to be performed by the SMS.
MOP_PPID	The identifier of the management operator. This identifier is provided at system configuration time by: Nagravision.
NASP_SID	A service identifier used for conditional access.
network_id	Corresponds to the network_id as described in the DVB Network Information Table (NIT).
pattern_ID	An identifier for blackout patterns (a blackout category and subcategory definition). This identifier is entered at the time of blackout pattern definition

ID	Definition
	through the IMS interface.
PPV_number	A PPV identifier provided by the SMS. It is entered in the system through the SMS gateway commands: 300, 302, 307 and 308.
right_ID	The identifier of an entitlement. It is generated by the IMS at the time of product creation.
Service_ID	An identifier for a service. This is the identifier that users of the service will know (the virtual channel number). It may change over time. It is entered in the system through the IMS editor at the time when services are defined. It corresponds to the service_id as described in the DVB Service Description Table (SDT). It may also correspond to the program number found in the MPEG Program Map Table (PMT).
service_UID	An identifier for a service. This identifier must not change. It always represents the same service in the IMS. This is not a number that users will see. It is entered in the system at the time when the system is configured. A mapping between service_ID and service_UID must be managed by the SMS.
SMS_event_ID	An event identifier provided by the SMS..
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.
transport_id	Corresponds to the network_id as described in the DVB Network Information Table (NIT).

CH - 1033 Cheseaux, Switzerland. +41.21.732.03.11 +41.21.732.03.00

2. Theory of Operation

2.1 Product Terminology

SMS products are composed of four types of products:

Service Product	A service is a single channel (audio plus video). A service product contains the right to view one single service. For instance, WRC is a service. Services will be initially defined on the IMS Editor, then re-keyed into the SMS.
Event Product	An event is a broadcast with a defined start and end time. An event product contains the right to view a single event. An example is a PPV event such as a single football game. Events can be ordered in advance, or can be impulse-purchased through the STB. Events are normally defined through the EPG stream, but predefinition of events that are outside the range of EPG can be done via the SMS.
Package of Service Product	Groups of services defined as a single product. A package of service products contains the rights to view all services of the package. For instance, a subscriber might buy Cinemax and HBO as a single unit. Packages of services are defined via the SMS.
Package of Event Product	Like services, events can be grouped into a single product. A package of events product contains the rights to view all events of the package. Packages of events are defined via the SMS. If some events of the package are outside the range of the EPG, they are predefined and final definition of the event occurs when the corresponding EPG feed comes.

2.2 Assumptions and Restrictions

- Service ID will be keyed in the IMS and in the SMS.
- All PPV and NVOD events are definable through the EPG data feed.
- The SMS_event_ID is unique in the IMS database.
- An event belonging to a package product and not existent in the IMS database will be created as a placeholder. The EPG data feed (when available) will finalize the event's definition.
- There will be no package containing both events and services.
- Service and event packages are not impulsively purchasable.
- Subscriptions per themes and level are not supported.
- PPV per token subscriptions are not supported.

2.3 Additional Functionality

- The SMS PRODUCT_DEF commands that create or modify products contain a parameter called "reference number" to ease a subscriber's purchase of a product by telephone. The SMS is responsible for the provision of this reference number.

3. Interface Specifications

3.1 Description

The SMS Gateway is the interface between the SMS and the CA system. It manages the data flow between the:

- SMS and the SAS: grant/remove entitlements, initialise subscriber, etc.
- SMS and the Call Collector: configure subscriber profile, get the purchase history, etc.
- Call Collector and the SMS: report purchase information, etc.
- SMS and the IMS (depending on the product management, create product, change product, etc.).

The SMS Gateway interface consists of text based commands transmitted over device IO, a proprietary protocol that is layered above TCP/IP. The SMS Gateway commands are grouped in 4 different categories:

- Subscriber related commands: add, cancel and suspend subscription. (All handled by the SAS.)
- Call Collector related commands: add a subscriber to the Call Collector database, request list of purchase, etc. (All handled by the Call Collector).
- Feedback commands: these provide the SMS with different kinds of information. Which also include message acknowledgements, a complete list of purchases from the Call Collector or IMS errors depending on the system configuration.
- Product management commands, depending on the system configuration.

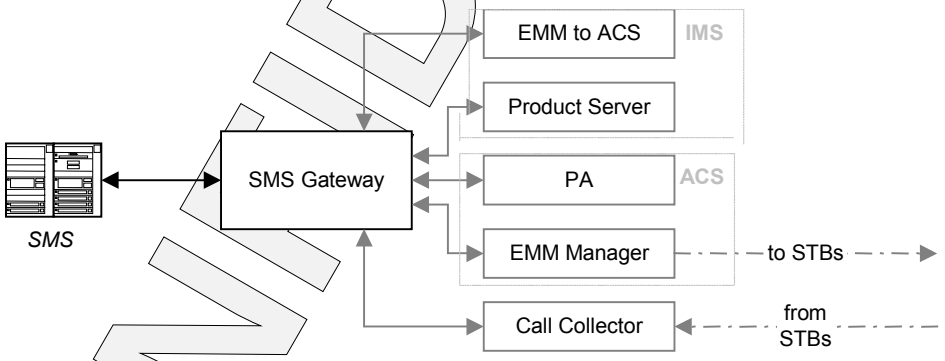


Figure 3.1

Figure 3.1 depicts the major data flows.
Note 1: Depending on the system configuration two more IMS processes could be present: “3nn” for product management and “SMS router” for feedback management. (Although, these processes are tending to be obsolete today.).
Note 2: SMS gateway currently uses GMT time as time reference.

3.1.1 SMS-SMS Gateway connections

There are 2 connections between one SMS and the SMS Gateway: one dedicated to commands from the SMS to the SMS Gateway (command numbers 0 to 199), the other dedicated to feedback commands from the SMS Gateway to the SMS (command numbers 200 to 299). Commands 1000 and 1001 are used as acknowledge messages and are exchanged in the opposite direction at each connection. That is, commands from the SMS to the SMS Gateway are acknowledged by the SMS Gateway using command 1000 or 1001 (depending on successful command completion) on the same connection the command from the SMS was received. The

1000 and 1001 commands are sent in an asynchronous way. The same applies to the SMS that must acknowledge feedback commands on the same connection it receives them on. See § 9.3 for usage of command 1002.

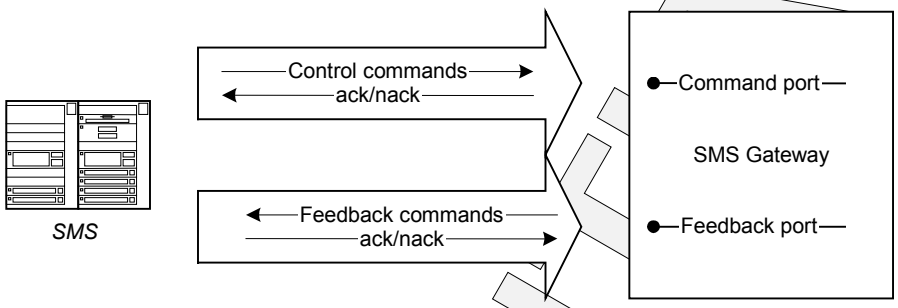


Figure 3.2

These connections occur on two different TCP ports, however, the SMS initiates both connections, regardless of the main data flow.

3.1.1.1 Connection establishment

Both connections are established in the same manner. Noting that, in order to route SMS Gateway commands in a multi-SMS environment, at the end of the **connection establishment sequence** an additional mandatory step is required—defined in § 10 (Note: The syntax of all device IO messages are already defined).

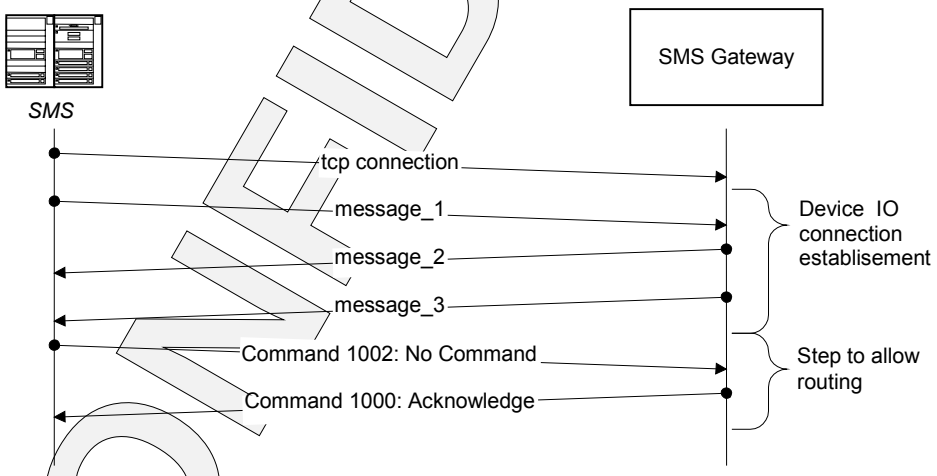


Figure 3.3

The step consists of the exchange of a command 1002: No Command. This step is necessary for the SMS Gateway to associate a source-id with the connection. This is mostly important for the feedback connection.

The SMS gateway still supports a single SMS connection.

Here are the specifications for a multiple SMS connection versus a single SMS connection:

Multiple SMS connections	Single SMS connection
a SMS must connect using the relating connection protocol described above. The SMS gateway routing table is then updated.	a single SMS simply connects device IO without using the multi SMS connection protocol.
a SMS must send a #104 command with the correct SMS Id, this value will then be kept in the CC record.	
if a SMS is disconnected, the SMS gateway routing table is then of no use, the related feedback commands will be logged.	if the SMS is disconnected, the related feedback commands are held in the CC.

CH - 1033 Cheseaux, Switzerland.
+41.21.732.03.11
+41.21.732.03.00

3.1.1.2 Command syntax

If wanted, the SMS may send packets of commands to the SMS Gateway, provided the protocol and device IO layer are respected.

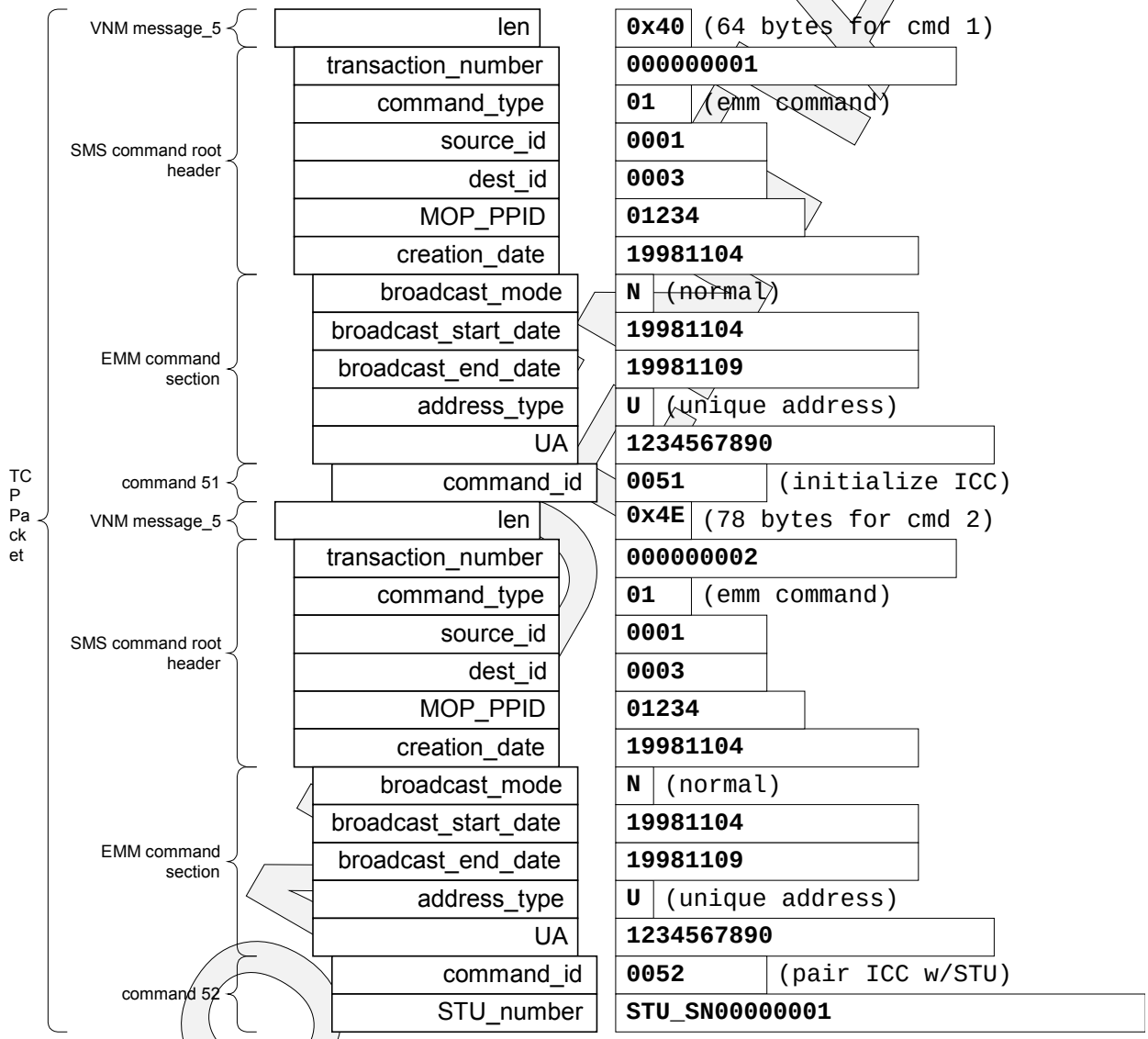


Figure 3.4

Apart from correctly formatting the command, this includes adding the 2-bytes len field at the top of every command sent, as shown in Figure 3.4.

3.2 SMS Gateway Commands

There are five types of SMS commands: EMM, CONTROL, PRODUCT_DEF, FEEDBACK, and OPERATION.

EMM, CONTROL, and PRODUCT_DEF commands are generated by the SMS and sent to the SMS Gateway. FEEDBACK commands are generated by the IMS, SAS or CC and sent to the SMS Gateway. OPERATION commands can be generated on either side.

The syntax of a SMS command is as follows:

SMS command := SMS command root header

SMS command section

SMS command root header

Field	Size	Format	Description
transaction_number	9	000000000 to 999999999	Number used to uniquely identify a transaction across the interface for each source.
command_type	2	00 to 99	Command type can be EMM, CONTROL, PRODUCT_DEF, FEEDBACK or OPERATION.
source_ID	4	0000 to 9999	A number that identifies a source such as the SMS or IMS (this number is provided by Nagravision)
dest_ID	4	0000 to 9999	Identifier of the addressed SMS. This number is defined at system configuration.
MOP_PPID	5	00000 to 99999	Identifier of the technical management operator. This number is provided at system configuration by: Nagravision.
creation_date	8	YYYYMMDD	Creation date of the command.

SMS command section

```
if ( command type == EMM command )
{
   EMM command section
}
elsif ( command type == CONTROL command )
{
   CONTROL command section
}
elsif ( command type == PRODUCT_DEF  command )
{
    PRODUCT_DEF command section
}
elsif ( command type == FEEDBACK command )
{
    FEEDBACK command section
}
elsif( command type == OPERATION command )
{
    OPERATION command section
}
else
{
    return error
}
```

- EMM commands are always addressed to a unique ICC or a set of ICCs.

- CONTROL commands are sent to the SAS or to the Call Collector (CC) requesting a special operation (**Note:** CONTROL commands do not affect ICCs).
- PRODUCT_DEF commands are used to send products definitions, updates and cancellations from the SMS to the IMS database.
- FEEDBACK commands provide information pertaining to a particular ICC status.
- OPERATION commands provide gateway transaction synchronization and handshaking.

NOTE:

- SMS commands are completely defined by the format given in this section for the root header.
- Command headers' format is given in the next sections and by the parameter data structures defined for each individual command (See § 5 and above).
- Terminators are not used either for records or individual fields.
- Fields are all of fixed length.
- The SMS Gateway ID (used as the value for the above dest_id and source_id fields) is defined as 0002.
- The CC ID (used as the value for the above dest_id and source_id fields) is defined as 0003 for one CC configuration.

3.2.1 EMM Command Section

EMM command section of SMS commands contains parameters used to send commands of type EMM to the ICC. The syntax is as follows:

EMM command section := EMM command header
EMM command parameters

EMM command header

Field	Size	Format	Description
broadcast_mode	1	N or B	Normal Standard broadcasting mode. Batch Command with a lower priority
broadcast_start_date	8	YYYYMMDD	Broadcast start date. The command must be sent to the ICC from this date[1]
broadcast_end_date	8	YYYYMMDD	Broadcast end date. The command must be sent to the ICC until this date
address_type	1	U or G	EMM addressing mode for EMM commands Unique (U) - An ICC is addressed by its unique address (UA). Global (G) - All ICCs of the MOP are addressed

if (address_type == U)
{
UA
}
elsif (address_type == G)
{
void
}
else
{
return error
			UA The Unique Address of the card for which the command is intended

[1] The broadcast start and end date is defined by the SMS or by the CA system when setting up the system.

}	

See § 5 for details of EMM commands' parameters.

3.2.2 CONTROL Command Section

CONTROL command section of SMS commands contains parameters used to send commands to the Call Collector. The syntax is as follows:

CONTROL command section := CONTROL command header
CONTROL command parameters

CONTROL command header

Field	Size	Format	Description
broadcast_mode	1	N or B	Normal Standard broadcasting mode. Batch Command with a lower priority
broadcast_start_date	8	YYYYMMDD	Broadcast start date.
broadcast_end_date	8	YYYYMMDD	Broadcast end date.
address_type	1	U or G	EMM addressing mode for EMM commands Unique (U) - An ICC is addressed by its unique address (UA). Global (G) - All ICCs of the MOP are addressed

if (address_type == U)
{
UA
}
elsif (address_type == G)
{
void
}
else
{
return error
}
			UA The address of the ICC to which the command refers.

The fields of CONTROL command headers must be documented as follows:

Field	Value
broadcast_mode	N
broadcast_start_date	current date
broadcast_end_date	current_date
address_type	U
UA	0000000000 to 4294967295, the UA of the ICC to which this command refers.

See § 6 for details of the CONTROL commands' parameters.

3.2.3 PRODUCT_DEF Command Section

PRODUCT_DEF command section of SMS commands contains parameters used to send commands to the IMS. The syntax is as follows:

PRODUCT_DEF command section := PRODUCT_DEF command parameters

See § 8 for details of the PRODUCT_DEF commands' parameters.

3.2.4 FEEDBACK Commands Section

FEEDBACK command section of SMS commands contains parameters used to send commands to the SMS. The syntax is as follows:

FEEDBACK command section := FEEDBACK command header
FEEDBACK command parameters

FEEDBACK command header

Field	Size	Format	Description
UA	10	0000000000 to 4294967295	ICC unique address Note: commands 208, 209 and 210 UA = 0000000000

See § 7 for details of the FEEDBACK commands' parameters.

3.2.5 SMS OPERATION Commands

These are the commands used for sending acknowledgments or non-acknowledgments. OPERATION command section of SMS commands contains parameters used to send commands back and forth between the SMS and the SMS Gateway. The syntax is as follows:

OPERATION command section := OPERATION command parameters

See § 9 for details of the OPERATION commands' parameters.

3.2.6 SMS Commands: field size and format

All fields are in ASCII format. They are described here below:

- Numeric types are decimal, 0 filled and right justified. For example, the value 3 for a LENGTH field will coded 003
- Alpha/Num are left aligned, blank padded, capitalized. For example, the value NASPv1.0 for a ICC_version field will be coded "NASPV1.0 " (two spaces at the end)

Names	Size	Type	Format
abnormal_phone	16	Alpha/Num + trailing spaces	
address_type	1	Alpha	U=UNIQUE or G=GLOBAL
all dates	8	Num	date in format YYYYMMDD. The value Forever is 20201231.
all duration	4	Num	0000 to 9999
all lengths (except length_event_name)	3	Num	000 to 999
all times	6	Num	HHMMSS (i.e. 000000 to 235959).
blackout_subtype	3	Num	001 to 128
blackout_type	2	Num	00 to 12, where 00 means no blackout.
broadcast_mode	1	Alpha	N=NORMAL or B=BATCH
call_freq	2	Num	01=annual 02=semi-annual 03=quarterly 04=monthly 05=bimonthly 1n=every n days (note: 1 should be considered as a flag) n should be 1 to 9 or A to F (10 to 15 in Hexadecimal)
cc_number	16	Alpha/Num	call collector phone number
cc_port	5	Num	call collector TCP/IP port
command_ID	4	Num	0000 to 9999
command_type	2	Num	01=EMM command 02=CONTROL command 03=PRODUCT_DEF command 04=FEEDBACK command 05=OPERATING command
credit	7	Num	0000000 to 6553599 meaning $00000.00 to $65535.99
credit_limit	7	Num	0000000 to 6553599 meaning $00000.00 to $65535.99
credit_mode	2	Num	01=ADD 02=SUBTRACT 03=SET CREDIT 04=SET BALANCE (reset debit to zero and set credit limit) 05=SUB OFFSET (subtract offset from debit and credit)
debit	7	Num	0000000 to 6553599 meaning $00000.00 to $65535.99
description	250	Alpha/Num	
dest_ID	4	Num	0000 to 9999
error_code	4	Num	0000 to 9999
error_code_ext	4	Num	0000 to 9999
event_name	32	Alpha/Num	
event_number	3	Num	000 to 999

Names	Size	Type	Format
force_emm	1	Alpha	Y=YES or N=NO
free_preview_time	2	Num	00 to 98, 99 means inherit value from free_preview_time defined for the service.
id	4	0000 .. 9999	unique message identifier
impulse_purchase	1	Alpha	Y=YES or N=NO
impulse_purchase_allowed	1	Alpha	Y=YES or N=NO
IMS_product_ID	12	Num	000000000000 to 999999999999 (IMS product ID)
ip_address	15	Alpha/Num	000.000.000.000 to 255.255.255.255 (valid IP address)
IRD_command_id	3	Num	00 to 99
IRD_data	96	Alpha/Num	Hexadecimal
IRD_data_len	2	Num	00 to 96
IRD_operation	3	Num	00 to 99
kind	1	0 .. 9	message kind (e.g advertisement, warning etc ...), to be defined between the SMS and the STB provider 0 is the default value
length_event_name	2	Num	00 to 17
lid	3	aaa .. zzz	language code compliant with ISO639_2
message	up to 999	Alpha/Num	message for mail
mode	1	0 .. 1	display mode (0 => normal, 1 => fingerprint)
MOP_PPID	5	Num	00000 to 65535
nack_status	1	Num	1=REJECTED or 2=POSTPONED
name	80	Alpha/Num	
nb_of_products	2	Num	00 to 99
nb_of_subtypes	3	Num	000 to 128
network_ID	5	Num	00000 to 65535
number_of_IPPV	2	Num	00 to 99, the number of IPPV reported in a callback
original_transaction_number	9	Num	0000000000 to 999999999
phone_number_n	16	Alpha/Num	phone number stored in the CC (n is 1 to 3)
PPV_number	7	Num	0000000 to 9999999
price	5	Num	00000 to 99999 meaning $000.00 to $999.99
priority	1	0 .. 1	message priority (0 => low, 1 => high)
purge_mode	1	0 .. 9	0 => purge only rights marked as call collected 1 => purge only rights marked as non-call collected 2 => purge both kind of rights 3 .. 9 => future use
range	1	Alpha	Y=YES or N=NO
reference number	4	Num	0000 to 9999
responding	1	Alpha	Y=YES or N=NO
reverse_blackout_flag	1	Alpha	Y=YES or N=NO
service_ID	3	Num	000 to 999
service_ID	5	Num	00000 to 65535
service_number	3	Num	000 to 999
service_UID	5	Num	00000 to 99999
SMS_event_ID	12	Num	000000000000 to 999999999999
SMS_product_ID	12	Num	000000000000 to 999999999999
source_ID	4	Num	0000 to 9999
special_PPV	1	Alpha	Y=YES or N=NO
special_PPV_event	1	Alpha	Y=YES or N=NO
STU_number	14	Num	Serial Number of set-top box unit.

Names	Size	Type	Format
suspend_ICC	1	Alpha	Y=YES or N=NO
suspend_product	1	Alpha	Y=YES or N=NO
threshold_credit	7	Num	0000000 to 6553599 meaning $00000.00 to $65535.99
tokens	4	Num	0000 to 9999
transaction_number	9	Num	000000000 to 999999999
transport_ID	5	Num	00000 to 65535
type_of_products	1	Alpha	S or E or B which stand for Service, Event and Both
UA	10	Num	Unique address range is 0000000000 to 4294967295
version	2	00 .. 99	message version
watched_criterion	3	Num	000 to 255, 999 means inherit watched_criterion value from service watched_criterion
watched_status	1	Alpha	Y=YES or N=NO
zip_code	5	Num	00000 to 99999. Used for example for US zip code
zip_code	10	Alpha/Num	a .. z, A .. Z, 0 .. 9 and space. Used for example for UK postzip code

CH - 1033 Cheseaux, Switzerland.
+41.21.732.03.11
+41.21.732.03.00

3.3 SMS Error Codes

If an error occurs when executing a SMS command, it is reported by the use of an error code and an error code extension. Where the code indicates an error category and the extension giving more precise detail concerning the error source (e.g., an incorrectly specified field or value exceeded etc.).

3.3.1 Error codes

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

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 ICC referenced in the command has been canceled.
UA_NOT_FOUND	0008	The ICC 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 an ICC in the CC database, but the ICC is bound to an already existing STB.
SERVICE_NOT_FOUND	0011	The service referenced in the command does not exist in the IMS database.
TOO_MANY_RIGHTS	0012	The command attempts to create a package whose number of item cause the number of rights exceeding the maximum value allowed (3).
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 an ICC that already exists.
BAD_EPG_FORMAT	0015	The format of the EPG data feed is not correct.
SMS_EVENT_ID_NOT_FOUND	0016	The command references a SMS_event_ID that does not exist in the IMS database.
PRODUCT_ON_NON_PPV_EVENT	0017	The command attempts to create a product containing a non-PPV event.
EVENT_ALREADY_IPPV	0018	The command creates an impulsive event product

		with an event already used in an impulsive event product.
BLACKOUT_TYPE_OR_SUBTYPE_NOT_FOUND	0019	The blackout type or subtype referenced in the command does not exist in the IMS database.
EVENT_WITH_PPVNB_ALREADY_PROGRAMMED	0020	The command attempts to create an event product on an event with a different ppv number than the one already set.
DB_INCONSISTENT_TOO_MANY_ROWS	0021	Inconsistency in the IMS database
DB_INCONSISTENT_INVALID_PRODUCT	0022	Inconsistency in the IMS database
MUTLTIPLE_EVENTS_WITH_SAME_PPVNB_ON_IPPV	0023	There are many events with the same ppv number when modifying a ppv product to an impulsive ppv product.
PRODUCT_INCONSISTENT	0024	There is an inconsistency between the received product definition and the internal SMS Gateway product database.
TOO_MANY_ITEMS	0025	Too many items are given in the list.

3.3.2 Error codes extension

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_SMS_PRODUCT_ID_FORMAT	0009	The format of the SMS_product_ID contained in 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_SMS_EVENT_ID_FORMAT	0012	The value of a SMS_event_ID 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.
IDENTICAL_PRODUCTS	0018	The command attempts to define a product already existing in the IMS database with an identical definition.
BAD_BROADCAST_MODE	0019	The format of the value of the broadcast_mode field of the command 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.
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_VERSION_FORMAT	0026	The version number of the EPG data feed is not in the correct format.
BAD_NUMBER_FORMAT	0027	A non-numerical character was found in the content of a Num field.
BAD_FLAG_FORMAT	0028	A value different from Y or N was found in a FLAG field.
BAD_TIME_FORMAT	0029	The format of the time specified in the EPG data feed is incorrect.
BAD_RATING_FORMAT	0030	The format of the value of a 'rating' field is incorrect.
BAD_CRC_32	0031	The CRC of the EPG data feed indicates an error.
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_PPV_NUMBER_FORMAT	0035	The format of the value of the PPV_number field is incorrect.
BAD_REFERENCE_NUMBER_FORMAT	0036	The format of the value of the reference_number field of the command is incorrect.
BAD_BLACKOUT_TYPE_FORMAT	0037	The format of the value of the blackout_type field of the command is incorrect.
BAD_NB_OF_SUBTYPES_FORMAT	0038	The format of the value of the nb_of_subtypes field in the command is incorrect.
BAD_BLACKOUT_SUBTYPE_FORMAT	0039	The format of the value of a blackout_subtype field of the command is incorrect.
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_TOKEN_NUMBER_FORMAT	0042	The format of the value of the token field of the command is incorrect.
BAD_EVENT_NUMBER_FORMAT	0043	The format of the value of the event_number field of the command is not correct.

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.
BAD_DEAS_MESSAGE_FORMAT	0046	The DEAS message format is incorrect.
BAD_FIPS_FORMAT	0047	The FIPS number format is incorrect.
EXTERNAL_SYSTEM_NOT_RESPONDING	0048	The external system is not responding.
EXTERNAL_SYSTEM_ERROR	0049	The external system has not processed the command, an error has occurred during its treatment.
TOO_MANY_ROWS	0050	A query returns many rows while only one is expected
INVALID PRODUCT	0051	The referenced product is incomplete (no right or no PPV number or no event or no service).
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.

3.3.3 Command root header error codes and extensions

Among the error codes and extensions defined above, those that are applicable to report errors in the root header fields are:

Error code	Error codes extensions
BAD_ROOT_HEADER_SYNTAX	BAD_COMMAND_TYPE BAD_SOURCE_ID BAD_DEST_ID BAD_MOP_PPID BAD_DATE_FORMAT

The error code and error code extension are used in the following way:

Field	Error codes	Error codes extension
transaction_number	BAD_ROOT_HEADER_SYNTAX	NO_EXTENDED_ERROR
command_type	BAD_ROOT_HEADER_SYNTAX	BAD_COMMAND_TYPE
source_ID	BAD_ROOT_HEADER_SYNTAX	BAD_SOURCE_ID
dest_ID	BAD_ROOT_HEADER_SYNTAX	BAD_DEST_ID
MOP_PPID	BAD_ROOT_HEADER_SYNTAX	BAD_MOP_PPID
creation_date	BAD_ROOT_HEADER_SYNTAX	BAD_DATE_FORMAT

3.3.4 Command header error codes and extensions

Error code	Error codes extensions
BAD_HEADER_SYNTAX	BAD_BROADCAST_MODE BAD_ADDRESS_TYPE BAD_DATE_SEQUENCE BAD_DATE_FORMAT
UA_NOT_FOUND	NO_EXTENDED_ERROR_CODE
CANCELED_CARD	NO_EXTENDED_ERROR_CODE

The error codes and error code extensions are used in the following way to report errors in command header fields (when applicable, that is when the header contains the specified field, see § 3.3.1 and § 3.3.2:

Field	Error codes	Error codes extension
broadcast_mode	BAD_HEADER_SYNTAX	BAD_BROADCAST_MODE
broadcast_start_date	BAD_HEADER_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE
broadcast_end_date	BAD_HEADER_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE
address_type	BAD_HEADER_SYNTAX	BAD_ADDRESS_TYPE
UA	UA_NOT_FOUND CANCELED_CARD BAD_HEADER_SYNTAX	NO_EXTENDED_ERROR_CODE NO_EXTENDED_ERROR_CODE BAD_UA_FORMAT

3.4 EPG Data Feed Errors Specification

Errors found in the EPG data feed are reported to the SMS through FEEDBACK commands 208, 209, and 210.

- Command 208 is used to report overlapping events.
- Command 209 is used to report the presence of an event of duration 0 (zero) in the data feed.
- Command 210 is used to report format errors in the EPG data feed, the error codes and error code extensions used are:

Error code	Error codes extensions
BAD_EPG_FORMAT	BAD_DATE_FORMAT BAD_VERSION_FORMAT BAD_SMS_EVENT_ID_FORMAT BAD_FLAG_FORMAT BAD_TIME_FORMAT BAD_RATING_FORMAT BAD_CRC_32

4. Scenario for Use

4.1 Creating a Physical Transmission Channel

All PTCs have to be created in the IMS using the IMS editor, before creating services.

4.2 Creating a service

Services have to be created before events can be scheduled for them. Each service is entered manually with the IMS editor.

A service is always linked to a PTC that has circuits connected to it. The circuits are either entered manually with the IMS editor or attached automatically by the SCC automation interface (description of SCC automation interface to be provided by DiviCom).

4.3 Updating a service definition

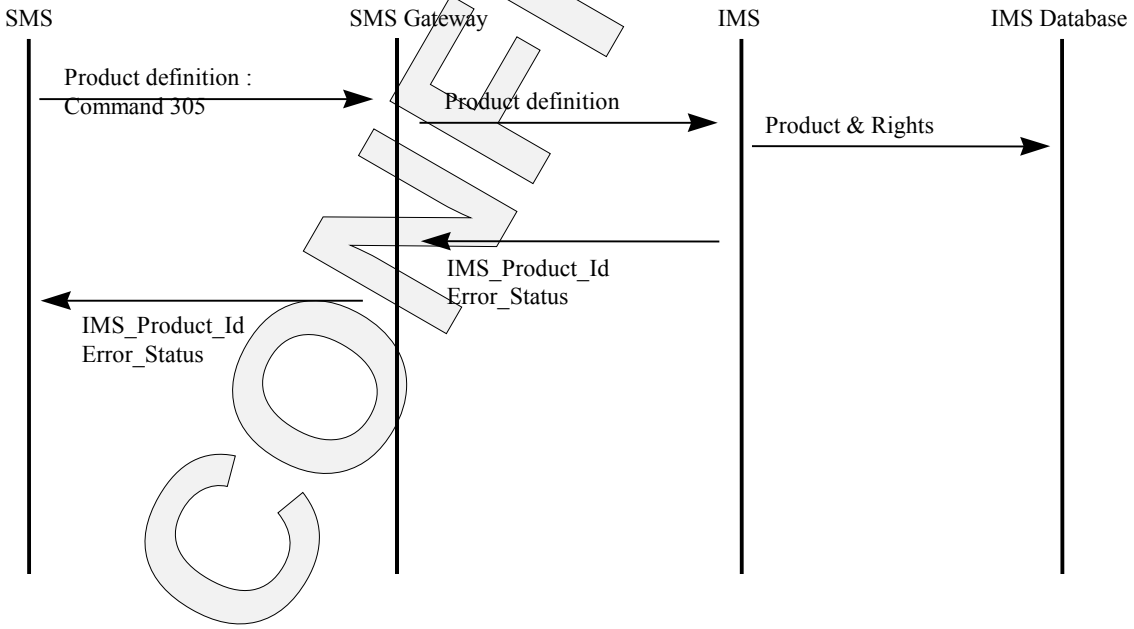
A service definition can be updated through the IMS editor interface.

4.4 Deleting a service

A service is deleted with the IMS editor. A service cannot be deleted when it has events, products or circuits attached. The service should be deleted on the SMS as well.

4.5 Creating a service package product

Here is a pictorial flow diagram



A package is identified by an IMS_product_ID, and contains a list of services.

A command (Command 305) is sent to the SMS gateway to create the product in the IMS database.

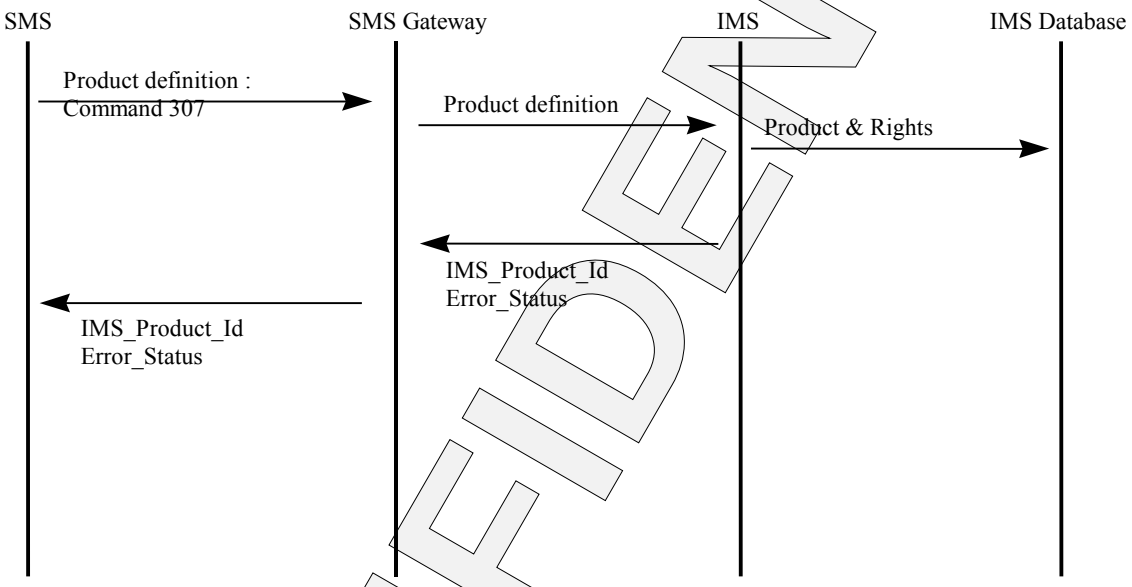
If a product with the same reference number is found in the IMS database, the two products are compared. The command is NACKed and the message returned reflects whether the products were the same or not.

If one service of the list is not found in the IMS database, the command is NACKed.

If there are too many items in the package (the package requires more than a given predefined number (3) of rights on the ICC), the command is NACKed, returning the maximum number of items allowed. The maximum number of rights is a parameter set through the PDL interface.

If the product does not exist and all services have been found in the IMS database, the product is created and the command is ACKnowledged

4.6 Creating an event package product



A package is identified by an IMS_product_ID, and contains a list of events.

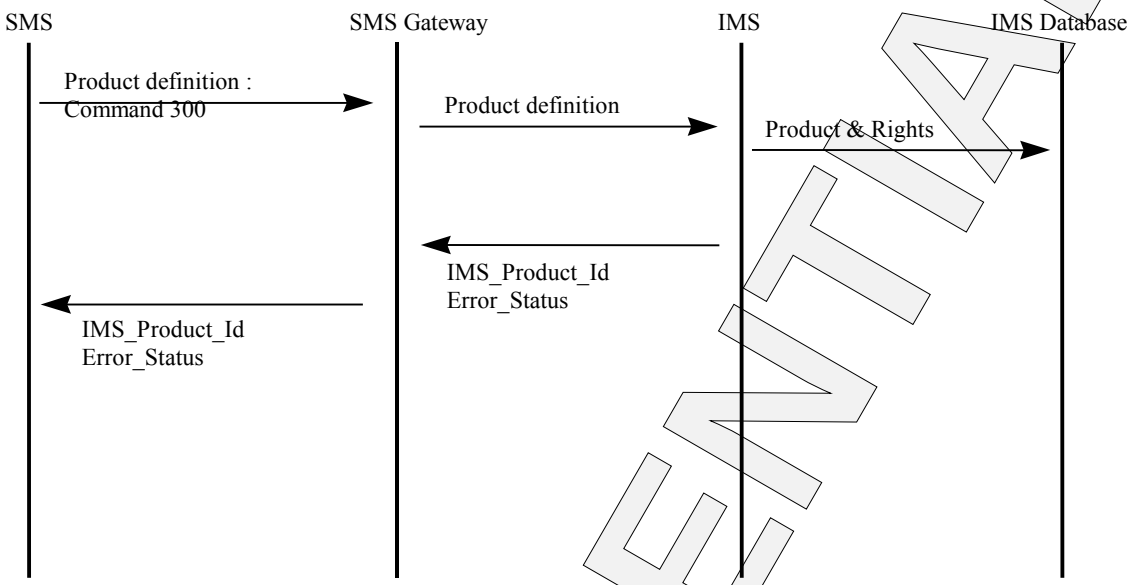
A command (Command 307) is sent to the SMS gateway to create the product in the IMS database.

If a product with the same reference number already exists, the two products are compared. The command is NACKed and the message returned reflects whether the products were the same or not.

If there are too many items in the package (the package requires more than a given predefined number of rights on the ICC), the command is NACKed, returning the maximum number of items allowed. The maximum number of rights is a parameter set through the PDL interface.

If the product does not exist it is created and the command is ACKnowledged.

4.7 Creating an event product



Issuing Command 300 creates an event product. It contains a PPV number provided by the SMS and the SMS_event_ID (also provided by the SMS) identifies it. This event will be impulsively purchasable unless specified otherwise in the Command 300 parameters.

If an event with the same SMS_event_ID does not exist in the IMS database, a placeholder for the event is created. The placeholder will be subsequently replaced by the actual event definition from the EPG data feed.

4.8 Updating an event

An event may be updated automatically by the EPG event feed. It may also be updated manually with the IMS editor in case of emergency. All fields may be modified, as long as these integrity rules are not violated:

- The SMS_event_ID must be unique.
- Date and time cannot overlap with another event.
- The service carrying the event must be defined in the IMS database.

4.9 Updating the schedule of an event product

If the End_date of the event product changes, then Command 64 (Update event right) is issued automatically. This command will update the rights in the ICCs that contained the rights for the original product. This mechanism is applied for updates coming from the EPG data feed or from the IMS editor.

Command 302 (Modify event product) should be issued by the SMS to update the validity dates of the event product.

4.10 Updating the schedule of an event in an event package product

If the End_date of an event changes to a date beyond the validity_end_date of the event package product, then Command 64 (Update event right) is issued automatically. This command will update the rights in the ICCs that contained the rights for the original product. This mechanism is applied for updates coming from the EPG data feed or from the IMS editor.

Command 308 (Modify event package product) should be issued by the SMS to update the validity dates of the event package product.

4.11 Deleting an event

Events are automatically deleted from the IMS database a certain number of days after their expiration. This number of days is a parameter of the IMS entered through interface of the IMS editor. Events can also be removed manually with the IMS editor.

4.12 Subscriber creation

Creating a subscriber in the system means completing the following process:

- Initialization of the subscriber's ICC.
- Pairing of the ICC with the STB.
- Entering subscriber information in the system.

The steps have to be done in sequence. Note that the first and the third one should be done as closely as possible as it does not make sense to have initialized ICCs that are not associated with any subscriber, nor does it make sense to have an information on a subscriber who does not have a card.

4.12.1 ICC Initialization

The initialization of an ICC is performed with the SMS Command 51. The information available to the SMS must be the Unique Address of the ICC to initialize. An ICC can be initialized several times without altering the content of the ICC.

4.12.2 ICC/STB Pairing

To be able to unscramble, a STB must be paired with a given ICC. The pairing process will (1) enable the ICC and the STB to operate together, and (2) will bind the ICC with the STB so that the ICC can only be used with the STB it has been paired with. Unless pairing data is overwritten on the ICC, the ICC is permanently paired to the STB.

In order to pair an ICC with an STB, the following information must be available at the SMS:

- the Unique Address of the card
- the serial number of the STB

Then Command 52 must be issued.

In almost all cases, pairing of the STB with an ICC will take place at the time of subscriber creation. However, in the case where a subscriber's ICC or STB cannot be operated any longer, pairing may have to be performed again.

Specifically, if a STB ceased to operate and were replaced, the subscriber could still use the same ICC with the new STB, but the STB having changed, the ICC would have to be repaired to enable unscrambling with the new STB. In such a case, only the pairing information on the ICC has to be updated, the ICC will afterwards operate just as with the previous STB.

If a subscriber lost his/her ICC but kept the same STB, then a new card will be issued, it will have to be initialized and paired.

4.12.3 Entering subscriber information

At the time a subscriber is created in the system, the following information must be available to the SMS:

- The UA (Unique address = ID) of his/her ICC (this information is printed on the ICC)

- The ZIP code
- Phone number(s) (unnecessary if ANI not supported)
- Values for initial credit and threshold
- The phone number and/or the network address of the call collector the ICC will report to
- The date of his first callback
- The period between two callbacks

The subscriber creation process includes the following operations (in sequence):

- Command 104 Add ICC to Call Collector database in order for callbacks to be processed.
- Command 48 Send EMM to ICC with Zip Code. The ICC must know its location Zip Code since time information and possibly blackout information is based on the card's location.
- Command 13 Send EMM to ICC with credit and threshold values. Create Credit for Impulse Purchase (optional), to enable impulsive purchases.
- Command 100 Update Call Collector database with credit limit for the ICC.
- Command 101 Update Call Collector database with up to 3 phone numbers from which this STB is allowed to call from (optional).
- Command 49 Send EMM to ICC with callback phone number or callback IP address.

or Command 54

- Command 61 Enable Automatic Call Back (optional). This command includes call time calculation by the Call Collector.

4.13 Callback

When does callback occur?
In general, the occurrence of callback depends on the type of circumstances generating the callback. A callback belongs to one of three categories: automatic, on command, and event based. In typical cases, callbacks will take approximately 1 minute to complete full data transfer from the ICC to the CC.

Automatic - Automatic callbacks are strategically scheduled to occur during the night to minimize phone rates and phone line contention. These are setup at the Call Collector and may be scheduled by ICC UA.

On-Command - *On*-Command calls are generated for various reasons at the head-end site dependent on necessary conditions defined by the billing center. These callbacks will be performed immediately upon reception of the EMM generated by the SMS command 60. Callback data to the call collector will typically be received within 10 minutes after issuing the SMS command.

Event based - Event based callbacks occur as soon as one or more of the following conditions are satisfied:

- *Threshold Limit* - The available credit falls below the threshold limit stored in the ICC. During this callback, expired PPV events (whose end times are prior to the actual time) will be reported. Credit is restored to the credit limit.
- *Memory Full* - The ICC memory is full. Upon callback, the call collector will collect expired PPV events and send a reclaim memory command to the ICC. Credit is restored to the credit limit.
- *Special Event* - A special event is an event defined as such by the SMS. When an event is defined as special, a callback will be triggered at the end of the event if the event has been watched (as defined by the watched flag in the smart card) and purchased impulsively.

The following steps occur during the callback process (the order of the various steps may change):

- verify phone number if ANI enabled.
- get current debit and credit.
- get list of expired impulsively purchased event products and not call collected already.
- reset credit in the ICC.
- cleanup of rights expired for more than 30 days.
- Set up the new callback date and time in the ICC

4.14 ICC - Lost/Stolen/Deleted

An ICC cancellation is performed each time a card is removed from a customer account because it has been stolen lost or is defective. In addition the ICC is flagged as canceled and deactivated in the SAS database.

The operations necessary to cancel an ICC are the following (in sequence):

- Command 50 Cancel ICC.
- Command 105 Cancel ICC On Call Collector.

Although the ICC has been canceled, all expired IPPVs may still be retrieved (special equipment needed) if the ICC is physically returned to the SMS. The ICC may be reused only after being reinitialized by Nagravision.

4.15 Grant the subscriber the rights to a product

First the command 2, Add product, must be issued to grant the subscriber the rights to access the product. The command 3, Product renewal, allows extending the product end date.

4.16 Subscriber suspension

Rights stored on an ICC can be temporarily suspended in three different ways:

- Command 4: Product suspension. This command allows the SMS to suspend rights for an individual product.
- Command 14: Suspend Impulse Purchase. This command is used to suspend the right to perform impulsive purchases of PPV event products. The subscriber may still call the SMS to order event products.
- Command 20: Suspend subscriber ICC. This command temporarily disables all rights, including the right to perform impulsive purchases of PPV event products.

Rights or ICC suspension do not prevent callbacks from occurring. In addition a suspended right or ICC can be restored to its original state with respectively Command 5, Command 15 and Command 21.

4.17 ICC Credit management

A new ICC does not contain any credit. During initialization, a command 13 “Create Credit for Impulse Purchase” is sent to the card. This creates a record in the ICC containing the amount specified in the command’s parameter. It defines also the amount of money under which a callback is triggered (threshold). The ICC is now enabled to purchase products impulsively.

A command 100 is also sent at initialization time that defines the credit limit for this subscriber in the call collector database.

4.17.1 Credit Allowance

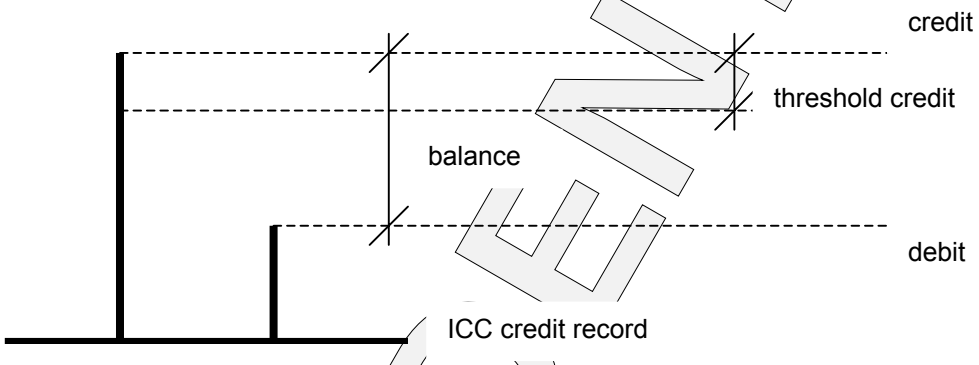
Credit is provided to customers through the entitlement mechanism. Credit levels may be issued on an individual subscriber basis or on a system wide basis. To ensure that sufficient credit is always available on the ICC, a credit threshold is defined to trigger callback.

The ICC stores a credit value and a debit value. The available credit is the difference of both values. When a purchase is performed, instead of subtracting the price from the credit, the price is added to the debit. When the credit is restored by the CC, the CC will add a value to the credit such that the difference between the credit and the debit (balance) equals the credit limit.

The outcomes and actions are given below:

Condition	Action
Purchase Price > Balance	Purchase Refused
Purchase Price < Balance and (Balance - Purchase Price) < Threshold	Purchase Allowed, Callback triggered
Purchase Price < Balance	Purchase Allowed

When a callback is triggered the credit limit is restored and the expired PPVs are reported. To avoid overflow of the debit and credit counters (maximum of $65,536.99), the SMS should issue Command 8 to set the debit counter to 0 and the credit counter to the credit limit.



In summary, the ICC credit record is created when the command 13, Create credit for impulse purchase, is issued, the credit as well as the threshold credit will then be defined. The command 8, Credit management, and the command 9, Update threshold credit, can be used to update these values, the only restriction is the ICC credit record can be updated only once per day (GMT day).

The command 100, Redefine credit limit, will update the Call Collector related UA record credit limit field. There is no restriction for this command, this is the command to use for credit tuning. The new ICC credit value after a callback is the current debit plus the Call Collector credit limit.

4.18 Deleting a subscriber

To delete a subscriber from the system, the following commands must be issued:

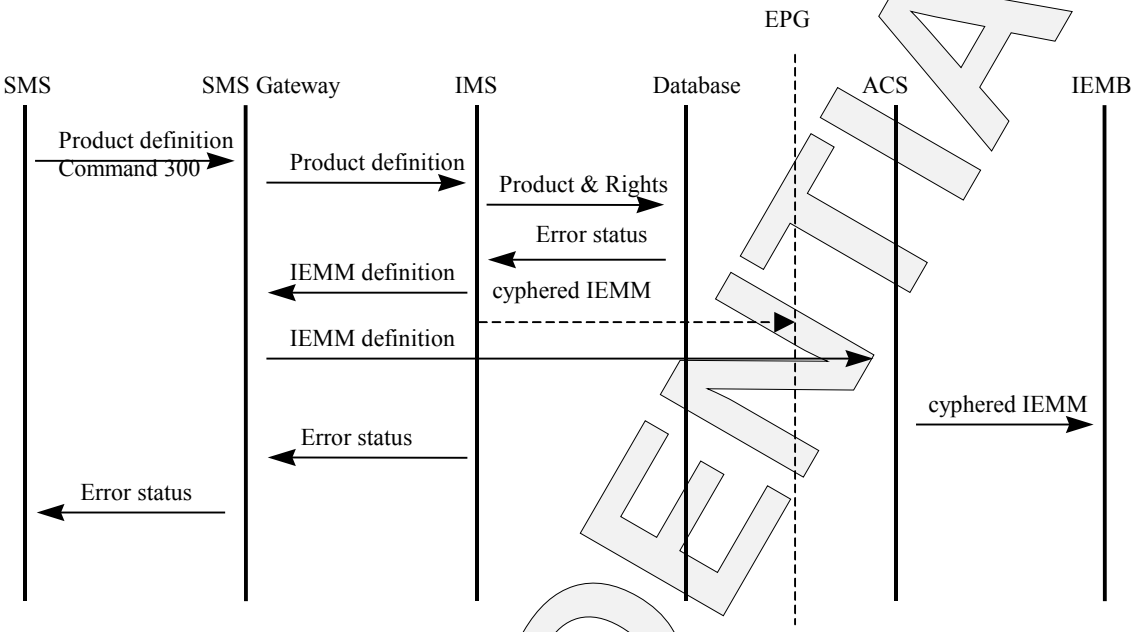
- Command 50, Cancel ICC, to flag the ICC as canceled in the SAS database.
- Command 105 to remove the ICC from the CC database.

All expired IPPVs may still be retrieved (special equipment needed) if the ICC is physically returned to the SMS. The ICC may be reused only after being reinitialized by Nagravision.

4.19 Event Product Purchase

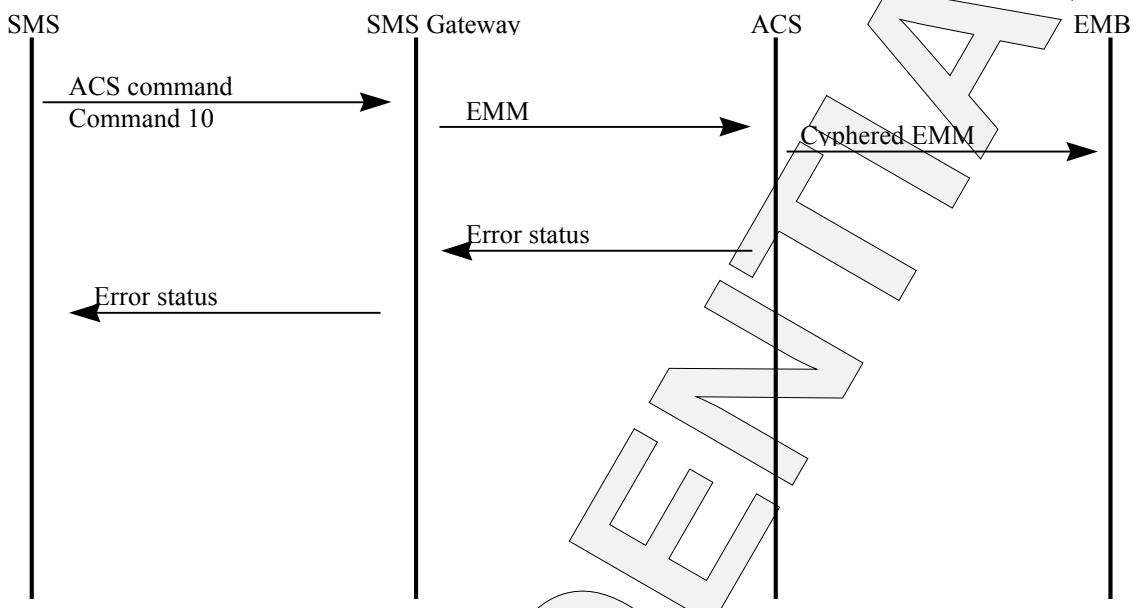
To create an impulsively purchasable event, command 300 must be executed for the event with the impulse_purchase_allowed field set to Y.

4.19.1 Impulse purchase (at subscriber site using remote control of STB)



When an event product has been created as impulsively purchasable, a subscriber may purchase the event through the STB. The IEMM is broadcast either through the EPG (DVB SI) or through the IEMM data stream generated by the IEMB (NB. The former is the preferred method). Once the purchase is confirmed, the rights for the event product are stored in the ICC, and the price of the event product is added to the debit. Such purchases are operations local to the STB and do not cause any message to be sent to the SMS. Expired event products will be reported to the SMS through callbacks (the structure of the data reported is described in § 7). An event product expires at midnight of the end date.

4.19.2 PPV purchase (through phone call to SMS)



Calling the SMS may always purchase a PPV event product. The SMS will then send a Command 10 (Create PPV) through the SMS. This command will be translated into a regular EMM and sent to the subscriber's ICC. There is no modification of the debit and credit amounts contained in the ICC, because the purchase is already recorded at the SMS.

Event products purchased through the SMS may be canceled by calling the SMS. Event products purchased through the SMS are not reported in callbacks.

4.20 Blackout

For a number of reasons (such as agreements with service providers), certain events or services should be denied, or blacked out, in particular areas. This means that subscribers living in the blacked out areas would not be allowed to view the blacked out event or service.

A blackout capability is supported in the conditional access system. To make use of this capability, three general functions must be performed:

1. Blackout types and subtypes must be defined.
2. ICCs must be associated with blackout types and subtypes.
3. Events or services must be associated with blackout types and subtypes.

When these functions are completed, ICCs associated with any particular blackout type and subtype would not be allowed to view any event or service assigned the same blackout values. For example, any viewer assigned to blackout type 7 and subtype 23 would not be able to view any event also assigned blackout type 7 and subtype 23.

4.20.1 Blackout Type and Subtype definition

The first step in making use of the blackout feature is to define blackout types and subtypes. Up to 12 blackout types are supported. These types can be associated to a category of event such as NFL Football or boxing. Blackout types are defined through the IMS editor. Their assigned number solely represents blackout types. Blackout types and subtypes can be updated through the IMS editor. Blackout subtype modifications,

performed at the IMS editor, involve updating the blackout subtype information on all ICCs and therefore, this is a lengthy process.

For each of the 12 blackout types, up to 128 subtypes can be defined. Each of these subtypes defines a "geographic" area by associating each of the sub-types with a group of zip codes.

For example,

1. Blackout type 1 is based on state; an event must be blacked out in Texas. In this case 50 subtypes must be defined, one for each state. Representing each state with its number in the alphabetical order, subtype 1 represents Alabama, ...
2. Blackout type 12 is NHL games blackout. In this type 26 subtypes must be defined, one for each NHL stadium location (assuming that blackouts have to be enforced in the area surrounding each stadium). Subtype 1 represents Anaheim, subtype 2 represents Boston, etc

In summary, up to 12 blackout types can be defined. For each type, up to 128 subtypes can be defined. Black out geographic areas are then defined by associating zip codes or state number to each blackout subtype. Zip codes give the finest geographic granularity for blackout areas.

4.20.2 Assigning blackout type and subtype to an ICC

The second step in making use of the blackout capability is to ensure that each ICC is assigned the proper blackout subtype for each of the blackout types. Periodically (at least once a month) the IMS generates and sends EMM to all ICCs to update their blackout information according to the blackout types and subtypes definitions. Should the zip code information be updated on an ICC, the blackout information on the ICC would be updated accordingly within 72 hours.

If a new zip code is created by the postal service, it should be input into the IMS at least two month before it can be used.

4.20.3 Event or service blackout type and subtype assignment

The third and final step in making use of the blackout capability is to assign a predefined blackout type and subtype to an event or a service. For instance, in the examples, an NHL game will be given the type 12 and the subtype representing the area where this game is actually taking place, for instance subtype 2 for a game taking place in Boston. An event or service is associated to predefined blackout types and subtypes through the IMS editor.

4.20.4 Blackout operation

Along with the event, data are broadcast that define the blackout type and subtype. Upon reception of that information the ICC will compare it with its own blackout information, and if both pieces match, the event is not unscrambled.

4.20.5 Blackout Over Free Access Precedence

Even for event or services in free access mode (they can be viewed by all valid ICC bearers) blackouts are enforced exactly as described above.

4.20.6 Reverse Blackout

The opposite operation to blackout is supported in the system. The blackout type and subtypes used to blackout a certain set of areas can be used instead to enable these areas and blackout all the others.

4.20.7 Zip Code Information Update

New Zip code information should be provided to the IMS at least 2 months before being used by the SMS.

4.21 Event runs late

In case an event runs late (overtime in sport event for instance), this constitutes an emergency for which events have to be modified at the Event editor console. Modifying the event over the EPG data feed is possible but it would take much more time before the change is really effective in the database. Then one must notify the change to the broadcast manager so that the IMS regenerates at once the EPG data, but more importantly it continues sending the ECM attached to the same event and not the following one (which may prevent the subscriber from viewing the overtime...).

To avoid overwriting the emergency change with the EPG data feed, the IMS operator must lock all events modified through the IMS editor. The services carrying those events will not be updated by EPG data feeds until all locked events are in the past.

4.22 Changing to/from daylight savings time

The STB always keeps track of the current local time thanks to information provided by the EPG and by the ICC. The IMS generates batch EMMs containing time zone information per zip code. These EMMs are sent continuously to all ICCs. Furthermore, the EPG will carry the GMT and the information whether daylight savings time is currently in effect or not. This data allow the STB to always know the current local time.

4.23 Tokens (FUTURE)

Tokens are elementary units that can be used to impulsively purchase events within an event package. Tokens are prepaid and are associated to event package products. If a customer purchases an event package product, the right received contains the quantity of tokens associated with the package. Then events in the package are impulsively purchasable by the customer using tokens, as long as there are enough tokens left. A number of tokens is associated with an event package product through Command 307, and can be modified with Command 308.

Because event packages are ordered by calling the SMS, the list of the events purchased is not reported during callback.

4.24 SMS vs ICC customer data discrepancy

If a discrepancy is suspected between customer data as it is known by the SMS and as it is in the ICC, the following commands must be issued:

- Command 110 EMM cleanup
- Command 7 All products cancellation. During the period when this EMM is broadcast (about 4 hours), the subscriber will not have any rights on the ICC.

Subsequently, the SMS must resend products to this customer. This sequence of commands will cause all products to be erased, including all future IPPVs. It is the responsibility of the customer to purchase these event products again, either impulsively (IPPVs) or through the SMS. The credit data of the customer will be restored during the next callback.

5. Appendix: SMS EMM Commands

5.1 Command 2: Add Product

Description

Used to add a new product (service product, non-impulsively purchasable event product and package product) to an ICC. This command must not be used to renew an existing product or to send an impulsively purchasable event product to a subscriber. Command 3 must be used to renew an existing product and Command 10 to send an impulsively purchasable event product (which will then appear in the purchase list).

Syntax

Command 2: Add Product			
Field	**byte**	**format**	**Description**
command_ID	4	0002	command_ID = 2
IMS_product_ID	12	000000000000 to 999999999999	IMS product ID
begin_date	8	YYYYMMDD	Subscription begin date. Subscription is not valid before this date.
end_date	8	YYYYMMDD	Subscription end date. Subscription is not valid after this date.

SMS Acknowledge

Field	**Error codes**	**Error codes extension**
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
IMS_product_ID	BAD_COMMAND_SYNTAX PRODUCT_NOT_FOUND	BAD_IMS_PRODUCT_ID_FORMAT NO_EXTENDED_ERROR_CODE
begin_date	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE
end_date	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE

Gateway interface concerned

SMS and SAS

5.2 Command 3: Product Renewal

Description

Used to indicate that the subscriptions included in the product have been renewed up to a specified date. All subscription services included in the product are renewed to the same date.

Syntax

Command 3: Product Renewal			
Field	**byte**	**format**	**Description**
command_ID	4	0003	command_ID = 3
IMS_product_ID	12	000000000000 to 999999999999	IMS Product ID
end_date	8	YYYYMMDD	Subscription end date. Subscription is not valid after this date.

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
IMS_product_ID	BAD_COMMAND_SYNTAX PRODUCT_NOT_FOUND	BAD_IMS_PRODUCT_ID_FORMAT NO_EXTENDED_ERROR_CODE
end_date	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE

Gateway interface concerned

SMS and SAS

5.3 Command 4: Product Suspension

Description

Suspends the subscription to services of the product. The subscriber will not be able to watch the product until a product reactivation command is sent. This command may be used when there is a payment problem with the subscriber. This command does not impact callbacks.

Syntax

Command 4: Product Suspension			
Field	**byte**	**format**	**Description**
command_ID	4	0004	command_ID = 4
IMS_product_ID	12	000000000000… 999999999999	IMS Product ID

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
IMS_product_ID	BAD_COMMAND_SYNTAX PRODUCT_NOT_FOUND	BAD_IMS_PRODUCT_ID_FORMAT NO_EXTENDED_ERROR_CODE

Gateway interface concerned

SMS and SAS

5.4 Command 5: Product Reactivation

Description

Grants access to all the subscriptions included in the product that has been previously suspended. The subscriber will be able to watch the product again. Used after *command 4: Product Suspension.*

Syntax

Command 5: Product Reactivation			
Field	**byte**	**format**	**Description**
command_ID	4	0005	command_ID = 5
IMS_product_ID	12	000000000000… 999999999999	IMS Product ID

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
IMS_product_ID	BAD_COMMAND_SYNTAX PRODUCT_NOT_FOUND	BAD_IMS_PRODUCT_ID_FORMAT NO_EXTENDED_ERROR_CODE

Gateway interface concerned

SMS and SAS

5.5 Command 6: Product Cancellation

Description

Allows the SMS to remove the product (subscription services) from the ICC in case of error or if a subscriber asked for a cancellation. Event products (purchased through the SMS) which have been purchased with a viewing time in the future may also be canceled through this command. The cancellation of a non-impulsively purchased event product has no impact on the subscriber's credit in the ICC.

Syntax

Command 6: Product Cancellation			
Field	**byte**	**format**	**Description**
command_ID	4	0006	command_ID = 6
IMS_product_ID	12	000000000000… 999999999999	IMS Product ID

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
IMS_product_ID	BAD_COMMAND_SYNTAX PRODUCT_NOT_FOUND	BAD_IMS_PRODUCT_ID_FORMAT NO_EXTENDED_ERROR_CODE

Gateway interface concerned:

SMS and SAS

5.6 Command 7: All products cancellation

Description

Allows the SMS to remove all products stored on the ICC. This command does not affect non-call collected IPPVs that have been watched (this also applies to events in progress). Such IPPVs will be normally collected when the next callback occurs. However, non-watched IPPVs will be removed. The credit will also be suspended.

Syntax

Command 7: All products cancellation			
Field	**byte**	**format**	**Description**
command_ID	4	0007	command_ID = 7

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID

Gateway interface concerned

SMS and SAS

5.7 Command 8: Credit management

Description

This command is used to manage an existing credit reserved for impulse-pay-per-view on the ICC. The credit is downloaded the first time through the *command 13:Create credit for impulse purchase*. This command may be used, for instance, to update immediately the credit limit of a subscriber's ICC.

Important remark

The credit information of a smartcard can be modified only once per day (GMT day). If a second modification is sent to the smartcard, it will be ignored. The credit update during a callback must also be taken into account.

Syntax

Command 8: Credit management			
Field	**byte**	**format**	**Description**
command_ID	4	0008	command_ID = 8
credit_mode	2	01 = ADD	Increase the credit. Exceptionally used to add credit to all subscribers or group of subscribers. The credit being usually handled with mode SET CREDIT or SET BALANCE.
		02 = SUBTRACT	Decrease the credit. Used in case of error correction.
		03 = SET CREDIT	Set the new credit value.
		04 = SET BALANCE	Set the debit to 0 and the credit to the new value.
		05 = SUB OFFSET	Subtract the debit and credit by the same amount. Used to avoid counters overflow on the ICC.
credit	7	0000000…6553599	Credit amount (in the local currency) representing the range $00000.00 to $65535.99

SMS Acknowledge

Field	**Error codes**	**Error codes extension**
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
credit_mode	BAD_COMMAND_SYNTAX	BAD_CREDIT_MODE
credit	BAD_COMMAND_SYNTAX	BAD_CREDIT_FORMAT

Gateway interface concerned

SMS and SAS

5.8 Command 9: Update Credit Threshold

Description

Updates the impulse credit threshold value below which the ICC should call back the Call Collector. This command should **not** be used to initialize the Credit Threshold value. The initialization must be done through the *command 13: Create Credit For Impulse Purchase*. If the credit threshold (field threshold_credit) is set to 0 (zero), then no callback will be issued when the threshold limit is reached.

Syntax

Command 9: Update Credit Threshold			
Field	**byte**	**format**	**Description**
command_ID	4	0009	command_ID = 9
threshold credit	7	0000000…6553599	Lower limit under which the ICC must do a low credit call back representing the range $000.00 to $65535.99

SMS Acknowledge

Field	**Error codes**	**Error codes extension**
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
threshold credit	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_THRESHOLD_CREDIT_FORMAT CREDIT_THRESHOLD_TOO_HIGH

Gateway interface concerned

SMS and SAS

5.9 Command 10: Add Event Product

Description

Adds an event product right in the ICC. Used to allow PPV purchase by phoning the customer service center. If the product referenced by IMS_product_ID is no longer valid then a PPV_IN_THE_PAST error is raised. When an event product is purchased through the SMS (by calling customer service), the ICC credit and debit values are not impacted.

Syntax

Command 10: Add Event Product			
Field	**byte**	**format**	**Description**
command_ID	4	0010	command_ID = 10
IMS_product_ID	12	000000000000… 999999999999	IMS Product ID
length_event_name	2	Up to 17 characters	Length of event_name
event_name	32	Alpha/Num	Event name as displayed in the PPV purchase history.
price	5	00000 to 99999	Price of the product, representing $000.00 to $999.99

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
IMS_product_ID	BAD_COMMAND_SYNTAX PRODUCT_NOT_FOUND PPV_IN_THE_PAST	BAD_IMS_PRODUCT_ID_FORMAT NO_EXTENDED_ERROR_CODE NO_EXTENDED_ERROR_CODE
length_event_name	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	LENGTH_TOO_LONG BAD_NUMBER_FORMAT
price	BAD_COMMAND_SYNTAX	BAD_PRICE_FORMAT

Note

The event_name field must be filled in as indicated in section 3.4, left aligned, blank padded and capitalized. Its length is always 32 bytes.

Gateway interface concerned

SMS and SAS

5.10 Command 13: Create Credit for Impulse Purchase

Description

This command creates the initial right to allow the subscriber to perform impulse PPV purchase. This command is sent only at the ICC initialization process. If threshold_credit is equal to 0 (zero) – No callback will be issued when the threshold limit is reached.

Syntax

Command 13: Create Credit for Impulse Purchase			
Field	**byte**	**format**	**Description**
command_ID	4	0013	command_ID = 13
credit	7	0000000…6553599	Credit amount (in the local currency). Representing $000.00 to $65535.99.
threshold_credit	7	0000000…6553599	Lower limit under which the ICC must initiate a low credit call back. Representing $000.00 to $65535.99.

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
credit	BAD_COMMAND_SYNTAX	BAD_CREDIT_FORMAT
threshold_credit	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_THRESHOLD_CREDIT_FORMAT CREDIT_THRESHOLD_TOO_HIGH

Gateway interface concerned

SMS and SAS

5.11 Command 14: Suspend impulse purchase

Description

This command suspends the privilege of making impulse purchases. Reactivation of impulse purchases may be completed using *command 15: Reactivate impulse purchase*. This command has no impact on callback operations.

Syntax

Command 14: Suspend Impulse Purchase			
Field	**byte**	**format**	**Description**
command_ID	4	0014	command_ID = 14

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID

Gateway interface concerned

SMS and SAS

5.12 Command 15: Reactivate impulse purchase

Description

Reactivates the privilege of making impulse purchases. Used after *command 14: Suspend impulse purchase.*

Syntax

Command 15: Reactivate Impulse Purchase			
Field	**byte**	**format**	**Description**
command_ID	4	0015	command_ID = 15

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID

Gateway interface concerned

SMS and SAS

5.13 Command 20: Suspend subscriber ICC

Description

Suspends all subscriptions on the ICC and the impulse purchase capability. The SMS can still continue to send EMM command to a deactivated ICC (no error returned). To reactivate the subscriber ICC use *command 21: Reactivate subscriber ICC.* This command does not impact callback operations.

Syntax

Command 20: Suspend subscriber ICC			
Field	**byte**	**format**	**Description**
command_ID	4	0020	command_ID = 20

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID

Gateway interface concerned

SMS and SAS

5.14 Command 21: Reactivate subscriber ICC

Description

Grants access to all the subscriptions again. Used after *command 20: Suspend subscriber ICC.*

Syntax

Command 21: Reactivate subscriber ICC			
Field	**byte**	**format**	**Description**
command_ID	4	0021	command_ID = 21

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID

Gateway interface concerned

SMS and SAS

5.15 Command 22: Reserved

Description
This command is reserved for Nagravision usage.

5.16 Command 30: Reserved

Description
This command is reserved for Nagravision usage.

5.17 Command 31: Reserved

Description
This command is reserved for Nagravision usage.

5.18 Command 32: Reserved

Description
This command is reserved for Nagravision usage.

5.19 Command 48: Set Zip code

Description

Sets or updates the Zip code on the ICC. This should also be used when the subscriber moves to another location. Zip code information is used for blackout and time zone management.

Syntax

Command 48: Set Zip code			
Field	**byte**	**format**	**Description**
command_ID	4	0048	command_ID = 48
zip_code	5	00000…99999	subscriber's zip code[2]

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
zip_code	BAD_COMMAND_SYNTAX	BAD_ZIP_CODE_FORMAT

Gateway interface concerned

SMS and SAS

[2] This is the US zip code format, the UK 10 alpha/num format is as well supported. This is configured at the system setup.

5.20 Command 49: Set Callback phone number

Description

Sets or updates the ICC field containing the phone-number or MAC address to be used to call back the Call Collector.

The number can be a phone number or a MAC address. The STB will determine itself how the number must be interpreted:

- If it is a phone number, 18005551212 (with 5 trailing spaces) represents 1-800-555-1212.
- If it is a MAC address in decimal, 281474976710655 (with 1 trailing space) represents the MAC address FFFFFFFFFFFF.

To set an IP address, command 54 (Set Callback IP address) must be used.

Syntax

Command 49: Set Callback phone number			
Field	**byte**	**format**	**Description**
command_ID	4	0049	command_ID = 49
cc_number_1	16	Alpha/Num	Call Collector phone number. This field must be padded with trailing ASCII space characters for numbers not requiring 16 digits. Example: 18005551212 (with 6 leading spaces) represents 1-800-555-1212

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
cc_number_1	BAD_COMMAND_SYNTAX	BAD_PHONE_NUMBER_FORMAT

Gateway interface concerned

SMS and SAS

5.21 Command 50: Cancel ICC

Description

An ICC cancellation is performed when a card is removed from operation (stolen, lost, or presumed failed for example). Because the ICC *is not recoverable* after such operation, the SMS is not allowed to reuse the ICC. In addition the ICC is flagged as canceled and deactivated in the SAS database.

Syntax

Command 50: Cancel ICC			
Field	**byte**	**format**	**Description**
command_ID	4	0050	command_ID = 50

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID

Gateway interface concerned

SMS and SAS

5.22 Command 51: Initialize Card

Description
Initializes the Management Operator (MOP) zone of a new ICC. This command is used when a new ICC is registered in the SMS. In addition the ICC is flagged as initialized in the SAS database. This command must be performed before subscriptions will be allowed on the ICC. Initialization of an ICC can occur several times and has no effect on the rights stored in the ICC.

Syntax

Command 51: Initialize Card			
Field	**byte**	**format**	**Description**
command_ID	4	0051	command_ID = 51

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID

Gateway interface concerned

SMS and SAS

5.23 Command 52: Pair the ICC with the STB

Description

Used to pair an ICC with the STB. This command is mandatory before any services may be authorized. The ICC must have previously been initialized with *command 51: Initialize ICC* before pairing can take place. This command may be issued several times to pair an ICC with a different STB.

Important remark

The pairing-key process is limited to a 4 bytes value, which is converted in decimal, the STU_number will be then a 10 digits string followed by four (4) space characters.

Syntax

Command 52: Pair the ICC with the STB			
Field	**byte**	**format**	**Description**
command_ID	4	0052	command_ID = 52
STU_number	14	Num	Nagravision STB serial number in decimal.

SMS Acknowledge

Field	**Error codes**	**Error codes extension**
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
STU_number	BAD_COMMAND_SYNTAX	BAD_STU_NUMBER_FORMAT

Gateway interface concerned

SMS and SAS

5.24 Command 53: Clear PIN code

Description

Clears the PIN code. This command is used when the subscriber requests a PIN code reset. It is the responsibility of the customer service center to check the authority of the person asking for the cancellation.

Syntax

Command 53: Clear PIN code			
Field	**byte**	**format**	**Description**
command_ID	4	0053	command_ID = 53

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID

Gateway interface concerned

SMS and SAS

5.25 Command 54: Set Callback IP address

Description

Sets or updates the ICC field containing the IP-address / phone-number of the Call Collector.

Syntax

Command 54: Set Callback IP address			
Field	**byte**	**format**	**Description**
command_ID	4	0054	command_ID = 54
ip_address	15	Alpha/Num	Call Collector IP address. Field format is 000.000.000.000 to 255.255.255.255. The length is fixed to 15 bytes and the sub-addresses are filled with leading "0" characters. **Example:** "001.112.025.002"
CC_port	5	Num	Call Collector TCP/IP port.

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
ip_address	BAD_COMMAND_SYNTAX	BAD_IP_ADDRESS_FORMAT
CC_port	BAD_COMMAND_SYNTAX	BAD_CC_PORT_FORMAT

Gateway interface concerned

SMS and SAS

5.26 Command 60: Immediate Call Back

Description

Requests an immediate feedback information from an ICC. 4 separate FEEDBACK commands will be generated and sent to the SMS:

Command 201: Current credit and debit
Command 202:PPV purchase list
Command 205: Calling phone discrepancies (if ANI is enabled)
Command 206:STU Responding Status (responding = Y)
Command 211: Start of report
Command 212: End of report

In the event the STB does not phone back within the designated time defined by the call collector the following FEEDBACK command will be generated:

Command 206: STU Responding Status (responding = N)

Syntax

Command 60: Immediate Call Back			
Field	**byte**	**format**	**Description**
command_ID	4	0060	command_ID = 60
CbDate	8	YYYYMMDD	Date of call back. **The provided value is ignored and replaced by the call collector**
CbTime	6	HHMMSS	Time of call back. **The provided value is ignored and replaced by the call collector**

SMS ACK Acknowledge

Field	**Error codes**	**Error codes extension**
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID

Gateway interface concerned

SMS and SAS

5.27 Command 61: Enable Automatic Call Back

Description
Turns on the automatic feedback feature of the STB. Starting from there, the STB will call at the given date and after every given period. Automatic callback timing is not impacted by event or immediate generated callbacks. 4 separate FEEDBACK commands will be generated and sent to the SMS:

Command 201: Current credit and debit
Command 202: PPV purchase list
Command 205: Calling phone discrepancies (if ANI is enabled)
Command 206: STU Responding Status (responding = Y)

In the event the STB does not phone back within the designated time defined by the call collector the following FEEDBACK command will be generated by the SMS.

Command 206: STU Responding Status (responding = N)

Syntax

Command 61: Enable Automatic Call Back			
Field	**byte**	**format**	**Description**
command_ID	4	0061	command_ID = 61
call_freq	2	01 02 03 04 05 1m	annual semi-annual quarterly monthly bi-monthly every m days (1 should be considered as a flag)
date first call	8	YYYYMMDD	first date on which the ICC should call back.
Cbtime	6	HHMMSS	Time of call back. **The provided value is ignored and replaced by the call collector**

SMS Acknowledge

Field	**Error codes**	**Error codes extension**
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
call_freq	BAD_COMMAND_SYNTAX	BAD_FREQUENCY_FORMAT
date first call	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT

Gateway interface concerned

SMS and SAS

5.28 Command 62: Disable Automatic Call Back

Description

Disables automatic call back for an ICC. Callbacks triggered by Memory_full, threshold limit reached, special events, or callback on demand will still occur.

Syntax

Command 62: Disable Automatic Call Back			
Field	**byte**	**format**	**Description**
command_ID	4	0062	command_ID = 62

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID

Gateway interface concerned

SMS and SAS

5.29 Command 63: Reserved

Description
This command is reserved for Nagravision.

5.30 Command 64: Update event right

Description

This command updates the End_date for event products and event package products in the ICCs. It is generated automatically by the IMS when an event schedule change is detected for an event product or for an event belonging in an event package product.

Syntax

Command 64: Update event right			
Field	**byte**	**format**	**Description**
command_ID	4	0064	command_ID = 64
IMS_product_ID	12	000000000000… 999999999999	IMS Product ID that covers the data stream.
End_date	8	YYYYMMDD	The new end date of the event product.

IMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
IMS_product_ID	PRODUCT_NOT_FOUND	NO_EXTENDED_ERROR_CODE
End_date	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT

Gateway interface concerned

IMS and SAS

5.31 Command 65 to 68: Reserved

Description

Command 65 to 68 are reserved for Nagravision.

5.32 Command 69: Send Generic IRD Command

Description
This command allows SMS to send any command to the STB. In this way, the CA system provides a secure transport mechanism between the head-end and the STB.

The data received by the STB consists of the following data structure. The SMS Gateway calculates the field's `sequence_number` and `checksum` for the convenience of SMS.

```
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
}
```

- Command definitions are STB's responsibility.

Syntax

Command 69: Send Generic IRD Command			
Field	**byte**	**format**	**Description**
command_ID	4	0069	command_ID = 69
IRD_command_id	3	Num	command_id field of IRD command_body (0..255)
IRD_operation	3	Num	operation field of IRD command_body (0..255)
IRD_data_length	2	Num	length in **bytes** of useful part of IRD_data field (0..48)
IRD_data	96	Alpha/Num	data field 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.

SMS Acknowledge

Field	**Error codes**	**Error codes extension**
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
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

Gateway interface concerned

SMS and SAS

5.33 Command 70: Reserved

Description

Command 70 is reserved for Nagravision.

5.34 Command 71: Get Products

Description

This command allows SMS to retrieve all the non-impulsive products (subscriptions or PPV) known to the CA system for a subscriber. The information comes from the CA system databases. The products are returned asynchronously in command 215: Products List.

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

Syntax

Command 71: Get Products			
Field	**byte**	**format**	**Description**
command_ID	4	0071	command_ID = 71

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID

Gateway interface concerned

SMS and SAS

5.35 Command 72: Set Products

Description
This command allows modifying the list of products on the subscriber smart card. This command operates in conjunction with the known data in Positive Addressing to send only the minimum required EMMs.

The following is a summary of the command algorithm. The update of the known state is never mentioned, but of course occurs as necessary.

1. Check the command syntax and semantics[3].
   If bad, send NACK and stop
2. Remove duplicate entries in the command product list.
   If two entries have the same product-id but another field is different, send NACK and stop
3. Check that all the IMS-product-ids in the command product list exist and are of the correct type
   If not, send NACK and stop
4. If the force-EMM flag is set, send commands unconditionally according to the command content.
5. If the command suspend-ICC flag is set and different from the known state, produce the equivalent of command 20: Suspend subscriber ICC
6. If the command suspend-ICC flag is not set and different from the known state, produce the equivalent of command 21: Reactivate subscriber ICC
7. If the nb-of-products field is 0 and the type_of_products field is B (empty command list), produce an equivalent of command 7: All products cancellation
8. For all subscription products in the command that are known to Positive Addressing, all fields taken into account:
   Refresh the subscription on the card if the validity dates are different.
9. For all products in the command that are not known to Positive Addressing, taking the type into account:
   Produce an equivalent of command 2: Add Product
10. For all products known to Positive Addressing that are not in the command, taking the type into account:
    Produce an equivalent of command 6: Product Cancellation
11. For all products known to Positive Addressing that are in the command, taking the type into account:
    If the suspend-product flag is different, produce the equivalents of commands 4: Product Suspension and/or 5: Product Reactivation as necessary.

Note: If for some reason, the command is NACKed by the SMS Gateway, it needs to be sent again with the force-EMM flag set.

[3] Explicitly, this means subscription_begin_date and subscription_end_date must be valid dates, that subscription_end_date is in the future and that subscription_begin_date < subscription_end_date.

Syntax

Command 72: Set Products			
Field	**byte**	**format**	**Description**
command_ID	4	0072	command_ID = 72
force_emm	1	Y..N	If set to Y, EMMs are sent unconditionally
suspend_ICC	1	Y..N	ICC Suspension status
type_of_products	1	S, E or B	Types of products defined in this command: S: Service products E: Event products B: Both event and service products
begin_date	8	YYYYMMDD	Subscriptions begin date. Subscriptions are not valid before this date. For event products, this date is ignored.
end_date	8	YYYYMMDD	Subscriptions end date. Subscriptions are not valid after this date. For event products, this date is ignored.
nb_of_products	2	00..99	Number of products defined in this command
for(i=0;i<nb_of_products;i++) {	3	Num	IRD command length
IMS_product_ID	12	000000000000 to 999999999999	IMS product ID Products must be of the type specified above The same product may occur multiple times
suspend_product	1	Y..N	Product suspension status
}			

CH - 1033 Cheseaux, Switzerland. +41.21.732.03.11 +41.21.732.03.00

SAS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
suspend_ICC	BAD_COMMAND_SYNTAX	BAD_FLAG_VALUE
type_of_products	BAD_COMMAND_SYNTAX	BAD_FLAG_VALUE
nb_of_products	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
IMS_product_ID	BAD_COMMAND_SYNTAX PRODUCT_NOT_FOUND PRODUCT_ALREADY_EXISTS PRODUCT_INCONSISTENT	BAD_IMS_PRODUCT_ID_FORMAT NO_EXTENDED_ERROR_CODE DIFFERENT_PRODUCTS NO_EXTENDED_ERROR_CODE
suspend_product	BAD_COMMAND_SUNTAX	BAD_FLAG_VALUE
begin_date	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE
end_date	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE
	CANCELED_CARD	NO_EXTENDED_ERROR_CODE

Gateway interface concerned

SMS and SAS

5.36 Command 73: Add ALC product

Description

Used to add a new “à la carte” product to an ICC. This allows the subscriber to pick up m services amongst n.

Syntax

Command 73: Add ALC Product			
Field	**byte**	**format**	**Description**
command_ID	4	0073	command_ID = 73
IMS_product_ID	12	000000000000 to 999999999999	IMS product ID
begin_date	8	YYYYMMDD	Subscription begin date. Subscription is not valid before this date.
end_date	8	YYYYMMDD	Subscription end date. Subscription is not valid after this date.
service_number	3	000 to 999	Number of services in the product
for (I=1; I<=number_services; I++) {			
service_UID	5	00000 … 99999	IMS service UID
}			

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
IMS_product_ID	BAD_COMMAND_SYNTAX PRODUCT_NOT_FOUND	BAD_IMS_PRODUCT_ID_FORMAT NO_EXTENDED_ERROR_CODE
begin_date	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE
end_date	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE
service_number	BAD_COMMAND_SYNTAX TOO_MANY_ITEMS	BAD_SERVICE_NUMBER_FORMAT maximum item #
service_UID	BAD_COMMAND_SYNTAX SERVICE_NOT_FOUND	BAD_SERVICE_UID_FORMAT NO_EXTENDED_ERROR_CODE

Gateway interface concerned

SMS and SAS

5.37 Command 74: Modify ALC product

Description

Used to modify a “à la carte” product in an ICC.

Syntax

Command 74: Modify ALC Product			
Field	**byte**	**format**	**Description**
command_ID	4	0074	command_ID = 74
IMS_product_ID	12	000000000000 to 999999999999	IMS product ID
begin_date	8	YYYYMMDD	Subscription begin date. Subscription is not valid before this date.
end_date	8	YYYYMMDD	Subscription end date. Subscription is not valid after this date.
service_number	3	000 to 999	Number of services in the product
for (I=1; I<=number_services; I++) {			
service_UID	5	00000 … 99999	IMS service UID
}			

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
IMS_product_ID	BAD_COMMAND_SYNTAX PRODUCT_NOT_FOUND	BAD_IMS_PRODUCT_ID_FORMAT NO_EXTENDED_ERROR_CODE
begin_date	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE

end_date	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE
service_number	BAD_COMMAND_SYNTAX TOO_MANY_ITEMS	BAD_SERVICE_NUMBER_FORMAT maximum item #
service_UID	BAD_COMMAND_SYNTAX SERVICE_NOT_FOUND	BAD_SERVICE_UID_FORMAT NO_EXTENDED_ERROR_CODE

Gateway interface concerned

SMS and SAS

5.38 Command 75: Renew ALC product

Description

Used to renew the “à la carte” subscription in the ICC up to the end_date.

Syntax

Command 75: Renew ALC Product			
Field	**byte**	**format**	**Description**
command_ID	4	0075	command_ID = 75
end_date	8	YYYYMMDD	Subscription end date. Subscription is not valid after this date.

SMS Acknowledge

Field	**Error codes**	**Error codes extension**
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
end_date	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE

Gateway interface concerned

SMS and SAS

5.39 Command 76: ALC product Suspension

Description

Suspends the subscription to services of the “à la carte” product. The subscriber will not be able to watch the product until a product reactivation command is sent. This command may be used when there is a payment problem with the subscriber.

Syntax

Command 76: Suspend ALC Product

Field	byte	format	Description
command_ID	4	0076	command_ID = 76

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID

Gateway interface concerned

SMS and SAS

5.40 Command 77: ALC product Reactivation

Description

Grants access to all the subscriptions included in the "à la carte" product that has been previously suspended. The subscriber will be able to watch the product again. Used after *command 76: ALC Product Suspension.*

Syntax

Command 77: Activate ALC Product			
Field	**byte**	**format**	**Description**
command_ID	4	0077	command_ID = 77

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID

Gateway interface concerned

SMS and SAS

5.41 Command 78: ALC product Cancellation

Description

Allows the SMS to remove the "à la carte" product from the ICC in case of error or if a subscriber asked for a cancellation.

Syntax

Command 78: Suspend ALC Product			
Field	**byte**	**format**	**Description**
command_ID	4	0078	command_ID = 78

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID

Gateway interface concerned

SMS and SAS

5.42 Command 79: Force Tune (future use)

Description

Forces the STB to tune a desired service.

Syntax

Command 79: Force Tune			
Field	**byte**	**format**	**Description**
command_ID	4	0079	command_ID = 79
network_id	5	00000 … 65535	corresponds to the network_id as described in the DVB Network Information Table (NIT).
transport_id	5	00000 … 65535	corresponds to the network_id as described in the DVB Network Information Table (NIT).
service_id	5	00000 … 65535	corresponds to the service_id as described in the DVB Service Description Table (SDT). It may also correspond to the program number found in the MPEG Program Map Table (PMT)

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
network_id	BAD_COMMAND_SYNTAX	BAD_NETWORK_ID_FORMAT
transport_id	BAD_COMMAND_SYNTAX	BAD_TRANSPORT_ID_FORMAT
service_id	BAD_COMMAND_SYNTAX	BAD_SERVICE_ID_FORMAT

Gateway interface concerned

SMS and SAS

5.43 Command 80: Send message (future use)

Description

Allows the SMS to send message and fingerprints to a STB.

Syntax

Command 80: Send message			
Field	**byte**	**format**	**Description**
command_ID	4	0080	command_ID = 80
id	4	0000 .. 9999	unique message identification
lid	3	aaa .. zzz	language code compliant with ISO639_2
version	2	00 .. 99	message version
priority	1	0 .. 1	message priority (0 => low, 1 => high)
mode	1	0 .. 1	display mode (0 => normal, 1 => fingerprint)
kind	1	0 .. 9	message kind (e.g. advertisement, warning etc ...), to be defined between the SMS and the STB provider 0 is the default value
validity_start_date	8	YYYYMMDD	message validity start date
validity_start_time	4	HHMM	message validity start time
validity_end_date	8	YYYYMMDD	message validity end date
validity_end_time	4	HHMM	message validity end time
length	3	000 .. 999	message length
message		alpha/num	message

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
lid	BAD_COMMAND_SYNTAX	BAD_LID_FORMAT
priority	BAD_COMMAND_SYNTAX	BAD_PRIORITY_FORMAT
mode	BAD_COMMAND_SYNTAX	BAD_MODE_FORMAT
kind	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
id	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
version	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
validity_start_date	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT
	BAD_COMMAND_SYNTAX	BAD_DATE_SEQUENCE
validity_start_time	BAD_COMMAND_SYNTAX	BAD_TIME_FORMAT
validity_end_date	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT
	BAD_COMMAND_SYNTAX	BAD_DATE_SEQUENCE
validity_end_time	BAD_COMMAND_SYNTAX	BAD_TIME_FORMAT
length	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT

Gateway interface concerned

SMS and SAS

5.44 Command 90: Reserved

Description

Command 90 is reserved for Nagravision.

5.45 Command 91: Reserved

Description

Command 91 is reserved for Nagravision.

5.46 Command 92: Purge old Products (future use)

Description

Purges the ICC database from old rights. This command is usually used in a system with impulse purchase capability without return path. Only the rights with end date in the past are purged.

Syntax

Command 92: purge old Products			
Field	**byte**	**format**	**Description**
command_ID	4	0092	command_ID = 92
purge_mode	1	0 .. 9	0 => purge only rights marked as call collected 1 => purge only rights marked as non-call collected 2 => purge both kind of rights 3 .. 9 => future use
date		0 .. 9	0 => purge only rights marked as call collected

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
mode	BAD_COMMAND_SYNTAX	BAD_PURGE_MODE_FORMAT

Gateway interface concerned

SMS and SAS

5.47 Command 93: Reserved

Description
This command is reserved for Nagravision usage.

6. Appendix: SMS CONTROL Commands

6.1 Command 100: Redefine Credit Limit

Description

Tells the Call Collector which credit limit for impulse purchase will be associated to an ICC. The Call Collector will restore the ICC credit to the sum of this value and the current debit at each billing date. The new credit limit will be communicated to the ICC during the next callback from the STB. If an immediate update on the ICC is required, the SMS should send a command 100 followed by command 8.

Syntax

Command 100: Redefine Credit Limit			
Field	**byte**	**format**	**Description**
command_ID	4	0100	command_ID = 100
credit_limit	7	0000000 … 6553599	the credit limit representing the range $000.00 to $65535.99

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
credit_limit	BAD_COMMAND_SYNTAX	BAD_CREDIT_FORMAT

Gateway interface concerned

SMS and CC

6.2 Command 101: Set Authorized Phone Number

Description

Sets or updates the Call Collector database with the phone numbers an ICC is allowed to use as caller id. These will be checked upon each callback. Command 205 will be generated from the SMS in case there is a difference.

The number can be a phone number, or an IP address or a MAC address. See command 49 for the exact description.

Syntax

Command 101: Set Authorized Phone Number			
Field	**byte**	**format**	**Description**
command_ID	4	0101	command_ID = 101
phone_number_1	16	Num + trailing spaces	Primary phone number. This field must be padded with trailing ASCII space characters for numbers not requiring 16 digits.
phone_number_2	16	Num + trailing spaces	Alternate phone number. This field must be padded with trailing ASCII space characters for numbers not requiring 16 digits.
phone_number_3	16	Num + trailing spaces	Alternate phone number. This field must be padded with trailing ASCII space characters for numbers not requiring 16 digits.

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
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

Gateway interface concerned

SMS and CC

6.3 Command 104: Create ICC On Call Collector

Description

Creates a new ICC record in the CC database. The correctly paired STB serial number must be provided for proper operation.

Syntax

Command 104: Create ICC On Call Collector			
Field	**byte**	**format**	**Description**
command_ID	4	0104	command_ID = 104
STU_number	14	Num	Nagravision STB serial number

SMS Acknowledge

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

Gateway interface concerned

SMS and CC

6.4 Command 105: Cancel ICC On Call Collector

Description

Cancels an existing ICC in the CC database. This command should be issued when the ICC is deleted from the SMS database.

Syntax

Command 105: Cancel ICC On Call Collector			
Field	**byte**	**format**	**Description**
command_ID	4	0105	command_ID = 105

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID

Gateway interface concerned

SMS and CC

6.5 Command 110: EMM cleanup

Description

Erases all EMMs in the SAS and EMM broadcaster for one ICC. This command has to be used if a discrepancy is suspected between SMS customer data and EMMs for this customer.

Syntax

Command 110: EMM cleanup			
Field	**byte**	**format**	**Description**
command_ID	4	0110	command_ID = 110

SAS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID

Gateway interface concerned

SMS and SAS

6.6 Command 111: Get History From Call Collector

Description

Allows the SMS to retrieve a two-month history of IPPV information stored in the Call Collector. Each PPV record will generate a separate *Command 202: PPV Purchase List* command from the SMS.

Syntax

Command 111: Get History From Call Collector			
Field	**byte**	**format**	**Description**
command_ID	4	0111	command_ID = 111

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID

Gateway interface concerned

SMS and CC

7. Appendix: SMS Feedback Commands

7.1 Command 200: Low credit alarm

Description

The credit level in the ICC is under the threshold credit. At the time when the SMS is made aware of this alarm, the credit has already been restored by the CC.

Syntax

Command 200: Low credit alarm			
Field	**byte**	**format**	**Description**
command_ID	4	0200	command_ID = 200
STU_number	14	Num	Nagravision STB serial number
credit	7	0000000 … 6553599	Credit amount representing $000.00 to $65535.99
debit	7	0000000 … 6553599	Debit amount representing $000.00 to $65535.99

Gateway interface concerned

CC and SMS

Errors must be reported by the SMS using the following table:

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

CH - 1033 Cheseaux, Switzerland. +41.21.732.03.11 +41.21.732.03.00

7.2 Command 201: Current Debit and Credit

Description

Sends the credit and debit reported by the ICC.

Syntax

Command 201: Current Debit and Credit			
Field	**byte**	**format**	**Description**
command_ID	4	0201	command_ID = 201
STU_number	14	Num	Nagravision STB serial number
credit	7	0000000 … 6553599	Credit amount representing $000.00 to $65535.99
debit	7	0000000 … 6553599	Debit amount representing $000.00 to $65535.99

Gateway interface concerned

CC and SMS

Errors must be reported by the SMS using the following table:

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

CH - 1033 Cheseaux, Switzerland. +41.21.732.03.11 +41.21.732.03.00

7.3 Command 202: PPV Purchase List

Description

Sends a single IPPV record to allow detailed billing.

Syntax

Command 202: PPV Purchase List			
Field	**byte**	**format**	**Description**
command_ID	4	0202	command_ID = 202
STU_number	14	Num	Nagravision STB serial number
IMS_product_ID	12	000000000000 … 999999999999	IMS_product_ID of the event product purchased impulsively through the EPG
purchase_date	8	YYYYMMDD	date of IPPV purchase
watched_status	1	Y or N	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 (see sections 4.1.2 and 4.1.4).

Gateway interface concerned

CC and SMS

Errors must be reported by the SMS using the following table:

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

7.4 Command 205: Phone Discrepancies

Description

Sends to the SMS any phone discrepancy information. Occurs if the phone number (caller_ID) of the customer does not match any authorized phone numbers for that given customer.

The number can be a phone number, or an IP address or a MAC address. See command 49 and command 54 for the exact description.

Syntax

Command 205: Phone Discrepancies			
Field	**byte**	**format**	**Description**
command_ID	4	0205	command_ID = 205
STU_number	14	Num	Nagravision STB serial number
phone_number_1	16	Alpha/Num	first phone number stored in the CC
phone_number_2	16	Alpha/Num	second phone number stored in the CC
phone_number_3	16	Alpha/Num	third phone number stored in the CC
abnormal_phone	16	Alpha/Num + trailing spaces	number used by STB to call CC

Gateway interface concerned

CC and SMS

Errors must be reported by the SMS using the following table:

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
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

7.5 Command 206: STU Responding Status

Description

Sends to the SMS the new STB responding status. This command is generated upon every successful callback as well as callbacks that are expected but not received. A non-responding STB is defined by the call collector's callback expiration parameter (typically 3 days). Automatic callbacks and immediate callbacks requested from the SMS will always generate the STB Responding Status command. Alarm based callbacks and Special event callbacks will never generate a STU Responding Status with responding = N.

Syntax

Command 206: STU Responding Status			
Field	**byte**	**format**	**Description**
command_ID	4	0206	command_ID = 206
STU_number	14	Num	Nagravision STB serial number
responding	1	Y … N	A STB may be responding (Y) or not (N).

Gateway interface concerned

CC and SMS

Errors must be reported by the SMS using the following table:

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
STU_number	BAD_COMMAND_SYNTAX	BAD_STU_NUMBER_FORMAT
responding	BAD_COMAMND_SYNTAX	BAD_FLAG_FORMAT

7.6 Command 207: ICC Memory Full Alarm

Description

Forwards to the SMS a message sent by the ICC: memory full. This information is retrieved by the CC during the callback. The ICC will remove all obsolete data (expired subscriptions).

Syntax

Command 207: ICC Memory Full Alarm			
Field	**byte**	**format**	**Description**
command_ID	4	0207	command_ID = 207
STU_number	14	Num	Nagravision STB serial number

Gateway interface concerned

CC and SMS

Errors must be reported by the SMS using the following table:

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
STU_number	BAD_COMMAND_SYNTAX	BAD_STU_NUMBER_FORMAT

7.7 Command 208: Event definition error

Description

Returned when an event overlapping is detected while loading a new version of the EPG data feed. All events validly defined in the EPG data feed are stored in the IMS database.

Syntax

Command 208: Event definition error			
Field	**byte**	**format**	**Description**
command_ID	4	0208	command_ID = 208
SMS_event_ID_1	12	000000000000 to 999999999999	SMS event ID which caused the error.
SMS_event_ID_2	12	000000000000 to 999999999999	SMS event ID which already exist in the IMS database.

Gateway interface concerned

IMS and SMS

Errors must be reported by the SMS using the following table:

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
SMS_event_ID_1	BAD_COMMAND_SYNTAX	BAD_SMS_EVENT_ID_FORMAT
SMS_event_ID_2	BAD_COMMAND_SYNTAX	BAD_SMS_EVENT_ID_FORMAT

7.8 Command 209: Null Event error

Description

Returned when an event in the IMS database has a duration of zero and a validity_start_date within the next 7 days. This can happen if an event product or an event package product referring to a non-existent event (outside of EPG range) is created through the SMS. In this case, a placeholder event is created in the IMS database with zero duration. This error is reported each time there is a new version of the EPG data feed and the problem is not solved.

Syntax

Command 209: Null Event error			
Field	**byte**	**format**	**Description**
command_ID	4	0209	command_ID = 209
SMS_event_ID	12	Num	SMS event ID of the event.

Gateway interface concerned

IMS and SMS

Errors must be reported by the SMS using the following table:

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
SMS_event_ID	BAD_COMMAND_SYNTAX	BAD_SMS_EVENT_ID_FORMAT

CH - 1033 Cheseaux, Switzerland.
+41.21.732.03.11
+41.21.732.03.00

7.9 Command 210: EPG data feed format error

Description

This command is used by the IMS to report EPG data feed format errors to the SMS. The field's error_code and error_code_ext must be used as indicated in § 3.3.

Syntax

Command 210: EPG data feed format error			
Field	**byte**	**format**	**Description**
command_ID	4	0210	command_ID = 210
error_code	4	0000 .. 9999	BAD_EPG_FORMAT
error_code_ext	4	0000 .. 9999	BAD_DATE_FORMAT BAD_VERSION_FORMAT BAD_SMS_EVENT_ID_FORMAT BAD_FLAG_FORMAT BAD_TIME_FORMAT BAD_RATING_FORMAT BAD_CRC_32

Gateway interface concerned

IMS and SMS

Errors must be reported by the SMS using the following table:

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
error_code	BAD_COMMAND_SYNTAX	BAD_ERROR_CODE
error_code_ext	BAD_COMMAND_SYNTAX	BAD_ERROR_CODE_EXT

CH - 1033 Cheseaux, Switzerland.
+41.21.732.03.11
+41.21.732.03.00

7.10 Command 211: Start of Report

Description

This command is used by the CC to signal the beginning of a call collector report. This command is sent as the first command of a call collector report (for the ICC whose UA is in the command header).

Syntax

Command 211: Start of Report			
Field	**byte**	**format**	**Description**
command_ID	4	0211	command_ID = 211
STU_callback_date	8	YYYYMMDD	date of the callback being reported
STU_callback_time	6	000000 .. 235959	time of the callback being reported

Gateway interface concerned

CC and SMS

Errors must be reported by the SMS using the following table:

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
STU_callback_date	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT
STU_callback_time	BAD_COMMAND_SYNTAX	BAD_TIME_FORMAT

7.11 Command 212: End of Report

Description

This command is used by the CC to signal the end of a call collector report. This command is sent as the last command after all call collector information (for the ICC whose UA is in the command header) is sent to the SMS.

Syntax

Command 212: End of Report			
Field	**byte**	**format**	**Description**
command_ID	4	0212	command_ID = 212
number_of_IPPV	2	00 .. 99	The number of IPPV reports (occurrences of command 202) that should have been sent in this report.

Gateway interface concerned

CC and SMS

Errors must be reported by the SMS using the following table:

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
number_of_IPPV	BAD_COMMAND_SYNTAX	BAD_NUMBER_OF_IPPV_FORMAT

7.12 Command 213: Event Product Schedule Change

Description

This command is used by the IMS to indicate that the schedule of an event, referenced by one or more event products has changed. The command is issued after processing the EPG data feed.

Syntax

Command 213: Event Product Schedule Change			
Field	**byte**	**format**	**Description**
command_ID	4	0213	command_ID = 213
SMS_event_ID	12	000000000000 .. 999999999999	SMS_event_ID of the event
Previous_start_date	8	YYYYMMDD	original start date of the event
Previous_start_time	6	HHMMSS	original start time of the event
Previous_duration	4	0000 .. 9999	original length of the event (0000..9999 minutes)
New_start_date	8	YYYYMMDD	new start date of the event
New_start_time	6	HHMMSS	new start time of the event
New_duration	4	0000 .. 9999	new length of the event (0000..9999 minutes)

Gateway interface concerned

IMS and SMS

Errors must be reported by the SMS using the following table:

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
SMS_event_ID	BAD_COMMAND_SYNTAX	BAD_SMS_EVENT_ID_FORMAT
Previous_start_date	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT
Previous_start_time	BAD_COMMAND_SYNTAX	BAD_TIME_FORMAT
Previous_duration	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT
New_start_date	BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT
New_start_time	BAD_COMMAND_SYNTAX	BAD_TIME_FORMAT
New_duration	BAD_COMMAND_SYNTAX	BAD_NUMBER_FORMAT

CH - 1033 Cheseaux, Switzerland. +41.21.732.03.11 +41.21.732.03.00

7.13 Command 214: Event remove error

Description

Returned when the IMS detects that an event referenced by one or more event products is no longer in the EPG data feed. In this case, a placeholder is created in the IMS database. This command is issued after processing the EPG data feed and only if the event was to occur within the next 7 days.

If the event was not in the 7 days time frame, then the placeholder is still created, but Command 214 will not be reported. The placeholder is used to allow the reporting of command 209 if the problem subsists.

Syntax

Command 214: Event remove error			
Field	**byte**	**format**	**Description**
command_ID	4	0214	command_ID = 214
SMS_event_ID	12	Num	SMS event ID of the event.

Gateway interface concerned

IMS and SMS

Errors must be reported by the SMS using the following table:

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
SMS_event_ID	BAD_COMMAND_SYNTAX	BAD_SMS_EVENT_ID_FORMAT

7.14 Command 215: Products List

Description

This command reports the list of products known to the CA system for one particular subscriber. It is sent asynchronously in answer from command 71: Get Products, and contains the transaction-number of the received instance of this last command as a reference.

Syntax

Command 215: Products List			
Field	**byte**	**format**	**Description**
command_ID	4	0215	command_ID = 215
original_transaction_number	9	Num	Transaction number of the command 71 that triggered the generation of this command
STU_number	14	Num	Nagravision STB serial number
ICC_suspended	1	Y..N	ICC suspend state
nb_of_products	2	00..99	Number of products returned in this command
for(i=0;i<nb_of_products;i++) {	3	Num	
IMS_product_ID	12	000000000000 to 999999999999	IMS product ID
product_suspended	1	Y..N	Product suspend state
}			

SMS Acknowledge

Field	Error codes	Error codes extension
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
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

Gateway interface concerned

SMS and SAS

8. Appendix: SMS PRODUCT_DEF Commands

8.1 Command 300: Create event product

Description

The event product is created in the IMS database. When the command is acknowledged the IMS product ID associated to this event by the IMS is returned in the IMS_product_ID field of acknowledge command (command 1000).

If there is no SMS_event_ID defined in the IMS database when the EPG enters the date frame delimited by the validity dates then command 209 (null event error) is reported to the SMS.

In addition to the errors that may occur because of a command's parameters, errors solely due to the database (impossibility to establish a connection to the database,...) are reported back with the error code DATABASE_ERROR. Such errors are not associated with any command parameter.

Syntax

Command 300: Create event product			
Field	**byte**	**format**	**Description**
command_ID	4	0300	command_ID = 300
SMS_product_ID	12	000000000000 to 999999999999	SMS product ID
PPV_number	7	0000000 to 9999999	Pay per view number
SMS_event_ID	12	000000000000 to 999999999999	SMS event ID
reference_number	4	0000 to 9999	Reference number for customer order
name	80	Alpha/Num	Product's name
description	250	Alpha/Num	Product's description
validity_start_date	8	YYYYMMDD	Validity start date.
validity_start_time	6	HHMMSS	Validity start time.
validity_end_date	8	YYYYMMDD	Validity end date. The SAS raises a PPV_IN_THE_PAST error if a command 10: Create PPV is received after this date.
validity_end_time	6	HHMMSS	Validity end time. The SAS raises a PPV_IN_THE_PAST error if a command 10: Create PPV is received after this time.
price	5	00000 … 99999	Price of the product representing the range $000.00 to $999.99
special_PPV_event	1	Y … N	define a special PPV (originating an immediate call back)
impulse_purchase_allowed	1	Y … N	YES is the default
watched_criterion	3	000 .. 999	Amount of time (in cryptoperiods) that must be spent on the event before it is considered as watched.

free_preview_time	2	00 .. 99	Length of the free preview (in minutes) at the beginning of the event.
reverse_blackout_flag	1	Y … N	N is the default
blackout_type	2	00 .. 99	number of the blackout type to enforce for this event.
nb_of_subtypes	3	000 .. 999	number of subtypes defined for this event.
for (I=1; I<=nb_of_subtypes; I++) {			
blackout_subtype	3	000 .. 999	subtype number for each subtype belonging to the blackout type.
}			

SMS Acknowledge

Field	Error codes	Error codes extension
	DATABASE_ERROR	
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
SMS_product_ID	PRODUCT_ALREADY_EXISTS PRODUCT_ALREADY_EXISTS BAD_COMMAND_SYNTAX	IDENTICAL_PRODUCTS DIFFERENT_PRODUCTS BAD_SMS_PRODUCT_ID_FORMAT
PPV_number	BAD_COMMAND_SYNTAX	BAD_PPV_NUMBER_FORMAT
SMS_event_ID	BAD_COMMAND_SYNTAX PRODUCT_ON_NON_PPV_EVENT	BAD_SMS_EVENT_ID_FORMAT NO_EXTENDED_ERROR_CODE
reference_number	BAD_COMMAND_SYNTAX	BAD_REFERENCE_NUMBER_FORMAT
validity_start_date	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE
validity_start_time	BAD_COMMAND_SYNTAX	BAD_TIME_FORMAT
validity_end_date	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE
validity_end_time	BAD_COMMAND_SYNTAX	BAD_TIME_FORMAT
price	BAD_COMMAND_SYNTAX	BAD_PRICE_FORMAT
Special_PPV_event	BAD_COMMAND_SYNTAX	BAD_FLAG_FORMAT
impulse_purchase_allowed	BAD_COMMAND_SYNTAX	BAD_FLAG_FORMAT
watched_criterion	BAD_COMMAND_SYNTAX	BAD_FLAG_FORMAT
free_preview_time	BAD_COMMAND_SYNTAX	BAD_TIME_FORMAT
reverse_blackout_flag	BAD_COMMAND_SYNTAX	BAD_FLAG_FORMAT
blackout_type	BAD_COMMAND_SYNTAX	BAD_BLACKOUT_TYPE_FORMAT
nb_of_subtypes	BAD_COMMAND_SYNTAX	BAD_NB_OF_SUBTYPES_FORMAT
blackout_subtype	BAD_COMMAND_SYNTAX	BAD_BLACKOUT_SUBTYPE_FORMAT

Gateway interface concerned

SMS and IMS database

8.2 Command 301: Remove product

Description

Removes the product from the IMS database. It is not removed from the ICCs. It is the responsibility of the SMS to manage the case of customers having already purchased the product.

Syntax

Command 301: Remove product			
Field	**byte**	**format**	**Description**
command_ID	4	0301	command_ID = 301
IMS_product_ID	12	000000000000… 999999999999	IMS Product ID

SMS Acknowledge

Field	Error codes	Error codes extension
	DATABASE_ERROR	
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
IMS_product_ID	BAD_COMMAND_SYNTAX PRODUCT_NOT_FOUND	BAD_IMS_PRODUCT_ID_FORMAT NO_EXTENDED_ERROR_CODE

Gateway interface concerned

SMS and IMS database

8.3 Command 302: Modify event product

Description

The event product is modified in the IMS database.

Syntax

Command 302: Modify event product			
Field	**byte**	**format**	**Description**
command_ID	4	0302	command_ID = 302
IMS_product_ID	12	000000000000 … 999999999999	IMS Product ID
reference_number	4	0000 … 9999	Reference number for customer order
validity_start_date	8	YYYYMMDD	Validity start date.
validity_start_time	6	HHMMSS	Validity start time.
validity_end_date	8	YYYYMMDD	Validity end date. The SAS raises a PPV_IN_THE_PAST error if a command 10: Create PPV is received after this date.
validity_end_time	6	HHMMSS	Validity end time. The SAS raises a PPV_IN_THE_PAST error if a command 10: Create PPV is received after this time.
price	5	00000 … 99999	Price of the product representing the range $000.00 to $999.99
special_PPV_event	1	Y … N	define a special PPV (originating an immediate call back)
impulse_purchase_allowed	1	Y … N	YES is the default
watched_criterion	3	000 .. 999	Amount of time (in cryptoperiods) that must be spent on the event before it is considered as watched.
free_preview_time	2	00 .. 99	Length of the free preview (in minutes) at the beginning of the event.
reverse_blackout_flag	1	Y … N	N is the default
blackout_type	2	00 .. 99	number of the blackout type to enforce for this event.
nb_of_subtypes	3	000 .. 999	number of subtypes defined for this event.
for (I=1; I<=nb_of_subtypes; I++) {			
blackout_subtype	3	000 .. 999	subtype number for each subtype belonging to the blackout type.
}			

SMS Acknowledge

Field	Error codes	Error codes extension
	DATABASE_ERROR	
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
IMS_product_ID	BAD_COMMAND_SYNTAX PRODUCT_NOT_FOUND	BAD_IMS_PRODUCT_ID_FORMAT NO_EXTENDED_ERROR_CODE
reference_number	BAD_COMMAND_SYNTAX	BAD_REFERENCE_NUMBER_FORMAT
validity_start_datedate	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE
validity_start_time	BAD_COMMAND_SYNTAX	BAD_TIME_FORMAT
validity_end_date	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE
validity_end_time	BAD_COMMAND_SYNTAX	BAD_TIME_FORMAT
price	BAD_COMMAND_SYNTAX	BAD_PRICE_FORMAT
Special_PPV_event	BAD_COMMAND_SYNTAX	BAD_FLAG_FORMAT
impulse_purchase_allowed	BAD_COMMAND_SYNTAX	BAD_FLAG_FORMAT
watched_criterion	BAD_COMMAND_SYNTAX	BAD_FLAG_FORMAT
free_preview_time	BAD_COMMAND_SYNTAX	BAD_TIME_FORMAT
reverse_blackout_flag	BAD_COMMAND_SYNTAX	BAD_FLAG_FORMAT
blackout_type	BAD_COMMAND_SYNTAX	BAD_BLACKOUT_TYPE_FORMAT
nb_of_subtypes	BAD_COMMAND_SYNTAX	BAD_NB_OF_SUBTYPES_FORMAT
blackout_subtype	BAD_COMMAND_SYNTAX	BAD_BLACKOUT_SUBTYPE_FORMAT

Gateway interface concerned

SMS and IMS database

8.4 Command 303: Create service product

Description

The product based on a service is automatically created in the IMS database. Command 1000 is returned with the IMS product ID.

Syntax

Command 303: Create service product			
Field	**byte**	**format**	**Description**
command_ID	4	0303	command_ID = 303
SMS_product_ID	12	000000000000 to 999999999999	SMS product ID
service_UID	5	00000 … 99999	IMS service_UID
reference_number	4	0000 … 9999	Reference number for customer order
name	80	Alpha/Num	Product name
description	250	Alpha/Num	Product description
validity_start_date	8	YYYYMMDD	Validity start date
validity_end_date	8	YYYYMMDD	Validity end date
price	5	00000 … 99999	Price of the product representing the range $000.00 to $999.99

SMS Acknowledge

Field	Error codes	Error codes extension
	DATABASE_ERROR	
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
SMS_product_ID	BAD_COMMAND_SYNTAX PRODUCT_ALREADY_EXISTS PRODUCT_ALREADY_EXISTS	BAD_SMS_PRODUCT_ID_FORMAT IDENTICAL_PRODUCTS DIFFERENT_PRODUCTS
service_UID	BAD_COMMAND_SYNTAX SERVICE_NOT_FOUND	BAD_SERVICE_UID_FORMAT NO_EXTENDED_ERROR_CODE
reference_number	BAD_COMMAND_SYNTAX	BAD_REFERENCE_NUMBER_FORMAT
validity_start_date	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE
validity_end_date	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE
price	BAD_COMMAND_SYNTAX	BAD_PRICE_FORMAT

Gateway interface concerned

SMS and IMS database

8.5 Command 304: modify service product

Description

The product based on a service is modified in the IMS database.

Syntax

Command 304: Modify service product			
Field	**byte**	**format**	**Description**
command_ID	4	0304	command_ID = 304
IMS_product_ID	12	000000000000 … 999999999999	IMS product ID
service_UID	5	00000 … 99999	IMS service UID
reference_number	4	0000 … 9999	Reference number for customer order
name	80	Alpha/Num	Product name
description	250	Alpha/Num	Product description
validity_start_date	8	YYYYMMDD	Validity start date
validity_end_date	8	YYYYMMDD	Validity end date
price	5	00000 … 99999	Price of the product representing the range $000.00 to $999.99

Note: IMS_product_ID identifies the product;
service_UID allows to change the service the product is based on.

SMS Acknowledge

Field	Error codes	Error codes extension
	DATABASE_ERROR	
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
IMS_PRODUCT_ID	BAD_COMMAND_SYNTAX PRODUCT_NOT_FOUND	BAD_IMS_PRODUCT_ID_FORMAT NO_EXTENDED_ERROR_CODE
service_UID	BAD_COMMAND_SYNTAX SERVICE_NOT_FOUND	BAD_SERVICE_UID_FORMAT NO_EXTENDED_ERROR_CODE
reference_number	BAD_COMMAND_SYNTAX	BAD_REFERENCE_NUMBER_FORMAT
validity_start_date	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE
validity_end_date	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE
price	BAD_COMMAND_SYNTAX	BAD_PRICE_FORMAT

Gateway interface concerned

SMS and IMS database

8.6 Command 305: Create service package product

Description

The product that is a package of services is created in the IMS database. Command 1000 is returned with IMS_product_ID. The number of services and their values is constrained by the fact that a product may require up to 3 rights.

Syntax

Command 305: Create service package product			
Field	**byte**	**format**	**Description**
command_ID	4	0305	command_ID = 305
SMS_product_ID	12	000000000000 to 999999999999	SMS_package_ID
reference_number	4	0000 … 9999	Reference number for customer order
name	80	Alpha/Num	Product name
description	250	Alpha/Num	Product description
validity_start_date	8	YYYYMMDD	Validity start date
validity_end_date	8	YYYYMMDD	Validity end date
price	5	00000 … 99999	Price of the product representing the range $000.00 to $999.99
service_number	3	000 … 500	Number of services included in the package
for (I=1; I <= service_number; I++) {			
Service_UID	5	00000 … 99999	IMS service UID
}			

SMS Acknowledge

Field	Error codes	Error codes extension
	DATABASE_ERROR	
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
SMS_product_ID	BAD_COMMAND_SYNTAX PRODUCT_ALREADY_EXISTS PRODUCT_ALREADY_EXISTS	BAD_SMS_PRODUCT_ID_FORMAT IDENTICAL_PRODUCTS DIFFERENT_PRODUCTS
reference_number	BAD_COMMAND_SYNTAX	BAD_REFERENCE_NUMBER_FORMAT
validity_start_date	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE
validity_end_date	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE
price	BAD_COMMAND_SYNTAX	BAD_PRICE_FORMAT
service_number	BAD_COMMAND_SYNTAX TOO_MANY_ITEMS	BAD_SERVICE_NUMBER_FORMAT maximum item #
service_UID	BAD_COMMAND_SYNTAX SERVICE_NOT_FOUND	BAD_SERVICE_UID_FORMAT NO_EXTENDED_ERROR_CODE

Gateway interface concerned

SMS and IMS database

8.7 Command 306: modify service package product

Description

The product that is a package of services is modified in the IMS database. The number of services and their values is constrained by the fact that a product may require up to 3 rights.

Syntax

Command 306: Modify service package product			
Field	**byte**	**format**	**Description**
command_ID	4	0306	command_ID = 306
IMS_product_ID	12	000000000000 to 999999999999	IMS product ID
reference_number	4	0000 … 9999	Reference number for customer order
name	80	Alpha/Num	Product name
description	250	Alpha/Num	Product description
validity_start_date	8	YYYYMMDD	Validity start date.
validity_end_date	8	YYYYMMDD	Validity end date.
price	5	00000 … 99999	Price of the product representing the range $000.00 to $999.99
service_number	3	000 … 999	Number of services included in the package
for (I=1; I <= service_number; I++) {			
Service_UID	5	00000 … 99999	IMS service UID
}			

IMS Acknowledge

Field	Error codes	Error codes extension
	DATABASE_ERROR	
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
IMS_product_ID	BAD_COMMAND_SYNTAX PRODUCT_NOT_FOUND	BAD_IMS_PRODUCT_ID_FORMAT NO_EXTENDED_ERROR_CODE
reference_number	BAD_COMMAND_SYNTAX	BAD_REFERENCE_NUMBER_FORMAT
validity_start_date	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE
validity_end_date	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE
price	BAD_COMMAND_SYNTAX	BAD_PRICE_FORMAT
service_number	BAD_COMMAND_SYNTAX TOO_MANY_ITEMS	BAD_SERVICE_NUMBER_FORMAT maximum item #
service_UID	BAD_COMMAND_SYNTAX SERVICE_NOT_FOUND	BAD_SERVICE_UID_FORMAT NO_EXTENDED_ERROR_CODE

Gateway interface concerned

SMS and IMS database

CH - 1033 Cheseaux, Switzerland. +41.21.732.03.11 +41.21.732.03.00

8.8 Command 307: Create event package product

Description

The product that is a package of events is created in the IMS database. The validity start_date and validity end_date are stored in the IMS database. Then the command 1000 is returned with IMS_product_ID.

If for any reason, when the EPG enters the validity dates and range, and there is no event with the SMS_event_ID, then Command 209 (null event error) is reported to the SMS.

The number of events and their values is constrained by the fact that a product may require up to 3 rights.

Syntax

Command 307: Create event package product			
Field	**byte**	**format**	**Description**
command_ID	4	0307	command_ID = 307
SMS_product_ID	12	000000000000 to 999999999999	SMS product ID
reference_number	4	0000 … 9999	Reference number for customer order
name	80	Alpha/Num	Product name
description	250	Alpha/Num	Product description
validity_start_date	8	YYYYMMDD	Validity start date.
validity_end_date	8	YYYYMMDD	Validity end date.
price	5	00000 … 99999	Price of the product representing the range $000.00 to $999.99
event_number	3	000 … 999	Number of events included in the package
for (I=1; I <= event_number; I++) {			
SMS_event_ID	12	000000000000 to 999999999999	SMS event ID
}			
tokens	4	0000 to 9999	Number of available prepaid tokens

CH - 1033 Cheseaux, Switzerland. +41.21.732.03.11 +41.21.732.03.00

SMS Acknowledge

Field	Error codes	Error codes extension
	DATABASE_ERROR	
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID_FORMAT
SMS_product_ID	BAD_COMMAND_SYNTAX PRODUCT_ALREADY_EXISTS PRODUCT_ALREADY_EXISTS	BAD_SMS_PRODUCT_ID_FORMAT IDENTICAL_PRODUCTS DIFFERENT_PRODUCTS
reference_number	BAD_COMMAND_SYNTAX	BAD_REFERENCE_NUMBER_FORMAT
validity_start_date	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE
validity_end_date	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE
price	BAD_COMMAND_SYNTAX	BAD_PRICE_FORMAT
event_number	BAD_COMMAND_SYNTAX TOO_MANY_ITEMS	BAD_EVENT_NUMBER_FORMAT maximum item #
PPV_number	BAD_COMMAND_SYNTAX	BAD_PPV_NUMBER_FORMAT
SMS_event_ID	BAD_COMMAND_SYNTAX SMS_EVENT_ID_NOT_FOUND PRODUCT_ON_NON_PPV_EVENT	BAD_SMS_EVENT_ID_FORMAT NO_ERROR_CODE_EXTENSION NO_ERROR_CODE_EXTENSION
tokens	BAD_COMMAND_SYNTAX	BAD_TOKEN_NUMBER_FORMAT

Gateway interface concerned

SMS and IMS database

CH - 1033 Cheseaux, Switzerland. +41.21.732.03.11 +41.21.732.03.00

8.9 Command 308: modify event package product

Description

The product that is a package of events is modified in the IMS database.

Syntax

Command 308: Modify event package product			
Field	**byte**	**format**	**Description**
command_ID	4	0308	command_ID = 308
IMS_product_ID	12	000000000000 to 999999999999	IMS product ID
reference_number	4	0000 … 9999	Reference number for customer order
name	80	Alpha/Num	Product name
description	250	Alpha/Num	Product description
validity_start_date	8	YYYYMMDD	Validity start date.
validity_end_date	8	YYYYMMDD	Validity end date.
price	5	00000 … 99999	Price of the product representing $000.00 to $999.99
event_number	3	000 … 999	Number of events included in the package
for (I=1; I <= event_number; I++) {			
SMS_event_ID	12	000000000000 to 999999999999	SMS event ID
}			
tokens	4	0000 to 9999	Number of available prepaid tokens

CH - 1033 Cheseaux, Switzerland. +41.21.732.03.11 +41.21.732.03.00

SMS Acknowledge

Field	Error codes	Error codes extension
	DATABASE_ERROR	
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID_FORMAT
IMS_product_ID	BAD_COMMAND_SYNTAX PRODUCT_NOT_FOUND	BAD_IMS_PRODUCT_ID_FORMAT NO_EXTENDED_ERROR_CODE
reference_number	BAD_COMMAND_SYNTAX	BAD_REFERENCE_NUMBER_FORMAT
validity_start_date	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE
validity_end_date	BAD_COMMAND_SYNTAX BAD_COMMAND_SYNTAX	BAD_DATE_FORMAT BAD_DATE_SEQUENCE
price	BAD_COMMAND_SYNTAX	BAD_PRICE_FORMAT
event_number	BAD_COMMAND_SYNTAX TOO_MANY_ITEMS	BAD_EVENT_NUMBER_FORMAT maximum item #
PPV_number	BAD_COMMAND_SYNTAX	BAD_PPV_NUMBER_FORMAT
SMS_event_ID	BAD_COMMAND_SYNTAX SMS_EVENT_ID_NOT_FOUND	BAD_EVENT_ID_FORMAT NO_ERROR_CODE_EXTENSION
tokens	BAD_COMMAND_SYNTAX	BAD_TOKEN_NUMBER_FORMAT

Gateway interface concerned

SMS and IMS database

8.10 Command 309: Update event PPV number

Description

Update the event PPV_number in the IMS database. If the PPV_number is set to 0 (zero), then the PPV_number cannot be used in any event product or event package product.

This command must be used to set the PPV_number to 0 (zero) when all event products and packages have been removed by the SMS for this event. Failing to do so may cause the event to be broadcast as a PPV, while no event product and no event package exists for this event.

Syntax

Command 309: Update event PPV number			
Field	**byte**	**format**	**Description**
command_ID	4	0309	command_ID = 309
SMS_event_ID	12	000000000000 to 999999999999	SMS event ID
PPV_number	7	0000000 to 9999999	Pay per view number

SMS Acknowledge

Field	Error codes	Error codes extension
	DATABASE_ERROR	
command_ID	BAD_COMMAND_SYNTAX	BAD_COMMAND_ID
SMS_event_ID	BAD_COMMAND_SYNTAX	BAD_SMS_EVENT_ID_FORMAT
PPV_number	BAD_COMMAND_SYNTAX	BAD_PPV_NUMBER_FORMAT

Gateway interface concerned

SMS and IMS database

9. Appendix: SMS OPERATION Commands

9.1 Command 1000: Acknowledge Command

Description

The command specified has been completed without error. If the command being acknowledged with the command 1000 contained a SMS_product_ID, the fields IMS_product_ID and SMS_product_ID of the acknowledgment command will be documented. In all other cases, these fields will be set respectively to 000000000000 and 000000000000.

Syntax

Command 1000: Acknowledge Command			
Field	**byte**	**format**	**Description**
command_ID	4	1000	command_ID = 1000
transaction_ number	9	000000000 to 999999999	transaction number acknowledged
IMS_product_ID (if applicable)	12	000000000000 to 999999999999	IMS product ID
SMS_product_ID (if applicable)	12	000000000000 to 999999999999	SMS product ID

Gateway interface concerned

All

9.2 Command 1001: Non-acknowledge Command

Description

The command specified could not be completed.

Syntax

Command 1001: Non-Acknowledge Command			
Field	**byte**	**format**	**Description**
command_ID	4	1001	command ID = 1001
transaction_ number	9	000000000 … 999999999	transaction_number acknowledged
nack_status	1	1 … 2	1 = REJECTED means that the command has been rejected. 2 = POSTPONED means that the command could not be completed because of a technical problem in the addressable system (SMS, SAS, CC or IMS database). In this case, the command should be resubmitted after a time delay.
error_code	4	0000 … 9999	Main error code (PRODUCT_NOT_FOUND, BAD_COMMAND_SYNTAX,...)
error_code_ext	4	0000 … 9999	Extension error code BAD_DATE_FORMAT, BAD_IMS_PRODUCT_ID_FORMAT, ...)
length_of_command_body	3	000 … 999	
command_section			command section of the command that caused the error.

Gateway interface concerned

All

9.3 Command 1002: No Command

Description

This command will be transmitted when no other commands are sent. It allows both systems SMS and SMS Gateway to check anytime the communication layer.

Syntax

Command 1002: No Command			
Field	**byte**	**format**	**Description**
command_ID	4	1002	command_ID = 1002

Gateway interface concerned

SMS Gateway and SMS

10. Appendix: Device_IO Protocol

10.1 Overview

The Device_IO protocol manages communications between applications running on the same or on different machines. It is based on the notion of server. Device_IO is a protocol that runs on top of TCP/IP.

Communications are established through entry points called services. An application wishing to establish a communication with another one must specify the service name it wants to communicate through.

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 questions 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 or from a Device_IO server.

Nagravision will provide two IP port numbers created by the SMS Gateway process. The first will be used to send commands from SMS to the SMS-Gateway as EMM commands, control commands and product definition commands. The second one will be used to receive commands from the SMS-Gateway, as feedback commands.

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. An example of how a client can initiate a connection (on top of TCP) is given in section 121.

Parameter description

call_address	Full address of the Device_IO server entry point. The call address contains the full specification of the server machine address (node name…) and the TCP-IP port number.
len	length of the message to be sent or received (in bytes) following the length field. The first transmitted byte is the most significant byte. Len = (MSB, LSB).
data	user data to send or receive during normal information exchanges (after connection setup). data is a string of bytes of length at least 0 and at most 32767.
op_mode	Data transfer operation mode 0 = Normal data transfer (allow message tracing) 1 = Fast data transfer (do not allow message tracing)
obj_name_len	Length of the object name attribute (in bytes).
obj_name	Name of the applicative service to which the connection should be made. Obj_name must be a string of bytes at least one byte long and at most 32 byte long: 1 <= obj_name_len <= 32. This Name is compulsory but its content is up to the client.
user_data_c	User data supplied by the client to the server. May contain information to help the server deciding whether the communication should be accepted or rejected. user_data_c is a string of bytes whose length is at least 0 and at most 1024.
connect_status	Connection status (see the connection status code table)
answer_code	Specify whether the communication is accepted or rejected (only if the connection status is success)

	0 = call accepted 1 = call rejected
user_data_s	User data sent by the Device_IO server to the client. Depend on the application. user_data_c is a string of bytes whose length is at least 0 and at most 1024.

10.1.1 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 given:

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

Once the communication with the server has been established, the name of the target service possibly along with user data, should be communicated to the Device_IO server in order to establish a link between the calling client and the requested service.

As a reply the server will send one or two messages. The first message contains a connection status and, if the communication attempt has been successful, a second message will specify whether the call has been accepted or rejected.

10.1.1.1 Operations

To open a communication with the Device_IO server, the following operations must be done in sequence by the client application.

10.1.1.1.1 call the server

open <call address>
send message_1

Syntax of message_1

Syntax	Bits	Format
len	16	uimsbf
op_mode	8	uimsbf
ob_name_len	8	uimsbf
for (I=0; I<ob_name_len; I++) {		
obj_name [I]	8	uimsbf
}		
for (J=I; I<len-2; I++, J++) {		
user_data_c [J]	8	uimsbf
}		

10.1.1.1.2 wait for the answer

receive message_2

if status = success :
receive message_3

Syntax of message_2

Syntax	Bits	Format
len	16	'0000000000000001'
connect_status	8	uimsbf

Syntax of message 3

Syntax	Bits	Format
len	16	uimsbf
answer_code	8	uimsbf
ob_name_len	8	uimsbf
for (I=0; len-1; I++) {		
user_data_s [I]	8	uimsbf
}		

The first answer is generally immediate as it comes directly from the Device_IO server itself (within a second). The second answer may take longer as it comes from the client application.

10.1.2 Data exchange between client and server

10.1.2.1 Operations

10.1.2.1.1 send data toward the Device_IO server

send message_5

Syntax of message_5

Syntax	Bits	Format
len	16	uimsbf
for (I=0;I<=len-1; I++) {		
data [I]	8	uimsbf
}		

10.1.2.1.2 receive data from the Device_IO server

receive message_5

10.1.3 Status code table

Connection status identifier	Code (hex)	Description
CONNECT_FAILURE	00	The connection has failed for any unexpected reason.
ERROR_PROROCOL	02	There was a protocol error during the communication establishment or during data transfer (see the remark below).
LINK_HANDLER_BUSY	04	The link handler is busy. The communication cannot be established at this moment

NO_FREE_LINK	05	There is no available free link.
SUCCESS	06	The operation has been successfully done.
UNKNOWN_SERVICE	09	The specified service is unknown

The protocol error status is returned for any of the following event:

1) The data length field does not match the actual data length.
2) The operation mode, the answer code or the connection status has an unexpected value.
3) Any situation that would raise an error on received or transmitted data (message too long…).

10.2 Example: connecting to a Device_IO server

An example of a client opening a connection with a Device_IO server from a C program follows:

```
struct device_call {                                    define the structure needed to hold the parameters
        short int       len;                            to send messages.
        unsigned char   op_mode;
        unsigned char   obj_name_len;
        char            obj_name[32];
        char            user_data[255];
}

struct device_call device_call;

struct device_status {                                  define the structure needed to hold the device_io status
        short int       len;
        unsigned char   status;
}

struct device_status device_status;

struct device_answer {                                  define the structure needed to hold data received
        short int       len;                            from the Device_IO link when the server is first answering
        unsigned char   code;
        char            data[1025];
}

struct device_answer device_answer;

struct device_data {                                    define the structure needed to hold data received
        short int       len;                            from the Device_IO link during subsequent communication
        char            data[1025];                     exchanges.
}

struct device_data device_data;
.
.
.
main ….. {
int I, channel, status:
.
.
tcp_connect (“machine1”, “TCP_IP_SERVICE”, &channel, &status);
                                                        create a TCP-IP socket, pass the name of the
```

```
.                                                    machine and the name of the service with
.                                                    which a communication must be established.
.
.
device_call.op_mode = 0;                             choose normal data transfer.
strcpy (device_call.obj_name, “SERVICE_NAME”);       and
device_call.obj_name_len = strlen (device_call.obj_name);   document the device_call structure
len = device_call.obj_name_len + 1 + 1;
.
l = write (channel, &device_call, len);              write the device_call structure.
.
recv (channel, &device_status, 3, 0);                receive from the device_server a device_status.
.
.
read (channel, &device_answer, sizeof (struct device_answer));  assuming the status indicated a success,
.                                                    receive from the server the first answer.
.
.
read (channel, &device_data, sizeof (struct device_data));      Past the first answer, read data sent from the
.                                                    Device_IO server into the device_data
                                                     structure.
.
write (channel, &device_data, sizeof (struct device_data));     Send data to the Device_IO server.
}
```